Minimum complexity principle for knowledge transfer in artificial learning

Pierre-Alexandre Murena

université PARIS-SACLAY

TELECOM ParisTech

NNT : 2018SACLT019

Thèse de doctorat

Minimum Complexity Principle for Knowledge Transfer in Artificial Learning

Thèse de doctorat de l'Université Paris-Saclay
préparée à Télécom ParisTech

Ecole doctorale n°580 Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Paris, le 14 décembre 2018, par

PIERRE-ALEXANDRE MURENA

Composition du Jury :

Jean-Louis Dessalles Maître de Conférences, Télécom ParisTech	Co-Directeur de thèse
Antoine Cornuéjols Professeur, AgroParisTech	Co-Directeur de thèse
Shai Ben-David Professeur, Universy of Waterloo	Rapporteur
Amaury Habrard Professeur, Université Jean Monnet	Rapporteur
Florence d'Alché-Buc Professeur, Télécom ParisTech	Présidente
Jean Lieber Maître de Conférences, Université de Lorraine	Examinateur
Laurent Orseau Chercheur, Google DeepMind	Examinateur
Gilles Richard Professeur, Université Paul Sabatier	Examinateur

"Roger has tried to explain to her the V-bomb statistics: the difference between distribution, in angel's-eye view, over the map of England, and their own chances, as seen from down here. She's almost got it: nearly understands his Poisson equation, yet can't quite put the two together—put her own enforced calm day-to-day alongside the pure numbers, and keep them both in sight. Pieces keep slipping in and out. "

Thomas Pynchon, *Gravity's Rainbow*

TÉLÉCOM PARISTECH

Abstract

Doctor of Philosophy

Minimum Complexity Principle for Knowledge Transfer in Artificial Learning

by Pierre-Alexandre MURENA

Classical learning methods are often based on a simple but restrictive assumption: The present and future data are generated according to the same distributions. This hypothesis is particularly convenient when it comes to developing theoretical guarantees that the learning is accurate. However, it is not realistic from the point of view of applicative domains that have emerged in the last years.

In this thesis, we focus on four distinct problems in artificial intelligence, that have mainly one common point: All of them imply knowledge transfer from one domain to the other. The first problem is analogical reasoning and concerns statements of the form "A is to B as C is to D". The second one is transfer learning and involves classification problem in situations where the training data and test data do not have the same distribution (nor even belong to the same space). The third one is data stream mining, ie. managing data that arrive one by one in a continuous and high-frequency stream with changes in the distributions. The last one is collaborative clustering and focuses on exchange of information between clustering algorithms to improve the quality of their predictions.

The main contribution of this thesis is to present a general framework to deal with these transfer problems. This framework is based on the notion of Kolmogorov complexity, which measures the inner information of an object. This tool is particularly adapted to the problem of transfer, since it does not rely on probability distributions while being able to model the changes in the distributions.

Apart from this modeling effort, we propose, in this thesis, various discussions on aspects and applications of the different problems of interest. These discussions all concern the possibility of transfer in multiple domains and are not based on complexity only.

Acknowledgements

I will like to express my gratitude to all the people who brought their support during the last three years and made my research possible and comfortable.

First of all, I would like to thank my two advisors, Antoine Cornuéjols and Jean-Louis Dessalles, for initiating this thesis, giving me lots of precious remarks and opening my eyes to many research problems, even research domains, the existence of which I could not even imagine. Thanks to our exchanges, I discovered thrilling domains, some of them being discussed in the following pages, some being left aside for future works. More specifically, I would like to thank Antoine for accepting me as a PhD student and allowing me to explore a large variety of problems. I would also like to thank Jean-Louis for the multiple off-topic discussions (in particular about Bartok, Jung and others...) and for the trust he puts in me when it comes to research but also to teaching.

I also thank my reviewers, Amaury Habrard and Shai Ben-David, for taking time to read these pages and providing me insightful remarks. In addition, my thanks go to the whole jury, to Florence d'Alché-Buc, Jean Lieber, Laurent Orseau and Gilles Richard. I am particularly grateful to Jean Lieber, who took an impressive amount of his time to correct my thesis in details and provide me with constructive remarks. I hope that we will have the opportunity to work together on some common problems in the future. Finally, I would like to thank Gilles Richard for having followed me in the last two years and being a supportive presence, despite the distance (Toulouse is definitely not too far, but shall I mention Australia?).

I am also grateful to the Institut Mines-Télécom and especially to the program Futur & Rupture for financing my research.

What would a PhD thesis be without the thanks to my colleagues? Insisting on the importance of the workplace and of the colleagues' support might sound like a platitude, but I have discovered how accurate it is actually.

My thanks go to the entire DIG team (former DBWeb team) which welcomed me warmly. With you all, I had particularly interesting discussions, some scientific, some not at all; but both contributed actively to my complete integration to the world of research. My thanks go to Albert Bifet, Mauro Sozio, Fabian Suchanek, Antoine Amarilli, Pierre Senellart (even if you left!), Talel Abdessalem and Marc Jeanmougin. Particular thanks go to doctors-to-be and ex-doctors-to-be Oana Balalau, Jean-Benoit Griesner, Thomas Rebele, Luis Galarraga, Jonathan Lajus, Julien Romero, Thomas Pélissier, Jacob Montiel, Maroua Bahri. A special mention comes to Quentin Lobbé for keeping me company in front of the Futur & Rupture posters and for the good conversations on cinema.

I would also like to thank LInK team, and in particular Stéphane Dervaux, Joon Kwon, Juliette Dibie, Cristina Manfredotti and Liliana Ibanescu; but also other teams in MIA-518, with special attention to Liliane Bel, Céline Lévy-Leduc, Jade Giguelay, Stéphane Robin and Julien Chiquet. My special thanks go to the old-timers from LInK, Jérémie Sublime (who will come back later in these acknowledgments), Mathieu Bouyrie, Joe Raad (not the singer) and Asma Dachraoui; and for the past, current and future PhD students of MIA: Mélanie Munch, Pierre Lejeail, Annarosa Quarello (che bene!), Martina Sundqvist, Rana Jreich, Raphaelle Momal, Mathieu Carmassi, Marie Perrot and Félix Cheysson. Of course, I could not forget two other old-timers, Mounia Zaouche and Paul Bastide, nor more transient elements: Irène Demongeot, Raphael Olivier, Serife, Dylan, Jules, Jiang, Camille and Marion. Last but not least, I would like to address all my gratitude to Sema Akkoyunlu, too many times the indisputable winner of the "Colleague of the Month" award.

In addition, I would like to express all my gratitude to my scientific collaborators, for opening my mind and spending excellent work time together. I think of course of Jérémie Sublime (again, but a “Docteur Sublime" undoubtedly deserves twice as many acknowledgments as a regular doctor) and Basarab Matei: I am sure we will keep working together in the future; and, of course, to Marie Al-Ghossein, even if we could not name our algorithm HILDA and if you did not respect my pictures of seals.

A special thought comes to my students at Télécom ParisTech who worked with me on my projects or who just brought me satisfaction and cheerfulness when the research was not at its highest: Antoine, Antoine, Julien, Lucas, Madeleine, Marie, Pierre-Francois, Raphaelle and Sibille. I wish you all a rich career!

To conclude with the “professional" relationships, I would like to thank all these people who contributed to my work in a less direct way: Irène Desécures, for being the most devoted, patient and considerate person, Lidia Forero, Stéphanie Druetta and Anne Vilnat.

The second part of these acknowledgments is dedicated to those who contributed in a more distant way to my work, by encouraging me, bringing me constant support and friendship.

First of all, how could I omit my family, who has always been here, even if research remains a strange world to them! I think in particular to my very close family, my mother and my grandmother who have never let me down. Almost in my family, even if not genetically related (at least not on the last dozens of generations!), I would like to express my warm gratitude to Daniel Gavrel who has always encouraged me, before and during my PhD.

I would also like to thank Lucia, to whom I could always complain about how difficult a PhD can be, even if her studies were not much easier!

Very high-ranked on the list of people without whom this thesis would not exist, I would like to mention a couple of teachers: Jean-Louis Orengia, Pierre Brunel and Philippe Manevy, for encouraging me way before I knew what I would do later, Stéphane Gonnord, for showing me how intuitive maths can be, Fabrice Monfront, for giving me an insight of how exquisite good maths can be, and Jean-Pierre Barani, for teaching me how to call the best out of me in maths.

I will conclude by mentioning my very close and faithful friends, Alexia, Marion and Bertrand, but also Sylwia and Serena, who were all comforting landmarks when I felt lost, as well as my old friend Nicolas with whom I had my first scientific discoveries more than ten years ago.

Finally, I would like to thank Olivia for her indirect but precious help, at a time where it was really needed.

Contents

List of Figures

List of Tables

List of Abbreviations

ADWIN	**AD**aptive **WIN**dowing
AWILDA	**A**daptive **W**indowing based **I**ncremental **L**atent **D**irichlet **A**llocation
CBR	**C**ase-**B**ased **R**easoning
DGM	**D**escriptive **G**raphical **M**odel
DL	**D**escription **L**ength
ERM	**E**mpirical **R**isk **M**inimization
KKT	**K**arush-**K**uhn-**T**ucker
LDA	**L**atent **D**irichlet **A**llocation
LCFQ	**L**earning with **C**oncern for **F**uture **Q**uestion
MDL	**M**inimum **D**escription **L**ength
MML	**M**inimum **M**essage **L**ength
MF	**M**atrix **F**actorization
PGM	**P**robabilistic **G**raphical **M**odels
RS	**R**ecommender **S**ystem

In memory of my grandfather

Chapter 1

Introduction: Knowledge Transfer in Artificial Intelligence

1.1 Scope

Recent advances in machine learning and artificial intelligence show an interesting phenomenon. On the one hand, computational intelligence models break unexpected records on very complex tasks, such as playing Go (Silver et al., 2016), image recognition (see for instance the impressive scores obtained in the Large Scale Visual Recognition Challenge (Russakovsky et al., 2015)) or self-driving vehicles. On the other hand, the "intelligence" of the systems is limited in unexpected directions. For instance, the learning systems usually can produce errors that humans would not: unrecognizable images can be attributed with near-certainty by deep convolutional neural networks to recognizable categories (Nguyen, Yosinski, and Clune, 2015), or addition of a simple and barely visible perturbation to images can be enough to lead to drastic classification errors (Moosavi-Dezfooli, Fawzi, Fawzi, and Frossard, 2017). Lastly, flexibility is an essential characteristic of human cognition which is barely present in artificial systems. Examples of flexibility can be found in humor: A global consensus can be found on the idea that humour is based on the unification of two *a priori* distinct concepts into one single representation, which is called "*bisociation*" by (Koestler, 1964). In this definition, it is clear that humour relies on a flexibility of interpretation since it involves the evolution of a first understanding of a word or a situation into a new interpretation.

What conclusions can be drawn from these observations? It appears that, even when artificial systems outperform human capabilities, they are bad at reproducing elementary behaviors, which means, in other words, that we are still far from successfully passing the Turing test. It follows that current machine learning does not follow the same objectives as human cognition, but is excellent at the tasks it is biased toward.

The now well-known no-free-lunch theorem provides a theoretical evidence that such biases are unavoidable in the context of artificial learning. A complete learning theory has been developed in the statistical setting, aiming to characterize learning performance in terms of success rate (called *risk* in the context of supervised learning). However, this theory cannot be directly adapted to a larger class of problems, and in particular into a universal learning theory.

All these remarks together point out to the fact that getting more "human" results requires a completely different approach to learning. Such an approach should incorporate more flexibility to learning, and thus emphasize knowledge transfer.

In the context of this work, we define *knowledge transfer* as a process in which a previously acquired knowledge is used and slightly modified to fit into a context different from the original one. Such a modification operation is hidden everywhere

in human cognition: As human beings, we constantly adapt previous knowledge. For example, knowing how to play tarot is a valuable help for learning how to play rummy, even if both games have apparently nothing in common: different cards, different rules, different underlying principle (tarot being a trick-taking game and not rummy). The question of knowledge transfer appears under various forms in the artificial intelligence literature.

For **symbolic machine learning**, it takes the form of analogical reasoning the purpose of which is to draw links between apparently unrelated domains. The most famous example of an analogy is Rutherford's planetary model of the atom in which the nucleus plays the role of the sun and electrons the role of planets. Other applications are to be found in character strings (especially with the works of Douglas Hofstadter) or grammatical inference. Analogical reasoning is also a fundamental step of case-based reasoning, where a new problem is solved by analyzing previous cases and adapting their solutions to the situation of interest.

For **knowledge engineering**, the question of adapting knowledge can be seen in data linking and ontology alignment. Data linking is an operation which consists in finding identical elements inside two distinct data sources. For this operation, transferring knowledge from one data source to the other is unavoidable. When such an operation is performed on *conceptual networks* (ie. networks the nodes of which are concepts), ontology alignment can be used as a cognitive science tool to map concepts perceived by an individual into the concepts perceived by another individual (or by the same individual at another time).

In **machine learning**, the corresponding task is called transfer learning. Transfer is used when the distribution changes between learning and training. A change in the distribution is not commonly admitted in machine learning since data are often supposed to be *independent and identically distributed* (i.i.d.) but in practice this hypothesis does not always hold in most situations: For instance, it is natural that users' behavior in an online shopping platform vary depending on the time of the year. Transfer learning considers a previously learned concept (in particular a labeling function) and adapts it to the new data distribution.

Other machine learning tasks involving knowledge can be found, mostly involved in new environments of data management such as internet of things or distributed networks. In such environments, data can be generated in the form of streams and have to be managed on the fly and in real time, while storing past data is not possible due to memory limitations. Data streams require an adaptation to changes in the data distribution (called *concept drifts*) which can be either brutal or incremental.

Finally, new methods emerge in machine learning, inspired in particular by the constraints of distributed systems, where multiple systems have to collaborate without exchanging data in a direct way. For example, several clients of a same service may have access to different views on the same data because of confidentiality restriction but may have the possibility to exchange non-confidential information together. In such architectures, only partial information can be transmitted from one system to the other, and knowledge has to be transferred and adapted before being re-used.

1.2 Position and Contributions

In this thesis, we propose to examine the general scope of such problems and to determine what they have in common. In particular, we propose to answer three

fundamental questions that emerge when considering the question of knowledge transfer:

1. **How to transfer knowledge?**

   The question of the transfer method is commonly the most addressed question in the literature. It provides an algorithmic way to operate transfer in the problems of interest. Plenty of methods exist for every possible task. A unified view on such methods remains to be found yet.

2. **Is it possible to transfer knowledge?**

   In some cases, transfer is not direct or even not possible at all. Human beings know what knowledge can or cannot help them to solve a task. Extending such a capability to machine is a challenging task which requires a deep modeling of transfer.

3. **Is knowledge transfer necessarily successful?**

   Just because the machine is able to determine whether a transfer is possible does not mean that the transfer will reach good performance. On the contrary, the notion of *negative transfer* applies to situations where a transfer is possible but does not help the system.

If these questions are very general and not specific to this thesis, our main originality relies in the use of *Kolmogorov complexity* to address them. Intuitively, complexity measures how long the description of an object is. Complexity is a powerful tool to measure the amplitude of a transfer and seems to play a fundamental role in human cognition. Many links already exist between Kolmogorov complexity and artificial intelligence. Our approach is different in that it does not consider the restriction of complexity to probabilities, but remains as general as possible. This position makes it possible to consider broader issues and to go beyond a merely statistical vision of learning. Moreover, by considering data description and not data generation, our work offers a new perspective and opens up new issues.

In this direction, our main contributions are the following:

- We proposed a new framework to describe learning problems. This framework, which is based on Kolmogorov complexity, is data-oriented and not distribution-oriented. We have applied this framework to a large variety of problems, ranging from analogical reasoning to incremental learning, and we proposed models and algorithms for a practical use of this framework. We have also developed cognitive models based on our model, which indicates that our approach also succeeds in mimicking human intelligence.

- Based on this framework, we proposed a new criterion to measure task relatedness and transfer feasibility. The developed notion is close to the idea of hypothesis transfer and depends on the considered machine. This notion is at stake in an interpretation of the no-free-lunch theorem generalized to non-stationary environments.

- We proposed a formalism to describe the problem of collaborative clustering. The formalism is inspired by Ben David's analysis of clustering (Ben-David, Von Luxburg, and Pál, 2006) but incorporates additional notions, characteristic of a collaborative framework.

1.3 Outline

This thesis is organized in four parts, which correspond to use cases we have mentioned (analogy, transfer learning, incremental learning and collaborative clustering). Their order corresponds to a logical order for the understanding of the problem, but does not describe the chronology of the research itself. Each part is divided into small chapters in which a specific sub-problem is investigated. Besides, the main content of the thesis is preceded by a short preamble which introduces some fundamental concepts on which the thesis relies (chapter 2).

In part I, we first focus on **analogical reasoning** and consider it as a source problem to understand knowledge transfer. We first introduce the question of analogical reasoning and present the main principles and state of the art (chapter 3). We then present a canonical task in analogical reasoning: Hofstadter's problem. It consists in analogies on character strings and shares common characteristics with basically any other analogical problem. Hence it is a good candidate to understand analogies. In order to solve this task, we consider a description language and show that the analogies favored by humans have a minimal description code (chapter 4). Using this idea, we present in chapter 5 an analogical framework based on Kolmogorov complexity and develop some notation and elementary results which will be used in the subsequent parts. We eventually discuss an existing notion, proportional analogy, and consider a limit case, when terms of the analogy are elements of a Riemannian manifold. These results are presented in chapter 6.

The principle developed for analogical reasoning is then extended to the **transfer learning** task in part II. After a short presentation of the problem and the state of the art solutions (chapter 7), we propose an adaptation of the principle developed for analogical reasoning (chapter 8). This principle relies on the notion of *models*, ie. intermediate objects used to enhance data compression. We introduce a very simple class of models and present corresponding experimental results. Our system is designed to address one specific task and does not aim to offer generalization. In chapter 9, we discuss how the framework can be extended to inductive transfer (ie. to transfer with generalization). As an intermediate tool, we consider the question of transferability and show that all transfers are not feasible.

Both analogical reasoning and transfer learning are targeting some future task. In part III, we propose to follow the inverse direction and to look at the past by considering the question of **incremental learning**. In incremental learning, the system has to consider knowledge about the past to adapt to streaming data. We show in chapter 10 that our framework can be extended to describe incremental learning tasks. We will compare our framework to the state of the art algorithms and we will show that it can offer a tool to assess incremental learning as a whole. In chapter 11, we present an application to the topic modeling task in streaming environments: the approach we present, based on Latent Dirichlet Allocation and Adaptive Windowing, can be described by our framework and has various applications, in particular in online document analysis and recommendation. As an application, we also show that our incremental learning principle, merged with the string description proposed in chapter 4, can describe the phenomenon of U-shaped learning, well-known in cognitive sciences. The results, as well as a discussion on a generalization of this phenomenon, are proposed in chapter 12.

In part IV, we focus on the unsupervised setting and more specifically on the task of **collaborative clustering**. After a short introduction to the problem and to the existing methods and approaches (chapter 13), we show that our complexity-based approach can also apply and propose a new algorithm for collaborative clustering

(chapter 14). We finally discuss the limitations of collaboration in clustering by providing theoretical tools inspired by learning theory (chapter 15).

1.4 List of Publications

Al-Ghossein, Marie, Pierre-Alexandre Murena, Talel Abdessalem, Anthony Barré, and Antoine Cornuéjols (2018). "Adaptive collaborative topic modeling for online recommendation". In: *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, pp. 338–346.

Al-Ghossein, Marie, Pierre-Alexandre Murena, Antoine Cornuéjols, and Talel Abdessalem. "Online Learning with Reoccurring Drifts: The Perspective of Case-Based Reasoning". In: *ICCBR 2018* (), pp. 133–142.

Murena, Pierre-Alexandre, M Al Ghossein, T Abdessalem, and Antoine Cornuéjols (2018). "Adaptive window strategy for topic modeling in document streams". In: *International Joint Conference on Neural Networks*.

Murena, Pierre-Alexandre and Antoine Cornuéjols (2016). "Minimum Description Length Principle applied to structure adaptation for classification under concept drift". In: *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016*, pp. 2842–2849.

Murena, Pierre-Alexandre, Antoine Cornuéjols, and Jean-Louis Dessalles (2017). "Incremental learning with the minimum description length principle". In: *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, pp. 1908–1915.

Murena, Pierre Alexandre, Jean-Louis Dessalles, and Antoine Cornuéjols (2017). "A complexity based approach for solving Hofstadter's analogies". In: *CAW@ ICCBR-2017 Computational Analogy Workshop, at International Conference on Case Based Reasoning*.

Murena, Pierre-Alexandre, Jérémie Sublime, Basarab Matei, and Antoine Cornuéjols (2018). "An Information Theory based Approach to Multisource Clustering." In: *IJCAI*, pp. 2581–2587.

Sublime, Jérémie, Basarab Matei, and Pierre-Alexandre Murena (2017). "Analysis of the influence of diversity in collaborative and multi-view clustering". In: *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, pp. 4126–4133.

Chapter 2

Preamble: The Problem of Learning

In this chapter, we introduce the main objective of the thesis from a more technical point of view, by drawing links with traditional approaches of machine learning. The chapter is organized in three sections. In a first section, we go back to the well-known problem of Supervised learning and provide evidences for the use of Empirical Risk Minimization (ERM) with lights of Probably Approximately Correct (PAC) learning. In a second section, we introduce the Minimum Description Length (MDL) principle as a valid inductive principle and discuss the provable efficiency of this principle. In a third section, we move forward to new problems that differ from supervised learning and will be discussed in a more extensive way in this thesis.

2.1 Reminder on Supervised Learning

In this section, we present classical notions of supervised learning. Even if the first definitions we will give are general, we will focus more on the classification problem, and more particularly to binary classification.

2.1.1 Problem and Notations

We consider the following problem of learning. This problem will be called *supervised learning*.

Definition 1. *Supervised learning A supervised learning algorithm is given by:*

- *An **input space** $\mathcal{X}$,*
- *An **output space** $\mathcal{Y}$,*
- *An **hypothesis class** $\mathcal{H}$, ie. a set of functions $h : \mathcal{X} \to \mathcal{Y}$. Each function $h \in \mathcal{H}$ is called a **hypothesis**.*

*and is defined as a function $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{H}$, where $(\mathcal{X} \times \mathcal{Y})^*$ designates a list of arbitrary length of elements of $\mathcal{X} \times \mathcal{Y}$. The input $\mathcal{D} = \{(X_i, Y_i)\}_{i=1...N}$ (where $N > 0$) of the supervised learning algorithm is called **training dataset**.*

When the output space $\mathcal{Y}$ is continuous (for instance $\mathcal{Y} = \mathbb{R}$), the algorithm is called **regression algorithm**. When the output space $\mathcal{Y}$ is discrete, the algorithm is called **classification** algorithm. A special case of classification is obtained when $|\mathcal{Y}| = 2$. This case is called **binary classification**. It will be used later in this chapter, in particular in the PAC analysis.

With this definition only, the problem of supervised learning still remainis ill-posed. Given a fixed dataset, all hypotheses $h \in \mathcal{H}$ are equivalent, or, equivalently, there is no hierarchy among the learning algorithms.

A first intuition is that the hypothesis inferred by the learning algorithm must correctly predict the observed dataset. In practice, if we consider a function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ which measures the non-negative error between a predicted output $\hat{y} \in \mathcal{Y}$ and the ground truth output $y \in \mathcal{Y}$, it seems straightforward to minimize the average loss over the whole dataset (called the **empirical risk**). The corresponding algorithm

$$\mathcal{A}_{ERM}(\{(x_1, y_1), \ldots, (x_n, y_n)\}) = \arg\min_{h \in \mathcal{H}} \quad \frac{1}{N} \sum_{i=1}^{N} L(y_i, h(x_i)) \tag{2.1}$$

is called **Empirical Risk Minimization**. Intuitively, this algorithm corresponds to selecting the hypothesis that produces the lowest number of mistakes on the training dataset. As a consequence, it could be easily considered as the most reasonable algorithm. The *no-free-lunch theorem* will prove that it is not the case.

2.1.2 The No-Free-Lunch Theorem

The no-free-lunch theorem (Wolpert, 1996) is a fundamental result in learning theory, which states that all supervised learning algorithms are similarly efficient on average.

The exact formulation of the no-free-lunch (NFL) theorem requires a formalism, called Extended Bayesian Framework (EBF) and introduced in particular in (Wolpert, 1997). Since this formalism goes beyond the scope of this thesis, we will only present a quick overview of it. For more details, we refer the reader to the papers mentioned above. This formalism differs a bit from the one presented in Section 2.1.1 (which could be cast in terms of EBF), but it will be used only in this subsection devoted to the NFL theorem.

As exposed in (Wolpert, 1996), the EBF corresponds to a conventional application of probability theory to the space of quadruples $\{h, f, d, C\}$ defined as follows:

- f is called the target input-output relationship. It corresponds to a probability distribution over $\mathcal{X} \times \mathcal{Y}$.
- A training dataset d is generated from the target distribution f.
- h corresponds to a hypothesis, which is also a probability distribution over $\mathcal{X} \times \mathcal{Y}$. (Note that this definition is different from the one given above)
- C corresponds to a cost and measures the cost of a chosen hypothesis h to assess target f from training dataset d.

We would like to emphasize the fact that the four objects described here are random variables and not deterministic values. Moreover, it comes directly from these notations that a learning algorithm is given by $P(h|d)$. Finally, it is important to notice that, in the EBF, the input and output spaces are supposed to be discrete (but not necessarily finite). This hypothesis is not to be seen as restrictive since the computing machines are discrete.

Using this framework, the NFL theorem can be given as follows:

Theorem 1 (No-Free-Lunch Theorem (Wolpert, 1996))**.** *Let $E_i(.)$ designate the expected value evaluated using learning algorithm $\mathcal{A}_i$. Then for any two learning algorithms $P_1(h|d)$ and $P_2(h|d)$ independent of the sampling distribution:*

- *Uniformly averaged over all f:* $E_1(C|f,m) = E_2(C|f,m)$
- *Uniformly averaged over all f for any training set d:* $E_1(C|f,d) = E_2(C|f,d)$
- *Uniformly averaged over all $P(f)$:* $E_1(C|m) = E_2(C|m)$
- *Uniformly averaged over all $P(f)$, for any training set d:* $E_1(C|d) = E_2(C|d)$

As explained in (Wolpert, 2002) for instance, this theorem means that there are as many situations in which a first algorithm is superior to a second algorithm, as situations where the converse is true (where "superior" has to be read in terms of the criteria of Theorem 1). In other words, this means that there is no *universal* supervised learning algorithm.

The NFL theorem is important in the theory of supervised learning, since it proves that no learning algorithm can be considered as superior to the others in a universal way. In particular, the ERM algorithm presented above is not superior to any other possible algorithm.

We end up this section by exposing the reasons that are given to justify the use of the ERM.

2.1.3 Justification of Empirical Risk Minimization

We now present the formalism of learning theory that is employed to give formal justification of the ERM. In order to introduce the hypotheses made by learning theory, we first introduce an example.

Consider a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1...N}$ of elements in $\mathcal{X} \times \mathcal{Y}$. We consider the case where $\mathcal{X} = \mathbb{R}^d$, ie. the case where the input space is a vector space. Consider that the hypothesis class $\mathcal{H}$ is the set of *all* functions $\mathcal{X} \to \mathcal{Y}$. There exists infinitely-many hypotheses $h \in \mathcal{H}$ that describe the data correctly. For instance, the function that outputs y_i if the input is equal to x_i for $i \leq N$ and y_1 otherwise. By construction, this hypothesis is an empirical risk minimizer for the dataset $\mathcal{D}$. It can be used directly for *rote learning*, in other words when the goal of the learning process is to remember the correct labeling of the set of training inputs. It is intuitively clear that this hypothesis will have extremely low performances on a task that require generalization, ie. if the learning is asked to classify new points outside of the training set. The purpose of statistical learning theory is to provide an evidence that ERM can perform well on a task of generalization in given conditions.

Describing generalization requires that we rigorously define the expected limits of this generalization. A key assumption in this definition is the fact that the data points are chosen **independent and identically distributed** (i.i.d.). We consider the existence of a generating distribution P on $\mathcal{X} \times \mathcal{Y}$. Based on this distribution, we define the **risk** of an hypothesis $h \in \mathcal{H}$ as:

$$R(h) = \mathbb{E}_{(X,Y)\sim P}\left[L(Y, h(X))\right] \tag{2.2}$$

where $(X, Y) \sim P$ means that the random variables X and Y are drawn from distribution P. Based on this definition, the main goal of learning would be to minimize $R(h)$ over $h \in \mathcal{H}$, but this minimization is impossible when the distribution P is unknown. In practice, the learner has only access to the dataset $\mathcal{D}$, which is a sample of N points drawn i.i.d. from distribution P. The empirical risk defined above is thus an estimator of $R(h)$ and can be used as a proxy for the risk minimization. A fundamental question, however, is to determine how close a solution of ERM principle is to the solution of risk minimization.

Several properties can be studied, relative to these notions. For an extensive description of these problems, we refer the readers to (Vapnik, 1995). The results presented below can be found in (Shalev-Shwartz and Ben-David, 2014). We will only introduce one of the notions, called Probably Approximately Correct learnability (**PAC learnability**).

Definition 2. *PAC learnability A hypothesis class $\mathcal{H}$ is PAC learnable if there exists a function $m : (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property. For every $\epsilon, \delta \in (0,1)$, for every distribution P over $\mathcal{X}$ and for every labeling function $f : \mathcal{X} \to \{0,1\}$, if there exists $h^* \in \mathcal{H}$ such that $R(h^*) = 0$, then when running the algorithm on $m \geq m(\epsilon, \delta)$ i.i.d. samples generated by P and labeled by f, the algorithm returns a hypothesis h such that, with probability of at least $1 - \delta$, $R(h) \leq \epsilon$.*[1]

This notion of PAC learnabitiliy is a very important property to observe in order to achieve generalization. It means that with high probability, good generalization will be obtained provided that a learning algorithm with a large enough number of data. In the case of the ERM, PAC learnability is linked to other fundamental notions of learnability (uniform convergence, VC-dimension, which will not be described) via the Fundamental Theorem of Statistical Learning:

Theorem 2 (Fundamental Theorem of Statistical Learning, (Shalev-Shwartz and Ben-David, 2014), Theorem 6.7)**.** *Let $\mathcal{H}$ be a hypothesis class of functions from the input domain $\mathcal{X}$ to the output domain $\mathcal{Y} = \{0,1\}$, and let the loss function be the 0-1 loss. Then the following are equivalent:*

1. *$\mathcal{H}$ has the uniform convergence property.*
2. *$\mathcal{H}$ is PAC learnable.*
3. *Any ERM rule is a successful PAC learner for $\mathcal{H}$.*
4. *$\mathcal{H}$ has finite VC dimension.*

This theorem can be seen as a formal justification of the ERM principle. The idea is that, if the class of hypotheses is well-chosen and for enough data, the hypothesis learned by the ERM principle will correctly generalize on unobserved data. This does not mean that the ERM is an exception to the no-free-lunch theorem though, but that the fundamental theorem of statistical learning defines a restriction on which good generalization scores can be obtained.

2.1.4 Conclusion on Supervised Learning

In this section, we defined the problem of supervised learning and showed that, without further hypotheses, there is no valid hierarchy of solutions. We then considered a classical and intuitive inductive principle, the Empirical Risk Minimization, and showed two contexts in which this principle is valid:

1. *Rote learning*: The learner is supposed to be able to predict well when facing the training points again.
2. *Generalization*: The learner is supposed to be able to generalize well on data drawn from the same distribution.

However, the no-free-lunch theorem states that there necessarily exist some contexts in which the ERM principle does not work well.

[1] In this definition, the terms $R(h^*)$ and $R(h)$ both correspond to a risk for generating distribution P and labeling f.

2.2 Minimum Description Length and Minimum Message Length Principles

In this section, we present the Minimum Description Length principle as an alternative to the ERM principle for supervised learning. Even if this principle is more general, we will consider it in the context of supervised learning only. The ideas developed in this section can be found in (Grünwald, 2007).

2.2.1 Learning as Compression

The second inductive principle we introduce is called the Minimum Description Length principle and relies on a different understanding of learning. The idea is that an hypothesis that would perform well on a dataset relies on regularity in the data. The notion of regularity can be associated to a notion of *compression*.

We propose a first analysis to understand this intuition. This analysis is taken from (Vapnik, 1995) (Section 4.6).

Consider the dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1...N}$. Intuitively, learning is possible from these data if the string $Y = y_1, \ldots, y_N$ can be compressed based on $x_1, \ldots, x_N$. Note that, in the context of binary classification, we have $y_i \in \{0, 1\}$, which means that a description of Y can be done in N bits. Besides, when describing y_i from X, we suppose independence of the pairs (x_i, y_i) and, consequently, that the description of y_i can be done with help of x_i only (which also means that the values of x_j, for $j \neq i$, are not used in the description of y_i).

Consider now that the learner has access to a codebook C_b containing a number $|C_b|$ of functions $\mathcal{X} \to \mathcal{Y}$ (called *tables*). The codebook corresponds to an algorithmic version of the hypothesis space. If there exists a table T_o which perfectly predicts all labels y_i by $T_o(x_i)$, then it is sufficient to give the index of o of T_o in the codebook. Given X, having access to T_o then gives a description of Y. $\lceil \log_2 |C_b| \rceil$ are sufficient to designate the index o. This leads to a compression ratio of:

$$\kappa(T_0) = \frac{\lceil \log_2 |C_b| \rceil}{N} \tag{2.3}$$

In the case where there is no perfect table in the codebook, this compression ratio can be obtained with a more complex approach (see Eq. 4.24 in (Vapnik, 1995)).

The idea of the learning is that, if the coefficient $\kappa(T)$ is small, the transformation $\mathcal{X} \to \mathcal{Y}$ given by the table T models a regularity in the data. This is the ideal case to be reached in learning. If there is no such table, then it is not possible to learn.

It is noticeable that this compression coefficient is involved in the following theorem that justifies the assumption of "learning as compression":

Theorem 3 ((Vapnik, 1995), Theorem 4.3)**.** *If on a given structure of codebooks one compresses by a factor $\kappa(T)$ the description of an output vector Y of length $N > 6$ using a table T, then with probability at least $1 - \eta$ one can assert that the probability to commit an error by the table T is bounded by:*

$$R(T) < 2\left(\kappa(T)\ln 2 - \frac{\ln \eta}{N}\right) \tag{2.4}$$

2.2.2 Introducing Minimum Description Length Principle

The Minimum Description Length (MDL) and Minimum Message Length (MML) principles are alternative inductive principles that exploit the idea of learning as

compression in a direct way. The idea of these principles originates in the theory of Kolmogorov complexity, that will be exposed in more details in Section 5.2. Kolmogorov complexity of an object x is defined as the length of the shortest program on a prefix Universal Turing Machine (UTM) that outputs string x. A major problem of Kolmogorov complexity is that it is not computable. The inference methods based on Kolmogorov complexity and ignoring the uncomputability problem are often referred to as “ideal MDL". They are opposed to “practical MDL" and “MML" that consider less expressive restrictions of the programs.

A first version of MDL principle, called **Crude Two-Part MDL** is proposed by Jorma Rissanen (Rissanen, 1978). This version can be defined as follows: If $\mathcal{H}$ is a set of hypotheses, the best hypothesis $h \in \mathcal{H}$ to explain the data D is the one which minimizes the sum $L(h) + L(D|h)$ where $L(h)$ is the description length, in bits, of hypothesis h and $L(D|h)$ is the description length, in bits, of the data when encoded with the help of hypothesis h. The description length corresponds to the number of bits necessary to entirely describe an object, and is thus similar to Kolmogorov complexity and not computable.

In practice, this version of MDL is rarely used, and a “refined" version is usually preferred. The problem of the crude version is in the choice of a description length for the hypothesis which acts as a prior. Depending on this choice, the term $L(h)$ can be arbitrarily large or low, which makes the decision arbitrary. In the context of crude MDL, the hypotheses are necessarily encoded in ad-hoc ways.

A solution has been found to this problem that lead to the definition of a “refined MDL". This version considers the use of the whole class of hypotheses $\mathcal{H}$ for the encoding of the data. It relies on the choice of a family of codes called *universal coding*. Unlike crude MDL which is necessarily two-parts (ie. involves two terms: one for the model description and one for the data description), refined MDL is generally used in its one-part formulation[2] which considers only complexity term, the complexity of the observation relative to a class of models. This second version of MDL is the most popular in late literature on MDL

Despite the theoretical impact of this refined version, we will consider, in the scope of this thesis, the crude version of MDL, and more specifically the Minimum Message Length (MML) principle.

2.2.3 Introducing Minimum Message Length Principle

Independently from the historical development of MDL, Wallace and Boulton developed another principle called Minimum Message Length (MML) (Wallace and Boulton, 1968). This principle relies on the idea that each hypothesis in the hypothesis set can be associated to a two-part message: the first part provides a description of the hypothesis itself and the second part a description of the data encoded based on the hypothesis. In this, MML is closely related to the crude MDL.

MDL and MML differ mainly in their objectives (Baxter and Oliver, 1994). The purpose of MDL is to select a class of hypotheses but not a single hypothesis. As an example, in the problem of polynomial regression, MDL will focus on choosing the order of the polynomial used as a regressor, while MML will select both the order of the polynomial and its coefficients. Apart from this difference in their goals, MDL and MML are also fundamentally opposed in their conception of priors. Refined MDL is characterized by a lack of acceptation of subjective priors (see for instance page 56 in (Rissanen, 1989)).

[2]Since refined MDL makes use of classes of hypotheses rather than single hypotheses, an explicit and separated description of the hypothesis is not required.

For an extensive description of encodings used in MML for various problems of inference, we refer the reader to (Allison, 2018).

2.2.4 Conclusion on MDL and MML

In this section, we introduced the idea that learning can be related to information compression. This idea is at the core of two similar, but different, principles: Minimum Description Length principle and Minimum Message Length principle. Both principles mainly differ in the fact that MDL aims to select a restricted class of hypotheses while MML aims to select a specific hypothesis. However, both agree on a fundamental point that differs from ERM for instance: They focus on a description of data but they do not assume the existence of an underlying distribution from which data would be drawn. In this thesis, we will mainly rely on this characteristic to justify the use of these principles in the considered problems.

2.3 Drifting Away from Supervised Learning

In this section, we present several problems that differ from the supervised learning framework on several aspects and on which the ERM is not valid or cannot even be defined. A claim of this thesis is that crude MDL or MML are potential candidates to solve such problems.

2.3.1 Unsupervised Domain Adaptation

As explained in Section 2.1.3, the Empirical Risk Minimization principle comes with generalization guarantees. These guarantees rely on the assumption that all data (both training dataset and future test set) are identically distributed.

In some problems, this assumption does not hold. These learning scenarios are often referred to as **transfer learning**. Among all the problems of transfer, we focus on the question of **Unsupervised Domain Adaptation**. Consider an input set $\mathcal{X}$ and an output set $\mathcal{Y}$. Following (Pan and Yang, 2010), we introduce the following taxonomy:

Definition 3. *A **domain** $\mathcal{D} = \{\mathcal{X}, P\}$ is given by an input space $\mathcal{X}$ and a probability distribution P over $\mathcal{X}$. A **task** $\mathcal{T} = \{\mathcal{Y}, f(.)\}$ on domain $\mathcal{D}$ is defined by an output space $\mathcal{Y}$ and a labeling function $f : \mathcal{X} \to \mathcal{Y}$.*

In unsupervised domain adaptation, one considers two domains, the *source* domain and the *target* domain. We suppose that the source and target input (resp. output) spaces are the same. The generating distributions might change. In general, one considers that the source and target hypothesis classes are the same (denoted by $\mathcal{H}$). For instance, the space $\mathcal{X}$ could be the space of size $n \times n$ images, with the source domain distribution corresponding to photos and the target domain distribution corresponding to black and white drawings. A task on these two domains could be to discriminate illustrations of cats from illustrations of dogs. The corresponding labeling function f would differ in both domains.

What is the validity of the ERM principle in this case? Several analyses answer this question by trying to find a common hypothesis for the source and for the target. Such methods propose PAC bounds for the unsupervised domain adaptation, which aim to justify to resort to the ERM. Among these studies, we refer the interested reader to the seminal article (Ben-David, Blitzer, Crammer, Kulesza, Pereira, and

Vaughan, 2010), the main result of which will be discussed in Chapter 7 (Theorem 7). These results often include a penalty relative to the divergence between the two distributions, called task discrepancy.

A main weakness of the existing PAC approaches to unsupervised domain adaptation is the assumption that data distributions have to be similar. This assumption is in particular a prerequisite on choosing a common hypothesis for the source and for the target.

2.3.2 Unsupervised Learning

The next problem we propose to discuss in this preamble is **clustering**, which is an unsupervised learning task. A clustering algorithm can be defined as follows:

Definition 4 (Clustering algorithm)**.** *Given an input space $\mathcal{X}$, a **clustering algorithm** is a function $\mathcal{A} : \mathcal{X}^* \to \mathbb{N}^*$ such that for any input $\mathcal{D} \in \mathcal{X}^n$, the corresponding output $\mathcal{A}(\mathcal{D})$ also has length n.*

With this definition, the task of clustering corresponds to a task of labeling. It is mostly admitted that the chosen labels must satisfy some properties, which intuitively express that “similar points must be attributed the same labels". Clustering raises new issues compared to supervised learning.

The first problem is related to the unsupervised nature of the task. In classification, the presence of labels on the training dataset makes it possible to measure the prediction skills of the chosen algorithm, which is the basis of the definition of the ERM principle: on a given dataset, the principle chooses the hypothesis making the minimal number of mistakes. Since the notion of error cannot be extended to an unsupervised setting, such a direct approach cannot be used for clustering. The absence of a universal quality criterion implies indeed the impossibility theorem (Kleinberg, 2003), which states that no clustering algorithms can satisfy the following three criteria: *scale invariance* (the clusters remain the sames if the scale is changed), *consistency* (the clusters remain the same if the intra-class distances are reduced and the inter-class distances are increased) and *richness* (the algorithm can produce any possible partition).

The second problem is the non-extensive nature of clustering. In classification, considering generalization makes sense, while it is not as direct in clustering. If some algorithms like K-means make generalization possible[3], some do not: This is for instance the case with DBScan (Ester, Kriegel, Sander, and Xu, 1996) or OPTICS (Ankerst, Breunig, Kriegel, and Sander, 1999). This makes a PAC analysis less direct and more restrictive: For instance, the study of stability proposed in (Ben-David, Von Luxburg, and Pál, 2006) relies on an extended definition of clustering that fits more the PAC framework but does not apply to all classes of clustering algorithms.

2.3.3 Analogies

The last problem we will tackle in this thesis is the question of analogies, which has been a prominent question in symbolic learning in the last decades (Prade and Richard, 2014). The problem of analogy originally comes from cognitive sciences and psychology, but has been studied in computer science on the following form:

[3] K-means algorithm infers virtual points in $\mathcal{X}$, called prototypes, and attaches points to their closest prototype. Once the prototypes are build, it is possible to generalize to any point in $\mathcal{X} \setminus \mathcal{D}$ by attaching them to their closest prototypes.

Definition 5. *Consider a source domain $\mathcal{D}_S = \mathcal{X}_S \times \mathcal{Y}_S$ and a target domain $\mathcal{D}_T = \mathcal{X}_T \times \mathcal{Y}_T$. An **analogy question** is a function $\mathcal{X}_S \times \mathcal{Y}_S \times \mathcal{X}_T \to \mathcal{Y}_T$.*

Even if this definition is given in its more general form, it is rarely used as such: several particular cases are observed in practice in the researches on analogies:

- The domains $\mathcal{D}_S$ and $\mathcal{D}_T$ are often not known and the definition has to be modified in order to include a definition of objects by predicates (Holyoak and Thagard, 1989).
- When the domains are known, they are often considered as identical, and most studies consider that $\mathcal{X} = \mathcal{Y}$ (see for instance (Miclet, Bayoudh, and Delhay, 2008)).

Considering the equality of the domains ($\mathcal{D}_S = \mathcal{D}_T$), a direct link can be seen between analogy and supervised learning. Given observations in $\mathcal{X} \times \mathcal{Y}$, analogy infers a transformation $h : \mathcal{X} \times \mathcal{Y}$ and applies it to a new input $x_T \in \mathcal{X}$. It is tempting then to apply the ERM principle to the problem, which would support analogy with the theoretical justifications discussed above. This is not the case though, since the training dataset contains only one observation, making the theorems of statistical learning inoperative: in statistical learning, induction is ensured by an inference made on a large number of observations, with support of the theorem of large numbers.

Another formal attempt has been proposed with the theory of **proportional analogies** (Miclet, Bayoudh, and Delhay, 2008). This theory provides a strict logical definition of valid analogies seen as a Boolean relation, based on a restriction of the set of all relations. Further discussion will be given on this framework later in the thesis (in particular in Chapter 3). Recent results have shown that proportional analogies can be extended for prediction but have a bias toward linear functions (Couceiro, Hug, Prade, and Richard, 2018).

When trying to answer the question “what makes a good analogy?", several difficulties arise (fairly similar to some of the problems encountered with clustering). In particular, it is necessary to define a notion of *similarity* in the input space $\mathcal{X}$ and in the output space $\mathcal{Y}$. In addition, analogy requires to estimate a mapping between the transformations on $\mathcal{X}$ and the transformations on $\mathcal{Y}$.

2.4 Conclusion

In this preliminary chapter, we proposed a short presentation of the question of induction and of its inherent difficulties. In a first part, we analyzed the problem of supervised learning and the necessity to choose an inductive principle to solve this problem. If this inductive principle can be chosen arbitrarily, an expert knowledge is necessary to choose a correct principle in the context of a specific task. The principle of Empirical Risk Minimization is proposed as a valid inductive principle in the context of supervised learning with theoretical guarantees for generalization in a context where all data (present and future) are identically distributed. Moreover, these guarantees also give directions for the choice of the class of hypotheses $\mathcal{H}$. This first part opens a general question: **What is the right approach to select an inductive principle over others?** The choice of the ERM as an inductive principle is motivated by the choice of an application framework and theoretical justifications on it.

We introduced a second inductive principle, called Minimum Message Length principle, and exposed some justifications that have been proposed for it. The purpose of this thesis is to use this principle in the context of knowledge transfer in various tasks, some of which have been introduced in Section 2.3 above.

The main contribution of this thesis is to test the experimental validity of the MML principle on these transfer tasks. In the proposed examples, the classes of models are chosen based on some expert domain knowledge, but independently on the application domain. Note that the purpose is not to propose a theoretical justification of a choice of a class over another (such as given by the VC-theory for the ERM). These justifications are important though and have to be done in future works. In Chapter 9, we investigate some theoretical tracks that might lead to a formal validation of the principle in the context of transfer.

Part I

A Fundamental Problem: Analogical Reasoning

Chapter 3

Introduction to Analogical Reasoning

The term *analogical reasoning* designates any kind of reasoning which draws parallels between two distinct and *a priori* unrelated situations. The fundamental idea behind this form of reasoning can be summed up in the simple conjecture that, if two situations are alike in some respects, they should be alike in some other respects. Considering such an informal definition, it becomes clear that the challenges faced by the analogical reasoning literature are to define under which conditions this conjecture is valid and what "alike" means.

The ability to produce and understand analogies is a fundamental ability shared by human beings in such a way that it is commonly used as a standard to measure human intelligence: IQ tests make extensive use of analogies. In everyday life, analogies are involved in various activities, such as metaphors, humor, or even scientific method.

In this chapter, we present a general overview on the question of analogy, its use in human cognition, the philosophical approaches to assess it and computational models. The chapter is structured as follows. In Section 3.1, we present cognitive and psychological evidences of conscious and unconscious modes of analogies for human beings. In Section 3.2, we describe formal models of analogies. Finally, Section 3.3 proposes some applications of computational analogies.

3.1 Analogies in Human Cognition

In this section, we briefly introduce the fundamental role played by analogical reasoning and similar transfer phenomena in human cognition. The purpose of this section is not to provide an extensive description of the state of the art, but to support the idea that analogies are inherent to the human mind and are a fundamental gap in the direction of a human-like artificial intelligence.

3.1.1 General Presentation of Analogy and Cognition

The importance of analogy in human cognition has been studied for decades now and is closely related to the fundamental ability to extract the similarities between two concepts or situations and to transfer the characteristics of the one to the other.

This skill appears at the lowest degree of cognition, for instance in the fundamental task of perception. Children learn to discriminate between thousands of categories in their environment, but are also able to tell the similarities from the differences. For instance, they will find many similarities between a dog and a cat, or even between a dog and another dog, but will also be able to understand them as

different. They are also able to apply a “transformation" from one object to the other. For instance, they will get that the transformation from a standard dog to a red dog can be applied to any other object that have similar characteristics, for instance a cat. More impressively, perception of an object can depend on the context: an animal in a picture will not be the same as a real animal, but children are able to map these two radically different entities while being able to tell which one is real. This ability is characteristic of analogical reasoning: An interpretation, given by Structure Mapping Theory (described below), suggests that the child extracts common descriptive information and builds a mapping between the real animal and the drawn animal.

The case of drawings is even more interesting, in the sense that sensitivity to the “style" is a basic ability. A drawn dog will not be the same depending on the cartoonist, which does not affect the perception. The transfer from one style to another is possible, and is even considered as a required competence in many artistic domains (for instance being able to paint in the style of a famous painter, to compose in the style of a composer or to write in the style of a writer). Melanie Mitchell proposes the example of music in (Mitchell, 2001): *Any two pieces by Mozart are superficially very different, but at some level we perceive a common essence. Likewise, the Beatles rendition of Hey Jude and the version you might hear in the supermarket have little in common in terms of instrumentation, tempo, vocals, and other readily apparent musical features, but we can easily recognize it as the same song.*

A domain where analogy is hidden everywhere (but most of the time in an unconscious manner) is natural language. In order to communicate ideas or concepts, we often rely on comparisons and analogies. For instance, it is not rare to hear expressions like “He is the Mozart of painting." while this expression is a priori very unclear. Does it mean that this painter is also a musician? Died at the same age as Mozart did? We intuitively understand what is meant here: Mozart, in the context of music, is often considered as a prodigy; transferring this characterization to the new context (painting), we understand that the painter of interest has to be considered as a prodigy as well. This example displays two of the most fundamental components of analogical reasoning: on the one hand, it shows that we are able to transfer unified descriptions from one domain to the other; on the other hand, we are even able to determine the domain when it is not defined. No indication was given that Mozart had to be considered in the scope of music, but it is natural. Mimicking this ability will be one of the objectives of connectionist models such as ACME or LISA (Section 3.2.4).

Another example, proposed by (Mitchell, 2001) concerns the sentences like “The same thing happened to me!". How to define the notion of sameness? To what extent did the *exact same thing* happen? Some details in the story can change without any incidence on the similarity: a different location, different characters, different weather... However, we are still able to determine that the most important facts are identical, and we are also able to predict how the differences between the two situations might lead to a different conclusion. This is typical of analogical transfer.

Finally, we would like to insist on the prominent role played by comparisons and metaphors in communication. Among the most frequently used examples to illustrate analogical reasoning, one plays a very special role: “A battery is like a reservoir" (more discussion on this example will be provided in Section 3.2.3. This formulation is typical of comparisons as they can be formulated in everyday life. From a general point of view, the link between metaphor and analogy is very strong, and in particular metaphor can be seen as a special case of analogy. It has been studied in language, in particular in articles such as (Steen, 2008) which attempts to solve an ambiguity of metaphor seen as an analogical process.

3.1.2 Evidences of Syntactic Priming

As a complement to the general introduction on analogy in human cognition, we propose now a brief presentation of the problem of syntactic priming, which is some kind of unconscious analogy observed in language production and interpretation.

Syntactic priming is a well-known phenomenon observed in cognitive sciences which leads to reusing previously encountered structures in sentence generation or comprehension. Such phenomenon is observed in several grammatical choices, such as active vs active ("Cats eat mice" vs "Mice are eaten by cats"), dative formation ("He bought the girl a book" vs "He bought a book for the girl") or relative clause attachment ambiguity ("I like the friend of my sister who plays the violin": who plays the violin?). In the first two problems, it is observed that encountering one of the forms favors the reuse of the same form; In the case of relative clause attachment, the disambiguation is primed by a former non-ambiguous relative clause. This transfer is typically unconscious.

First studies of the phenomenon emerged in the 1980s with pioneering works on the repetition of similar syntactic forms across successive utterances (Bock, 1986). For instance, some studies observe that, in case a speaker used a passive in a recent sentence, he is more likely to use one in future sentences (Weiner and Labov, 1983). These observations were not sufficient to conclude on the importance of syntactic priming, since they relied on corpus data and did not reflect a preference between two alternatives. Other phenomenons could be at stake here: facility of repetition in either lexical, thematical or metrical aspects. (Bock, 1986) provides the first real evidence of syntactic priming in language production. The existence of syntactic priming was shown in multiple languages (Hartsuiker and Kolk, 1998), and for both written (Branigan, Pickering, and Cleland, 1999) and oral productions (Potter and Lombardi, 1998).

A major breakthrough in the domain is the evidence of cross-linguistic priming effects. (Loebell and Bock, 2003) have shown the existence of a priming effect between German and English datives (*He bought the girl a book* vs *He bought a book for the girl*, and German equivalent). However, the results show no priming in the case of passive forms. In the same direction, (Hartsuiker, Pickering, and Veltkamp, 2004) showed cross-linguistic priming between comprehension and production in English and Spanish, two languages that are only weakly related. Following (Kantola and Gompel, 2011), there is actually no difference between inter-language and within-language syntactic priming. The authors propose experiments to validate or discard one of the two leading theories of syntactic priming: shared-syntax account, which assumes that the structures in languages are represented in a same and only mental space (Hartsuiker, Pickering, and Veltkamp, 2004), and language-specific account supported in particular by (De Bot, 2000) and (Ullman, 2001). Their results tend to validate the shared-syntax model, showing that there is no distinction between inter- and intra-language priming, neither on the existence of priming nor on its relative impact.

More surprisingly, these studies have extended outside the domain of language and evidences of syntactic priming from various domains such as music (Patel, 2003) or mathematics (Scheepers, Sturt, Martin, Myachykov, Teevan, and Viskupova, 2011) to language have appeared. These generalizations to multiple domains has been studied by (Cavey and Hartsuiker, 2016) which proposes general experiments based on several source domain in order to support the existence of structural priming in general domains and discard the classical domain-specific models of syntactic processing.

3.2 Formal Models of Analogical Reasoning

In this section, we present existing models of analogy. We focus mainly on the question of representation rather than the algorithmic resolution of analogical problems.

3.2.1 Logic Description

The first model that can be given for analogy is based on logic and exploits the intuitive idea of reasoning by analogy. Reasoning by analogy is made up of two steps:

1. A source problem S and a target problem T share identical properties.
2. If source problem S satisfies another property, then T *should* also satisfy it.

For instance, if two computers and have the same configuration (same processor, same memory, same operating system, same GPU, same peripherals...), it can be assumed that these two computers also have the same price. This reasoning is *inductive*, since its conclusion is not guaranteed to be true. For instance, in this example, it could be inferred by analogy that the two computers have the same brand, which is not necessarily true.

This principle has been formalized by (Davies and Russell, 1987). If P designates a property or a set of properties and Q designates a property, then the analogical reasoning process can be formalized as follows:

$$\begin{array}{l} P(S) \wedge P(T) \\ Q(S) \\ \hline \therefore \quad Q(T) \end{array} \tag{3.1}$$

As pointed out by the authors, this reasoning is obviously not deductive, and the conclusion cannot be inferred from the predicate $P(S) \wedge P(T) \wedge Q(S)$. The authors focus then on the justification problem, hence in this case the question of transferability: What would be a criterion that would make 3.1 a deductive inference? An obvious solution for this problem is the generalization criterion "$\forall x, P(x) \Rightarrow Q(x)$". Not only this criterion is very strong (it implies that the analogy holds *for any* target), but it is also non-satisfactory from the perspective of logic. Indeed, assuming it to be true, the initial criterion $P(S) \wedge Q(S)$ becomes useless, since the reasoning

$$\begin{array}{l} P(T) \\ \forall x, P(x) \Rightarrow Q(x) \\ \hline \therefore \quad Q(T) \end{array} \tag{3.2}$$

holds true. Therefore, adding this generalization criterion makes any knowledge on the source useless.

In order to give an importance to the source knowledge, an alternative criterion is suggested: the *determination rule*:

$$(\forall x, P(x) \Rightarrow Q(x)) \vee (\forall x, P(x) \Rightarrow \neg Q(x)) \tag{3.3}$$

This criterion expresses the fact that the satisfaction of Q is determined by the satisfaction of P, but not if the values are similar or different. The criterion on the source $P(S) \wedge Q(S)$ is thus required to overcome this ambiguity.

3.2.2 Analogical Proportion

Another model based on logic is the model of *analogical proportion*. This model is a formalization of principles already exposed by Aristotle that emerged in the late 1990s in parallel in the works of François Yvon and Yves Lepage (Lepage, 2000). An **analogical proportion** is then defined as a 4-ary relation A such that the three following properties are verified for any a, b, c, d:

- **Reflexivity:** $A(a, b, a, b)$ always holds
- **Symmetry:** $A(a, b, c, d) \Rightarrow A(c, d, a, b)$
- **Central permutation:** $A(a, b, c, d) \Rightarrow A(a, c, b, d)$

A proportion A that satisfies these three axioms is shown by (Lepage, 2004) to satisfy the following equivalence: For any a, b, c, d, $A(a, b, c, d)$ is equivalent to:

- **Inversion of ratios:** $A(b, a, d, c)$
- **Exchange of the extremes:** $A(d, b, c, a)$
- **Symmetry of reading:** $A(d, c, b, a)$

Analogical proportions can be applied to various domains, as exposed for instance by (Miclet, Bayoudh, and Delhay, 2008): analogies on Booleans (Miclet and Prade, 2009), on sets (Lepage, 2003), on character strings, on vectors... It is important to notice however that the three axioms of analogical proportions are in general not enough to define a *unique* proportion. However, in the context of Boolean analogies, (Prade and Richard, 2013) demonstrate that there is one unique logical proportion that satisfies the three axioms. In Chapter 6, we will show that there exists infinitely many analogical proportions on a vector space.

3.2.3 Structure Mapping Theory

Structure Mapping Theory (SMT), developed by (Gentner, 1983), is a particularly influential theory which is applied today in most cognitive approaches of analogical reasoning. Before we present SMT and its implementation, Structure Mapping Engine (Falkenhainer, Forbus, and Gentner, 1989), we first introduce Winston's theory, which is a precursor of SMT.

In (Winston, 1977), Winston introduces the idea of "transfer frames". Considering the learning process in the situation of an exchange of information from a teacher to a student, he notices the existence of two frames: the source frame, corresponding to the information that has to be transmitted, and the destination frame, that is the element onto which the information has to be transmitted. For instance, in the sentence "Robbie is like a fox", the source frame is "fox" and the destination frame is "Robbie". In order to learn, the student has to build a "transfer frame", which is an intermediate frame used to *filter* the useful information from the source frame applicable to the destination. This transfer frame can be seen as a mapping from the source to the destination, based mainly on the most pertinent properties of the source, the information contained in the target, and the context. In (Winston, 1980), this mapping is applied to the problem of analogical reasoning, which constitutes a first step toward mapping for analogical reasoning. However, this methodology differs from later theories (and in particular from SMT) on the way the mapping is built, since it considers mapping on attributes and in high order relations between objects.

On the contrary, the key idea of SMT is to link source and target domains on high order relations only, ignoring the possible mappings between attributes. This difference is motivated by the *relation-matching principle*, which states, informally, that superficial characteristics of objects are less important in analogies than structural characteristics. For instance, in the famous comparison "a battery is like a reservoir", the main characteristics that is transferred is the storage capacity, but not specific characteristics such as mass or color, nor characteristics of the storage (electric for the battery, volume for the reservoir). The second principle underlying SMT is the *systematicity principle*, which states that mappings of systematic structures has to be preferred over mappings of individual relations.

The general principle of SMT was first implemented in a LISP program called Structure Mapping Engine (SME) (Falkenhainer, Forbus, and Gentner, 1989). This program is the source of many analogical systems and has been widely used to support cognitive theories. However, it is not perfect and has been criticized in particular by (Chalmers, French, and Hofstadter, 1992) for its requirement for complex hand-coded representations. The critic targets the inherent necessity of a human coding of the representations of involved objects which is necessarily biased to the task. From this point of view, most of the work on analogy is done by humans rather than SME itself.

Many variants have been proposed, that are either direct evolutions of SME or explore new directions. Among other propositions, we can cite I-SME (Forbus, Ferguson, and Gentner, 1994), an incremental variant of SME, SEQL (Kuehne, Forbus, Gentner, and Quinn, 2000), which uses SME in order to infer general schemas, or the Latent Relation Mapping Engine (Turney, 2008) which attempts to solve the problem raised by (Chalmers, French, and Hofstadter, 1992) by automatically discovering semantic relations between the words from a corpus of texts.

Other methods based on structural mappings have been developed but will not be described here. For more details, we refer the reader to the surveys (Holyoak, 2005; Gentner and Forbus, 2011).

3.2.4 Connectionist Models

Structure Mapping is based on a representation of the world involving relational models of objects, relations and predicates. Some efforts have been done to incorporate connectist models into analogical reasoning.

The first connectionist model, ACME (Holyoak and Thagard, 1989), supports the idea that structural consistency required by SMT is not the only constraint that must be involved in analogical reasoning. The proposed approach relies on the idea of multiple constraint satisfaction: structural consistency, semantic similarity and pragmatic centrality (favoring correspondences in the mapping that are pragmatically important to the agent). ACME algorithm uses these three constraints to build a network the nodes of which are potential element mappings and vertices represent instantiation of the general constraints. Once the network has been built, the weights of the vertices are updated according to the constraints.

The LISA model (Hummel and Holyoak, 1997) follows the same multiple constraint satisfaction paradigm, but in a rather different way, since it adds, to the three constraints of ACME, a set of constraints relative to cognitive and neuronal plausibility. In particular, in contrast with SME and ACME, LISA relies on the use of a working memory and a long term memory which can interact during the mapping. The LISA model remains one of the most ambitious and influential connecionist models of analogy and is the support of cognitive and psychological studies.

3.2.5 Copycat and Metacat

The Copycat project (Hofstadter and Mitchell, 1995; Hofstadter, 1984) is a completely different approach to analogy which aims to include the very specific characteristics of human analogical reasoning into artificial analogy-making on a restricted world. Copycat solves analogical problems of the form "**ABC** is to **ABD** as **IJK** is to **?**". A major advantage of this restricted problem is that these alphabetic analogies capture the inherent difficulties of analogical reasoning while being limited to a very simple micro-world. A more precise description of Hofstadter's analogies will be provided in next chapter.

The Copycat program is based on the principle of parallel terraced scan (Hofstadter, 1995), a parallel multi-agent method inspired by ant colonies in particular, in which all possible hypotheses are explored in parallel with resources depending on their potential. In Copycat, these agents are teams of *codelets* and their allocated resource is the time available to discover a structure. The role of codelets is to code a structure perceived in the *workspace* (ie. the raw representation of the analogical problem) and they can rely on long-term knowledge stored in a network of concepts called *slipnet*. The nodes in the slipnet are associated to a dynamic activation value, measuring the relevance of the concept in the current analogy. These activation values can "slip" from one node to its conceptual neighbors.

Copycat is often described as a hybrid system (French, 2002), sharing characteristics of SME (the idea of converging to a unified encoding of source and target) and connectionist approaches (the slipnet), but this categorization is denied by the authors who judge that: "Copycat's architecture is neither symbolic nor connectionist, nor a hybrid of the two; rather, the program has a novel type of architecture situated somewhere in between these extremes" (Hofstadter and Mitchell, 1995).

Among other variants of Copycat, the Metacat program (Marshall, 2002) focuses on giving memory to the estimation process in order to make the search faster and to improve the diversity of found solutions. Such an architecture is called *self-watching*.

3.2.6 Analogies and Kolmogorov Complexity

A last influential model of analogical reasoning has been proposed by (Cornuéjols and Ales-Bianchetti, 1998) with a simple and effective approach based on algorithmic information theory (Li and Vitányi, 2008). The key notion at stake in this approach is *Kolmogorov complexity*, a measure of the information contained inside an object. Complexity of an object x is defined as the length of the shortest program on a universal Turing machine that can generate object x. For instance, the binary sequence 1111...11 (1 repeated 10^6 times) is long but very simple since it can be generated with a very short program (`for i=1..1e6: print(1)`) and does not require that every single bit be described individually. On the contrary, some objects can be shown to require very long descriptions.

One of the uses of Kolmogorov complexity are the Minimum Description Length (MDL) and Minimum Message Length (MML) principles, two inductive principles that formalize the classical idea of Ockham's razor. MML principle suggests that, when several hypotheses are possible to explain an observation, the "best" hypothesis is the one that minimizes the description length of the hypothesis and of the observation based on the hypothesis.

In the context of analogical reasoning, (Cornuéjols and Ales-Bianchetti, 1998) suggests that the transfer and mapping should not been done at the symbolic level (ie. directly on the involved objects) but at the level of intermediary structures called

models. For instance, in the case of Hofstadter's analogies, the model can be a description of the structure of objects. If $K(.)$ designates the complexity of an object, the proposed approach is to find the optimal source model M_S and target model model M_T that minimizes:

$$K(M_S) + K(X_S|M_S) + K(Y_S|M_S) + K(M_T|M_S) + K(X_T|M_T)$$

where the purpose is to solve the analogical equation $X_S : Y_S :: X_T : y$ with y being the unknown value.

Another model (Bayoudh, Prade, and Richard, 2012) is based on Kolmogorov complexity, but exploits its properties in a completely different way. The chosen approach is based on the theory of proportional analogy. From this point of view, the analogy $A : B :: C : D$ is equivalent to having equal distances between involved objects: $d(A, B) = d(C, D)$ and $d(A, C) = d(B, D)$. The idea here is to use the pseudo-distance offered by conditional complexity: $d(A, B) = K(B|A)$. In the proposed experimental validation, the complexity is evaluated using Shannon-Fano coding (ie. $K(x|p) = -\log p(x)$) and the probability is estimated from the frequencies. The same approach had already been proposed by (Prade and Richard, 2009) to solve analogies between general concepts. In that article, the complexity is estimated using Google information distance (Cilibrasi and Vitanyi, 2006) and the complexity of a concept is supposed to be related to the mass function of this concept on Google search engine.

3.3 Applications of Analogical Reasoning

In this section, we propose a couple of applications of analogical reasoning in artificial intelligence.

3.3.1 Word Embedding and Analogical Proportion

In vector spaces, the axioms of proportional analogy (see Section 3.2.2) are satisfied by a very simple proportion called the parallelogram rule: If X is a vector space and $(a, b, c, d) \in X^4$, the proportion $A : X^4 \mapsto \mathbb{B}$, such that $A(a, b, c, d) = 1$ if and only if $d = c + b - a$, defines an analogical proportion. This condition corresponds to having a, b, c, d making a parallelogram.

Parallelogram rule was first used by (Rumelhart and Abrahamson, 1973) in the restricted problem of analogies between animals (for instance "*mouse* is to *raccoon* as *cow* is to *?*"). The main hypothesis made by the authors follows the conclusion of (Henley, 1969) that the memory structure can be embedded in a vector space structure and that the judgment of similarity is inversely related to the distance in this space. In order to solve an analogical equation, the authors suggest to project the entities into the vector space and to choose the solution that is the closest to $c + b - a$. If $X_1, \ldots, X_n$ designate the vectors representing the n entities in the chosen vector space, and $d(.,.)$ corresponds to a distance on the vector space, then the chosen solution for the analogical equation $a : b :: c : x$ is:

$$x^* = \arg\min_{1 \leq i \leq n} d(X_i, c + b - a) \tag{3.4}$$

In practice, more sophisticated methods such as SME are usually preferred to the parallelogram rule, but the recent emergence of embedding techniques resurrected

the method. Word embedding technique, developed in particular with the `Word2Vec` method (Mikolov, Chen, Corrado, and Dean, 2013; Mikolov, Yih, and Zweig, 2013). In this technique, words are projected in a vector space that is learned according to a vast corpus of texts. The quality of embedding is usually tested on the task of solving analogies such as "man is to woman as king is to queen" using he parallelogram rule. Despite some critics on this technique (Drozd, Gladkova, and Matsuoka, 2016), it has also been used in other domains, such as visual object categorization (Hwang, Grauman, and Sha, 2013).

3.3.2 Linguistic Analogies

A very popular introducing analogies is Natural Language Processing. The key idea in this applicative domain is that, in morphology, in phonology or even in translation, one single example is necessary to provide a full knowledge of the domain. Consider for instance the problem of conjugation. Knowing the conjugation rule for one verb can be used by analogy to infer the conjugation of any other *similar* verb: If we observe the transformation *to make* $\longrightarrow$ *makes*, then we can assume that the solution of *to eat* $\longrightarrow$ *?* will be "*eats*".

The (apparently) simplest problem, which is also the problem presented as an example above, is the problem of **morphological inflection**, that is involved in particular in declension or conjugation. This problem involves analogies on character strings but is one example of applications that cannot be handled by Copycat, since almost no structure can be found. In order to solve this problem, two visions have been developed in parallel, relying on different understandings of proportional analogy. The first solution, introduced in particular by the seminal research of (Lepage, 1998), relies on proportions in letter counts and positions. The second solution, proposed in particular in (Yvon, 2003; Stroppa and Yvon, 2005), relies on the finding of specific factorizations. In particular, the *alea* solver (Langlais, Yvon, and Zweigenbaum, 2009) finds such a factorization by building an automaton and randomly shuffling the inputs.

A first paradigm for **automatic translation** was proposed by (Nagao, 1984) and consists in solving an analogical equation with observed sentences. For instance, consider the following example of French to English translation: If one wants to translate into English the French sentence "Il peut le faire", it is possible to consider the already available translation "Je veux lire ce livre" $\longrightarrow$ "I want to read this book" and *adapt* it in order to obtain the desired solution: "He can make it". Despite its apparent elegance, this model is not directly computable and requires a complete knowledge representation that cannot be a priori obtained automatically. A solution has been proposed by (Lepage and Denoual, 2005), based on proportional analogy. The axioms of proportional analogy make the system computable and knowledge-light.

A last application that has been proposed concerns pronunciation. This problem can be reformulated as the transcription of a word in a given language (in particular English) into International Phonetic Alphabet. The use of analogy to solve this task has been proposed by (Pirrelli and Federici, 1994).

3.3.3 Machine Learning Applications

As a last application of analogical reasoning, we propose to present the links between analogical reasoning and machine learning.

A first analogical model for machine learning is simply provided by k nearest neighbors (NN) algorithm, which is often considered as characteristic of *lazy learners*. In general, kNN can be seen as a perfect example of analogical reasoning, where the principle at stake is that “similar points have similar labels".

More complex applications have been considered yet. In general, two cases are considered: classification of Boolean vectors and classification of numerical vectors.

Among classifiers on Boolean vectors, (Miclet, Bayoudh, and Delhay, 2008) relies on a measure of “analogical dissimilarity" between four objects, which measures how far the quadruple is from being in analogical proportion. In order to classify an object d, the algorithm evaluates the analogical dissimilarity of the quadruple (a, b, c, d) for all available triples (a, b, c) and selects the solution according to a ranking by decreasing dissimilarity. Following a similar idea, analogical inference states that if $a : b :: c : d$, then $f(a) : f(b) :: f(c) : f(d)$ for a function f. In (Couceiro, Hug, Prade, and Richard, 2018), the authors show that analogical inference provides non-zero error under the condition that the function f is affine, but relax this result by providing error bounds depending on the distance of f to the set of affine functions, in the case where f is approximately affine. This affine nature is closely related to the nature of proportional analogy and would be different in other analogical frameworks.

Less work has been done on analogical classifiers for numerical vectors. A first proposition was done by (Prade, Richard, and Yao, 2012): By normalizing the attributes between 0 and 1, the proposed method considers attributes as truth degrees, and then applies a method similar to the one suggested with analogical dissimilarity. Using the same interpretation of normalization, (Bounhas, Prade, and Richard, 2017) follows a different way: The algorithm explores all possible triples (a, b, c) (with a restriction of c being in the nearest neighbors of the considered point), computes the corresponding credit and sums the credits for each possible classes (obtained by analogy).

Another recent application of analogical reasoning is recommendation. The task of recommendation can be summed up as follows: Given a set of observed user-item interaction (eg. ratings or clicks), a recommender system has to estimate possible future interactions. A famous example of a recommender system is the movie recommendation task, where the system has to select movies that users might like.

The first attempt to model recommendation in terms of analogical reasoning is the contribution of (Sakaguchi et al., 2011) for the context of dish recommendation. The authors model recommendation as a four-terms analogical problem. A user is associated to a list of already eaten dishes. If in a given context, a user A is known to eat dish B, then recommending a dish to a user C in the same context can be seen as solving the analogical equation $A : B :: C : x$ where x corresponds to the recommended dish. In practice, given a C, the two terms A and B are extracted from a case base.

In a different perspective, (Hug, Prade, and Richard, 2015) propose the following principle: If, for all items i rated by the users a, b, c, d, there is a proportional analogy between the ratings $r_{ai} : r_{bi} :: r_{ci} : r_{di}$, then there should be a proportional analogy for all items j not rated by user d as well. In this case the rating r_{dj} must be a solution of $r_{aj} : r_{bj} :: r_{cj} : x$.

3.4 Conclusion

In this chapter, we have presented a brief review of existing approaches of analogical reasoning. This review is far from being exhaustive, but we chose to focus on several questions and models that will be discussed in the following chapters. Naturally, the main perspective that will be discussed is the complexity-based approach of analogy, but we will provide links with structure mapping in general. The model of proportional analogy and the parallelogram rule will be discussed in details in Chapter 6. Regarding analogies on character strings, we will use Hofstadter's analogies as a milestone and will show that our framework can be extended to linguistic analogies as well. This extension will be proposed in Chapter 4 and then incorporated to the context of incremental learning in Chapter 12. Finally, the question of analogical reasoning and machine learning is the core of Parts II and III.

We would like to insist on the fact that analogical reasoning typically stands in the middle of two domains: machine learning and cognitive sciences. Even if the main topic of this thesis is undoubtedly machine learning, we think that cognition plays a prominent role in artificial intelligence and thus we chose to present methods and ideas from the community of cognitive sciences.

Chapter 4

Minimum Description Length Analogies on Character Strings

In the previous chapter, we proposed a general overview of the analogical reasoning methods and models. In this chapter we propose to introduce our concept of minimum complexity analogy (called *minimum description length analogies* in the context of this chapter) with a case-study: analogies on character strings. This choice is motivated by two main arguments. First, character strings are simple and natural objects which can be used and processed easily by human beings. Their simplicity and the natural variations that can be drawn around their structure and representation make them ideal toy examples to assess the general challenges of analogy. Secondly, character strings have various interesting applications in the domain of natural language processing.

In the scope of this chapter, we take Hofstadter's problem as a starting point. Hofstadter's micro-world, developed as a case study for analogical reasoning with the Copycat program (Hofstadter and Mitchell, 1995), involves problems of the form "**ABC** is to **ABD** as **IJK** is to x" (denoted **ABC**:**ABD**::**IJK**:x) where x has to be found. It extends directly to other classes of problems, for instance **walk**:**walked**::**fight**:x. The chosen approach to solve such analogies involves a description of the target strings by a generative programming language (similar to the language developed by (Strannegård, Nizamani, Sjöberg, and Engström, 2013) in the context of sequence continuation). This language defines a strict general framework and offers a generative and cognitively plausible description of analogies. With a small experiment on human beings, we validate our claim that relevance in analogical reasoning can be measured by description length.

The chapter is an extended version of (Murena, Dessalles, and Cornuéjols, 2017) and is organized as follows. In a first section, we describe Hofstadter's micro-world and the variants considered here: we will discuss the reasons why this toy example is particularly significant for the study of analogies. In Section 4.2 we propose our representation bias for Hofstadter's problem and describe the generative procedure to describe character strings analogies. Based on the presented language, we introduce the notion of relevance in Section 4.3 and discuss automation perspectives in Section 4.4.

4.1 Introduction: Hofstadter's Micro-World

In this section, we present the original micro-world developed by Douglas Hofstadter and his team in order to experiment on fluid concepts and analogical reasoning. A discussion will be developed on the motivation and goals of this toy example.

We also propose an extended variant of this micro-world and expose that our variant can be used to solve linguistic analogies as well. Lastly, we present the general idea of our approach.

4.1.1 Hofstadter's Micro-World: Presentation and Discussion

In order to study general properties of proportional analogy, Douglas Hofstadter introduced a micro-world made up of letter-strings (Hofstadter and Mitchell, 1995). The choice of such a micro-world is justified by its simplicity and the wide variety of typical analogical problems it covers. The base domain of Hofstadter's micro-world is the alphabet, in which letters are considered as Platonic objects, hence as abstract entities. Elementary universal concepts are defined relatively to strings of letters, such as *first*, *last*, *successor* and *predecessor*. These concepts do not describe elements of the alphabet directly as they are abstract entities, but their concrete appearances. To this domain is added a base of semantic constructs defined by Hofstadter: copy-groups, successor-groups and predecessor-groups (Hofstadter, 1984). The typical problem considered by Hofstadter in this micro-world is the following: if **ABC** changes to **ABD**, what is the analogous change of **IJK**? Such a problem corresponds to the analogical equation **ABC : ABD :: IJK : x** where **x** is the unknown parameter to be found.

As exposed in Section 3.2.5, this micro-world was initially designed by Douglas Hofstadter in order to assess the problem of fluid concepts in analogy, by providing a simple but complete domain where the major difficulties of analogical reasoning are present, but without involving the complexity of domains such as those assessed by Structure Mapping Theory or connectionist models.

We consider a slightly modified version of Hofstadter's problem. Our modifications correspond to an extension of the micro-world and are justified by some weaknesses of the original model. In particular they define a cross-domain problem, which is by nature more realistic. Another weakness overcome by our modification is the interconnection between symbols and semantics: In Hofstadter's rules, a letter cannot be interpreted in terms of its position in the alphabet.

The modifications we propose are the following.

First, we consider the possibility to use additional base alphabets, among which the number alphabet occupies a very special place. New alphabets offer the possibility to consider cross-domain analogies, which raises the issue of transfer between different domains. In particular, the analogical equation **ABC : ABD :: 123 : x** seems very basic for a human mind while it corresponds to a change of representation from the domain of letters to the domain of numbers. Interestingly enough, the numerical alphabet adds an infinite number of elements to the problem but does not make it fundamentally more complicated.

Besides, the use of other base alphabets is also justified by the flexibility of human cognition toward some prior knowledge. For instance, any system familiar with the English keyboard layout has the prior knowledge of this new domain in mind and will be particularly *efficient* to solve equations such as **ABC : ABD :: QWE : x**. The question of efficiency is not an easy problem and will be discussed in details in the following sections.

Secondly, we consider a mapping from numbers to any base alphabet. This operation makes it possible to describe a letter by its position in the alphabet and to design analogies involving a mapping between operation numerical parameters and letter position, which was discarded by Hofstadter's rules but seems important to us. The problem **ABC : ABD :: ABBCCC : x** relies on a such a mapping: the

string **ABBCCC** is naturally described as "*n-th letter of the alphabet repeated n times for* $n \in \{1, 2, 3\}$".

4.1.2 An Application: Linguistic Analogies

Originally, Hofstadter's micro-world was only intended to be applicable to artificial problems involving relational operations over character strings (with the relations exposed earlier: predecession, succession, copy). However it appears as a direct application case that the model can be used to solve linguistic analogies of the same nature as the ones presented in Section 3.3.2.

Traditionally, Hofstadter's works are not seen as a valuable alternative to manage these linguistic problems. In order to discard Copycat as a viable tool for solving linguistic analogical equations, one argument is often evoked: In linguistic examples, the order of letters does not bring any information and the system is overconstrained. In particular, it can be observed that Copycat fails to solve very simple declension problems such as **vita:vitam::rosa:rosam**. Compared to Hofstadter's approach, the methods designed for solving linguistic analogies have a clear advantage: They do not involve any prior knowledge.

However, we propose to rehabilitate Hofstadter's micro-world as a viable description tool for linguistic analogies.

From our point of view, having more descriptive power is not necessarily a disadvantage. To our knowledge, linguistic problems do not involve any transformation based on letter order, thus the descriptive tools relative to letter order can simply be ignored when describing linguistic analogies. We will show later that the system we designed is very modular: In particular, it can be used without any background knowledge at all and works particularly well for linguistic analogies in these conditions.

Based on this remark, it might seem strange that Copycat fails at solving even the simplest linguistic analogies. The justification of this observation has to be found in another direction. The description power is not the only limitation of a technique. Just because *in theory* an object can be described by a method does not mean that it will be actually described. The description power corresponds to what is called *representation bias*. This bias affects the theoretically reachable set of problems. For instance, the framework we propose can be applied to any analogy on character strings of any kind, either based on structural manipulations (e.g. **ABC:ABD::IJK:IJL**), on linguistically plausible analogies (e.g. **rosa:rosam::vita:vitam**) or on completely random logic (e.g. **XJT:YKNVTK::FY:XLL**). In practice, a reasoning method is necessarily biased by the way it explores the space of solutions: We will refer to this bias as *research bias*. In the case of Copycat, the exploration is based on finding local and global structural changes. If such changes are not detected, which is the case in particular when the strings at play correspond to natural language words, the algorithm fails to apply changes.

The observed weakness of the Copycat project to solve linguistic analogies cannot be attributed to the fact that Hofstadter's micro-world can involve background knowledge, but is a consequence of the fact that the program searches structural changes only. We will show that our technique avoids this pitfall.

As a conclusion to this discussion, we would like to notice that standard techniques for solving linguistic analogies are also biased by the choice of rules used to determine the solutions. Methods based on analogical proportion also have their

own biases, even if they do not rely on *a priori* knowledge. For example, the algorithm proposed in (Lepage, 1998) is explained by the authors to be inefficient in multiple cases, including reduplication (e.g. **orang : orang-orang :: burung : burung-burung** in Indonesian[1]) and permutation (e.g. **yaqtilu : yuqtilu :: qatal : qutal** in Protosemitic[1]). This observation is explained by the bias inherent to the axioms of proportional analogy.

4.1.3 Method Overview

In order to solve analogies in Hofstadter's extended micro-world, we propose a description language for alphanumerical analogies. The purpose of this language is to mimic descriptions that human beings could give of character strings and analogies. A major difference between our approach and Hofstadter's original works lies in the consideration of descriptive groups. While Hofstadter's approach is merely descriptive, we adopt a generative formalism in which the way strings were formed is taken into account. The static description of *copy-groups*, *successor-groups* or *predecessor-groups* is replaced in our framework by methods such as *copy*, *succession* or *predecession*.

Once the description of an analogy is available in our generative language, the instructions are turned into a binary code according to strict coding rules. Our claim in this chapter is that the solution of an analogical equation perceived as the most *relevant* one is produced by the code of minimal length. This claim will be justified based on some experimental results. We will discuss the impact and limitations of the observed results in the conclusion.

4.2 Representation Bias for Hofstadter's Problem

In this section, we present the rules of the language developed to describe character strings analogies. These rules limit the description and can be seen as a representation bias.

4.2.1 A Generative Language

As mentioned, a major difference between our perspective and Hofstadter's works is the generative point of view. Largely inspired by Leyton's theory of shapes (Leyton, 2001), we consider a description of the process generating analogies rather than a description of the analogies themselves. According to Leyton, the history of an object is perceived as the result of a sequence of transformations starting from a completely symmetric state to an asymmetric state. In his examples, Leyton considers forms in a two dimensional space, however his theory is general and it can apply in particular to our domain of interest. In the context of our micro-world, the only totally symmetric structures are the alphabets themselves. Because they are pure abstract base elements, they have a fundamental pre-existence for our system. Any string will result from a transformation of the base alphabet: for instance, **ABCDE** is perceived as the sequence of the first five letters in the alphabet and **ZYX** as the sequence of the first three letters in the reversed alphabet.

In order to integrate this sequential transformation of an original string, we consider that the machine has access to a one-dimensional discrete tape. At each

[1] Example taken from (Lepage, 1998).

time step, the machine writes on this tape (from the left to the right if no counter-indication is given) or modifies the previously written string. Thus, the base operation consists in copying the alphabet onto the tape. In the two preceding examples, the transformations can be easily expressed in human words. More generally speaking, one of the desired properties of a generative language for analogies is its cognitive interpretability. In our solution, the generative procedure consists in a sequence of operations read from left to right and separated by commas. The operations are applied one by one and refer to understandable manipulations. Even if any operation may be incorporated to the language, we will consider here only a restricted set of predefined transformations, called operators $\{\mathcal{O}_1, \mathcal{O}_2, \dots\}$.

The core of the language is the use of a triple memory: a long-term domain memory, a long-term operator memory and a short-term memory. The long-term domain memory stores all accessible domain descriptions, including the alphabets that are accessible to the system. This memory will be denoted by $\mathcal{M}_d = \{\mathcal{A}_1, \mathcal{A}_2, \dots\}$ where the $\mathcal{A}_i$ designate the alphabets. The long-term operation memory stores the repertoire of all applicable operators and is denoted by $\mathcal{M}_o$. Both long-term memories contain prior knowledge and cannot be modified by the machine. All memory modifications are done in the short-term memory which stores *concepts* to be reused in the description. Here, concepts can be either strings or system-defined operators. The short-term memory is designated by $\mathcal{M}_s$.

Using the ideas exposed above, we design a set of rules defining a sketch of a grammar for the presented generative language. The rules presented here are general and do not describe the available operators. A list of elementary operators will be presented and discussed later.

1. A program is encoded as a list of predicates separated by commas. Instructions are read from left to right: this order coincides with the execution order.

2. The program uses a one-dimensional and infinite tape. Intermediate results are written on the tape. Each instruction modifies the content of the tape, either by adding new elements or by correcting previous characters.

3. The base element of a string is called a *group*. A group is recursively defined as a concatenation of groups. The minimal group is made up of one letter. The whole string written on the tape corresponds to one group.

4. Instructions generate groups, either by replacing the group on which they apply or by concatenating their output to it.

5. By default, operators apply to the whole string. To apply the operator to one precise group only, the instruction is declared inside another special instruction called `group`. The `group` instruction can be seen as a way to change the scope.

6. Operators apply to the preceding group and are specified with at most one single parameter. If no parameter is given, a default parameter is used.

7. A string can be put into short-term memory by means of the special instruction `let`. The short-term memory can be accessed with the key instruction `mem`. This instruction requires a parameter, which is the position in the list. The position is given by an integer, with the convention that lastly memorized strings can be retrieved with low index[2] (see Section 4.2.3 for more details).

[2]This convention is motivated by the idea that lastly memorized items are the easiest ones to recall.

8. Operators can be put into short-term memory and accessed respectively with the key words `let` and `mem`. In the declaration, the parameter is indicated by the character `?` and may be used at several places in the instruction.

9. The instruction `next_block` is used to move to the next term in the analogy definition. For the analogy **A : B :: C : D**, the order of the blocks is **A**, **B**, **C** and **D**.

The choice of a mono-dimensional tape can be discussed in the perspective of solving analogies. By construction, analogies are read in a two-dimensional way (which is particularly clear in the symmetry and central permutation axioms of proportional analogies). Our choice is motivated by two ideas. First, we recall that our initial motivation is to mimic natural language, which is by nature mono-dimensional. Even if this argument applies mainly to the generative language itself, it also affects the produced objects. The second motivation concerns the use of the language: we observe that the apparent linearity is perturbed by the memory which can store partial descriptions and thus model cross-domain dependencies.

4.2.2 Basic Operators

The list of operators available for the language determines a bias for the machine. The more operators are given to the system, the more sophisticated the obtained expressions can be.

The most basic set of programs is empty: it corresponds to a system capable of giving letters one by one only. Such a system is sufficient in some contexts. Consider for example the real problem of learning declension in a language. In order to learn a declension, students learn by heart a single example and transfer the acquired knowledge to new words. This corresponds for instance to the analogy **rosa : rosam :: vita : vitam** for a simple Latin declension. This analogy is encoded by the following code:

```
let('r','o','s','a'), let('v','i','t','a'),
  let(?, next_block, ?, 'm'),
  mem, 0, mem, 2, next_block, mem, 0, mem, 1;
```

This program has to be interpreted as follows: In the first line, the groups `'rosa'` and `'vita'` are put in short-term memory. The second line defines a new operator which displays the argument, switches to the next block, displays the argument again and finally adds the character `'m'`. The third line retrieves the just-defined operation and applies it successively to the two words, also retrieved from memory.

In order to build effective descriptions for more complex systems, additional operators can be defined. We propose a summed-up list of the defined operators in Table 4.1 and a list of code examples in Table 4.2.

Two basic operations can be considered as a generative equivalent of the *copy-groups* and *successor-groups* in Copycat: `copy` and `sequence`. The operator `copy` repeats the group of interest a given number of times. The parameter of the operator is the number of copies and has 2 as default value. The operator `sequence` outputs the sequence of the first n elements of the group, where n is the parameter. The default value for the parameter is 1. The elements selected by the operator correspond to subgroups of the total group, not necessarily to actual characters.

The operator `sequence` alone is not as general as Hofstadter's successor-groups: For example, it cannot describe the sequence `ijk`. In order to cope with this difficulty,

Name	Description	Example
`copy`	Repeats the group a given number of times. Equivalent to Hofstadter's *copy-group*.	`'a', copy, 4;` outputs `aaaa`
`sequence`	Outputs the sequence of the first n elements of the group. Equivalent to Hofstadter's *copy-group*.	`alphabet,sequence,3;` outputs `abc`
`shift`	Shifts the subgroups of n positions.	`alphabet,shift,3;` outputs `defg...yz`
`shift_circular`	Circular version of the `shift` operator	`alphabet,shift_circular,3;` outputs `defg...yzabc`
`reverse`	Reverses the order of elements in a group.	`alphabet,sequence,3, reverse;` outputs `cba`
`find`	Searches all occurrences of a group given as parameter.	`'a', 'b', 'a', find, 'a', copy, 2;` outputs `aabaa`
`last`	Selects last group	`'a','b','a',last,copy,2;` outputs `abaa`
`map`	Maps an operation to the children of a group	`alphabet,map,copy,2` outputs `aabb..yyzz`

TABLE 4.1: Example of operators used by the language.

we introduce the `shift` operator. Given with parameter n, the operator shifts the subgroups in the subgroup of n steps. The shift is not circular, but a circular version of it may be defined if needed.

The operator `reverse` is used to reverse the order of elements in a group. This operator does not have parameters.

Lastly, the operator `map` applies an operator (specified as parameter) to all children of the group of interest. The parameter is an operator specified with its parameter (if needed).

To these writing operators, we have to add another class of operators, which will be designated in the following as *pointing operators*. Unlike previously described operators, pointing operators are used to extract subgroups on which the following operator will apply. By default, an operator applies to the whole string in its scope. However, in some cases, an operation is needed on several subgroups inside the total group, hence operators are needed to point toward desired subgroups. We propose two pointing operators: `find` and `last`.

The operator `find` searches all occurrences of a group `g` inside the group of interest. The group `g` is given as a parameter. The operator `last` (specified with no parameter) selects the last subgroup.

4.2.3 Using Memory

The strength of the proposed language lies in its use of a triple memory to access elements of different nature: a long-term domain memory $\mathcal{M}_d$ storing domain descriptions (e.g. alphabets), a long-term operator description $\mathcal{M}_o$ storing system procedures to modify objects, and a short-term memory storing temporary elements. Managing memory is of major importance when it comes to producing programs of minimal length.

Instruction	Output
`'a', copy, 4;`	`aaaa`
`group('a', 'b'), copy, 2;`	`abab`
`alphabet, sequence, 3;`	`abc`
`group('a'),group('b','c'),group('d'),sequence,2;`	`abc`
`alphabet, shift, 3;`	`defg...yz`
`alphabet, shift_circular, 3;`	`defg...yzabc`
`alphabet, sequence, 3, map, copy, 2;`	`aabbcc`
`alphabet, sequence, 3, reverse;`	`cba`
`'a','b','a', find, 'a', copy, 2;`	`aabaa`
`alphabet, sequence, 4, last, copy, 3;`	`abcddd`

TABLE 4.2: Example of instructions involving various possible operators. The outputs correspond to the strings generated by the corresponding code.

The access to elements in long-term memories $\mathcal{M}_d$ and $\mathcal{M}_o$ is hidden in the language for simplicity purpose, but it cannot be ignored. The designation of support alphabets (`alphabet`, `numbers`, `utf8`, `qwerty-keyboard`...), hence of the domain, and the designation of operators (`copy`, `sequence`, `find`...) are treated as proper nouns to encapsulate an access to an ordered memory. The rank of entities in memory is a characteristics of the machine and cannot be changed.

The user is in charge of the management of short-term memory. Entities (operators or strings) are stored in memory with the `let` meta-operator and accessed with the `mem` meta-operator. For example, the instruction `let('a')` will store the generation of `a` but the string is not written on the band. It will be written only when invoked from memory. The short-term memory is organized as a stack (hence last-in first-out): the parameter given to the `mem` operator is the depth of the element in the stack. Thus, the last element memorized will be invoked using `mem,0` or simply `mem,`, since default parameter for `mem` is 0.

The declaration order for memorized entities may be arbitrary. For instance, there is no difference between instruction `let('a'),let('b'),mem,1,mem,0;` and instruction `let('b'),let('a'),mem,0,mem,1;`. Both output `ab` even if the order in memory is not the same.

Using short-term memory is not compulsory to describe a string: The language syntax does not prevent from repeating identical instructions. However, in a context of finding a minimal description (which is the purpose of our framework), using memory is an important way to pool identical entities.

4.2.4 Remarks on the Language

We would like to end up this section with several remarks on the proposed language.

First, we proposed a compiler for this language, implemented in Python. The compiler is able to consider a sequence of instructions and to compute the corresponding character string. As it will be discussed in Section 4.4, we did not provide any method to obtain the optimal sequence of instructions to describe one analogy given as a parameter. The compiler exploits the notion of group, which is mostly implicit in our language. Each operator produces a list of groups, and the output of a sequence is the concatenation of all strings in the list. However, we imposed a restriction on the compiler by introducing an arbitrary maximal number for integers. This restriction is used to avoid infinite loops (for instance with instructions such

as `numbers, shift, 2`) but should be removed in a more proficient implementation and replaced by a syntax error.

The second point we would like to insist on is the general aspect of the language, in the sense that it can be used to generate any possible analogy, including those which might seem random. When no regularity is found, a character string can be described by encoding each character one by one, which our language is able to do.

Finally, despite this complete description ability, the language is trivially not Turing-complete: It can be verified by considering that the halting problem can be solved for it.

4.3 Relevance of a Solution

In this section, we discuss the notion of relevance of a solution. We will propose to use the developed language to describe the analogies and measure their "simplicity".

4.3.1 Relevance: Problems and Intuitions

The language exposed in Section 4.2 defines the expressive power of our method. At this point, a couple of remarks can be done.

First, the language does not apply to Hofstadter's analogies only but can be extended to any problem involving structural description of character strings. As such, a natural application is string completion, in the same spirit as performed by (Strannegård, Nizamani, Sjöberg, and Engström, 2013). This application will not be discussed in this thesis.

A second problem raised by our language is the **non-unicity** of description for a given string. In general, infinitely-many instructions can produce a same result. For instance, the observation `abc` can be generated *inter alia* by any of the following instructions:

1. `alphabet, sequence, 3`
2. `'a', 'b', 'c'`
3. `alphabet, sequence, 2, 'c'`
4. `alphabet, sequence, 3, copy, 1`

The question that arises from this observation concerns the *relevance* of a description among others: **What makes one description better quality than another description?**

The third problem is closely related to the resolution of analogical equation. When only three of the four terms in the analogy are known, is there any criterion to discriminate relevant and admissible solution? In other words: **What makes one solution better quality than another solution?**

The solutions we explore use a fundamental theoretical tool, *Minimum Message Length* (MML). This idea is in line with various models of analogy and of letter string continuation, including (Cornuéjols and Ales-Bianchetti, 1998) and (Strannegård, Nizamani, Sjöberg, and Engström, 2013).

The Minimum Message Length principle states that the optimal solution of a problem is the shortest in terms of description length. This principle, which will be discussed in more details in the following chapters, can be seen as a formalization of Ockham's razor principle. At this stage of the thesis, we use only an intuitive and informal (but accurate) expression of MML principle in order to test our idea.

Index	1	2	3	4	5	6	7	8	9	10	11	12
Code		0	1	00	01	10	11	000	001	010	011	100
DL	0	1	1	2	2	2	2	3	3	3	3	3

TABLE 4.3: Positional code in a list and corresponding description length (DL, in bits)

In what follows, we will apply Minimum Message Length principle to the questions addressed here: How to measure the *relevance* of a description and the *relevance* of an analogy? Before we can actually measure these two quantities, we have to quantify the length of a description: For this purpose, we turn the instructions written in our language into a binary code, the length of which will be considered as a measure for the actual description length.

4.3.2 From Language to Code

We consider here the question of coding a description of analogies based on the proposed language. We denote the alphabet of instructions by $\mathcal{A}$ and $\bar{\mathbb{B}} = (\mathbb{B}^*)^*$ the set of sequences of binary sequences. We propose now to build a pseudo-code $\mathcal{C} : \mathcal{A} \to \bar{\mathbb{B}}$ which associates each instruction word to a sequence of binary sequences. We call the extension of the code the function $\tilde{\mathcal{C}} : \mathcal{A}^* \to \bar{\mathbb{B}}$ defined as follows: For all $x_1, \ldots, x_n \in \mathcal{A}, \tilde{\mathcal{C}}(x_1, \ldots, x_n) = (\mathcal{C}(x_1), \ldots, \mathcal{C}(x_n))$.

These definitions differ a bit from the usual definitions used in coding theory, in which a code is a binary sequence, and not a list of binary sequence. This difference is motivated by the cognitive inspiration of our method. we consider that a cognitive system does not use a prefix binary code and is able to discriminate between instructions and words on a cognitive layer.

The basic idea we use to obtain an efficient code consists in using a *positional code* in lists. This code associates the empty sequence to element 0, `0` to element 1 and increments of 1 bit for each element (`0`, `1`, `00`, `01`, `10`...: see Table 4.3). Using this code, the description length of the n-th element of a list is $\lceil \log_2 n + 1 \rceil$.

The global presentation of the language is organized as a list of lists: A word is designated by the path inside the sequence of lists. For instance, the code for the character `d` corresponds to the code of domain memory (`1`), alphabet (`0`) and `d` (`01`), hence `1,0,01`. This comma-delimited sequence corresponds to a program on a ternary alphabet $\{0, 1, B\}$ where the blank character B corresponds to the comma.

The corresponding binary code (obtained when the blanks B are ignored) is not self-delimited.Without the commas, the code of the programs would **not** be decodable, in particular due to the presence of empty codewords. In this chapter, we diverge from the mathematical theory of complexity by making two strong assumptions: the system is able to split the instructions correctly and these delimiters are not considered as being part of the code (which affects the measure of complexity). These two assumptions are motivated by cognitive modeling and will be used in this chapter only.

To make the code consistent with the formal theory of complexity and have it as a binary prefix code, a classical doubling code can be used. Given a binary sequence x (of length $l(x)$), the doubling code of x is the concatenation of 3 elements: `0` repeated $\lceil \log_2 l(x) + 1 \rceil$ times, followed by a `1`, followed by the binary coding of $l(x)$ followed by x. This doubling code has the property to be a prefix code and thus uniquely decodable. Even if we will use this convention in the following, we omit it in this introductory chapter, for its lack of cognitive relevance.

Since a language word corresponds necessarily to a tree leaf, the code is uniquely decodable.

Proposition 1. *Any instruction sequence encoded with the described positional code is uniquely decodable.*

Proof. Consider two ternary sequences $p_1, p_2 \in \{0, 1, B\}$. By construction of the code, if p_1 is a prefix of p_2, then p_1 does not correspond to a leaf of the tree and thus is not a code word. This shows that the code is a prefix code, and consequently is uniquely decodable.

□

An example of an instruction tree is given in Figure 4.1. The ordering suggested by this instruction tree is of course entirely arbitrary (except for the blank element the purpose of which is to avoid specifying unnecessary parameters, and thus has to count for a low description length). This arbitrary order is another bias of our system. The order chosen in Figure 4.1 has been determined by favoring more frequently expected operators at lower positions than less frequently expected ones. A way to build a cognitively plausible language encoding would consist in evaluating the ordering based on human experiments. Such experiments would have to be made in future research.

The length of an instruction is determined from the corresponding code. We propose to consider that the program length corresponds directly to the number of bits required in the code[3]:

$$\forall p \in \{0, 1, B\}, \quad L(p) = \sum_{i=1}^{l(p)} \mathbb{I}(p_i \neq B) \tag{4.1}$$

For instance, the length of the instruction `2` will be the number of bits in `1,0,0`, hence $L(2) = 3$. The same reasoning is applied to any instruction, including complex instructions describing complete analogies. Note that we improperly associate the length of a program in the proposed language to the length of its associated instruction sequence $p \in \{0, 1, B\}$.

The comma delimiter is considered as costless when computing description length. This idea is in use in the Morse code for example. Morse code encodes letters by sequences of dashes and dots (ie. with a binary alphabet). A full word is given by a succession of letters separated by short breaks. These breaks are not part of the Morse code but are used to indicate the transition from one letter to another. In such contexts, the delimiters are supposed to be processed by the physical layer of the system, hence to ensure a uniquely decodable code while having no influence on actual description length. Considering costly delimiters would increase the description length of an instruction p by a constant in $\mathcal{O}(L(p))$ bits.

4.3.3 Relevance of a Description

Several acceptable instructions can generate a given string. For example, the string `abc` can be produced by at least three instructions:

- **Instruction 1:** `alphabet, sequence, 3;`

[3]Note that, using this definition, the length of an operator is independent of the operation it performs but is only related to its position in the operator memory.

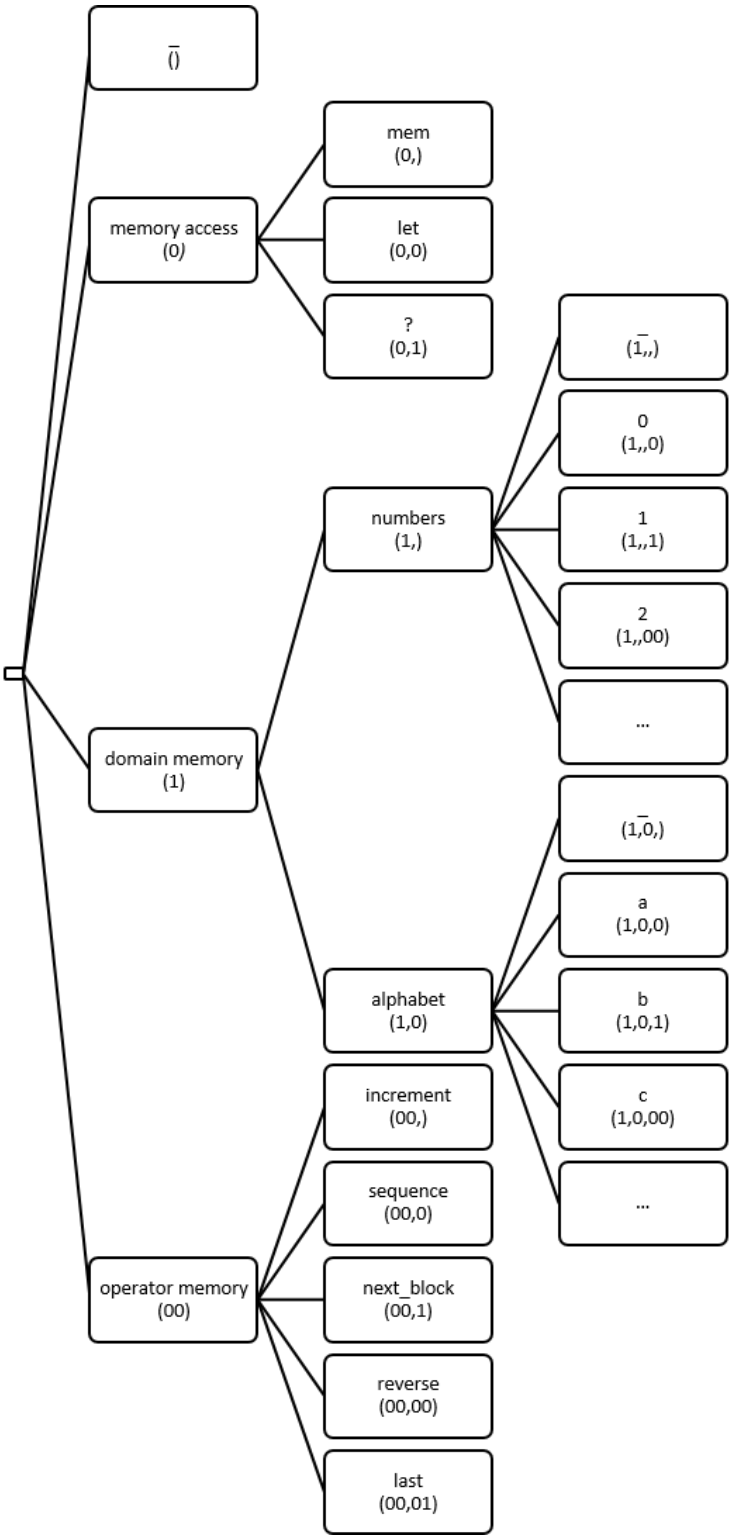


FIGURE 4.1: Instruction tree used for the code. Each element is indicated with its name in the language and the corresponding code (written in parenthesis).

- **Instruction 2:** `'a','b','c';`
- **Instruction 3:** `alphabet, sequence, 2, 'c';`

These three instructions do not seem equally satisfying from a human point of view. We submit that the difference in terms of relevance can be quantified by their length.

Using a specific code description, the description lengths for the three previous instructions are respectively $L_1 = 8$, $L_2 = 10$ and $L_3 = 12$. In this example, it is observed that the instruction with minimal description length corresponds to a cognitively relevant description of the string, since it exploits its intrinsic structure.

As a first step of our reasoning, we state that *the most relevant generative description of a string is the description of minimal description length*. We define the description length $DL(.)$ of a character string (or an analogy on character strings) as the minimal length of an instruction that generates it. For instance, we have shown here that $DL(\texttt{abc}) \leq 8$ bits.

Despite the simplicity of our generative language (in particular compared to the unrestricted class of Turing machines generating character strings), the exploration of the space of programs is difficult, and all the estimations that are given in this chapter are only upper-bounds of the actual description length. In theory, an exploration of all possible instructions is possible, since at least one program is known and fixes a maximal value for the description length: the program that enumerates the letters of the analogy one by one. In practice, such an approach would have an exponential time complexity.

Several solutions can be adopted in order to build the optimal program. First, greedy approaches would impose a research bias by the mean of a locally optimal exploration of the space of programs. This solution will be discussed in the perspectives of this chapter. Additionally to this guided exploration of the space of programs, a resource-bounded research can be considered (Buhrman, Fortnow, and Laplante, 2001). In particular, a research time can be given to the computer, and only programs built within this specified time period are taken into account for the evaluation of the description length.

4.3.4 Relevance of a Solution for Analogical Equations

Using the description length obtained by our system as described above, it is possible to apply a minimum description length strategy for the selection of a relevant decision rule.

Consider the example of the simple analogy equation **ABC : ABD :: IJK : x**. Infinitely many solutions can be proposed to this equation, but we can consider two of the most frequent answers given by humans: **IJL** and **IJD**. We propose the two following descriptions for these solutions (note that in these instructions, the number 8 corresponds to the position of letter I in the alphabet):

```
// ABC : ABD :: IJK : IJL

// Step 1: Source problem description
let(alphabet, shift, ?, sequence, 3),
  // Step 2: Intra-domain description
  let(mem,, ?, next_block, mem,, ?, last, increment),
  // Step 3: Application of the descriptions
  mem,,, next_block, mem,, 8;
```

```
// ABC : ABD :: IJK : IJD

// Step 1: Source problem description
let(alphabet, shift, ?, sequence, 3),
  // Step 2: Intra-domain description
  let(mem,, ?, next_block, mem,, ?, last, 'd'),
  // Step 3: Application of the descriptions
  mem,,, next_block, mem,, 8;
```

These two programs are inspired of human explanations for the analogies. Consider the first program. The description of the source problem has to be read as follows: Take alphabet, shift it by a number n (input), and take the sequence of the first three letters. The intra-domain description consists then in taking the source problem description in memory, apply it an input parameter, and provide its solution by incrementing the last element. Finally, these instructions are applied in the source with no parameter (hence parameter 0), and in the target with parameter equal to 8. The second program is identical, except for the intra-domain description, in which the solution is obtained by replacing the last letter by a `d`.

These two programs cannot be proven to be optimal descriptions. We may consider them however as provisionally optimal in a resource-bounded approach and use them for the evaluation of description length. We can deduce the description length of the two analogies to be $DL(\mathbf{ABC} : \mathbf{ABD} :: \mathbf{IJK} : \mathbf{IJL}) = 37$ and $DL(\mathbf{ABC} : \mathbf{ABD} :: \mathbf{IJK} : \mathbf{IJD}) = 38$. In particular, the difference between these two solutions is 2 bits. The drop of description length from the higher DL solution down to the lower DL solution measures the *relative relevance* of the two solutions. The larger this drop, the more relevant the optimal solution.

4.3.5 Validation

In order to evaluate the way human beings react to analogy problems, we proposed an online experiment with several Hofstadter's analogy problems. Participants were 101 (62 female), ages 14-72, from various social and educational backgrounds. Each participant was given a series of analogies. The series were in the same order for all participants, and some questions were repeated several times in the experiment, in order to test a potential priming effect. All analogies had in common the source transformation **ABC : ABD**. The main results are presented in Table 4.4. The complete results, as well as the survey for the experiment, is available in Appendix A.

The results confirm that in most cases the most chosen solution corresponds to a minimal value of the global description length. The description length is calculated here using our small language and the coding rules exposed earlier. Its limits are visible with the two examples **ABC : ABD :: 135 : x** and **ABC : ABD :: 147 : x**. In these examples, the language fails at describing the progression of the sequence "two by two" (1-3-5-7) or "three by three" (1-4-7-10) which would decrease the overall description length.

However, despite the simplicity of the language used to assess the description, it is noticeable that the most frequent solution adopted by the users corresponds to a drop of description length. This property is not verified with only two problems: For the problem **ABC : ABD :: 122333 : x**, the large value of the description length in the most frequent case is due to the limitations of the language which fails at providing a compact description of the complete analogy because of a too rigid grammar. In the case of the analogy **ABC : ABD :: XYZ : x**, adding the circularity constraint has a cost in the language, while it seems to be a natural operation for human beings.

Problem	Solution	Proportion	DL (bits)
IJK	*IJL*	94%	37
	IJD	2.0%	38
BCA	*BCB*	50%	42
	BDA	38%	46
AABABC	*AABABD*	75%	33
	AACABD	15%	46
IJKLM	*IJKLN*	61%	40
	IJLLM	15%	41
123	*124*	96%	27
	123	3.0%	31
KJI	*KJJ*	40%	43
	LJI	33%	46
135	*136*	70%	35
	137	20%	37
BCD	*BCE*	83%	35
	BDE	3.0%	44
IJJKKK	*IJJLLL*	39%	52
	IJJKKL	26%	53
XYZ	XYA	84%	40
	XYZ	5.0%	34
122333	122444	35%	56
	122334	31%	49
RSSTTT	*RSSUUU*	41%	54
	RSSTTU	30%	55
IJJKKK	*IJJLLL*	40%	52
	IJJKKL	27%	53
AABABC	*AABABD*	68%	33
	AACABD	16%	46
MRRJJJ	*MRRJJK*	26%	64
	MRRKKK	23%	65
147	*148*	71%	36
	1410	10%	38

TABLE 4.4: Main results of the survey for Hofstadter's analogies. For each problem, only the two main solutions are presented, with their frequency and the corresponding description length (DL). Some problems are repeated multiple times in order to test a potential priming effect. The solution written in italic corresponds to the solution of minimal description length.

The experiment also reveals a major weakness of our model: The descriptions provided by our language are static and do not depend on the environment. On the contrary, the variations of the average answering time and the changes in the answers (when a same problem is repeated at several places) indicates clearly that having faced similar structures in the past helps in solving a new analogy. Finally, the relative relevance of two solutions is not necessarily sufficient to explain human preference in this matter, though. For instance, on the first problem, a large majority of people choose the **IJL** answer despite the small description length difference. This possible divergence is related to research biases which are not taken into account in our approach. This effect is particularly visible with the more difficult analogy equation **ABC : ABD :: AABABC : x**. Very few humans notice the structure **A-AB-ABC**, hence the corresponding solution **x** = **AABABCD**. However, the structure **A-AB-ABC** is perceived as more relevant when presented.

We have shown that description length offers a criterion to compare two given solutions to an analogy equation. This sole property is not sufficient in practice to obtain an analogy solver. Since the space of solutions is infinite, additional hypotheses must be considered in order to restrict the exploration space.

4.4 Perspectives: Finding an Optimal Representation

In order to develop an efficient automatic analogy solver based on complexity minimization, two issues have to be overcome: How to effectively compute the description length of a complete analogy, and how to explore the space of solutions. Regarding the second problem, we propose to draw a parallel between the main phases of the Copycat program and the construction of an instruction in our generative language.

Several phases are described in Hofstadter's Copycat program (Hofstadter, 1984): syntactic scanning, semantic phase, rule generation, world mapping, rule slipping, rule execution and closure checking. All these phases can be transposed directly in a description length minimization framework. We propose to examine them with the example **ABC : ABD :: IJK : x**. We will present code structures for each phase and show that the phases can be described in terms of memory. The ideas proposed here are not implemented yet and have to be handled in future works.

4.4.1 Syntactic Scanning and Semantic Phase

Syntactic scanning examines immediate syntactic connections inside all strings. For instance, successions or repetitions are targeted during this phase. This approach offers a bottom-up exploration of the description space. Unlike Copycat's approach (separating syntactic and semantic description), we propose to merge the two phases in a first description of the analogy. In our example, we obtain the following description:

```
alphabet, sequence, 3, next_block, alphabet, sequence, 2, 'd',
next_block, group(alphabet, shift, 8, sequence, 3)
```

4.4.2 Rule Generation

Rule generation focuses on the first domain of the analogy (hence **ABC : ABD**) and aims at factorizing it. The purpose is to make a transformation appear during factorization. Here, the idea is to factorize the common structure **ABC** or **AB** and to propose a transformation for both of them. The factorization is made using memory.

```
// Factorization of ABC
let(alphabet, sequence, 3), mem,, next_block, mem,, last, increment;

// Factorization of AB
let(alphabet, sequence, 2), mem,, 'c', next_block, mem,, 'd';
```

Once the structure is memorized, the transformation is stored into a second memory instance:

```
let(...), let(mem,, next_block, mem,, last, increment), mem,;
```

4.4.3 World Mapping and Rule Slipping

World mapping is a crucial step in analogical reasoning: it consists in unifying both domains by finding a correlation between them. In our example, the correlation has to be found in the expression of `alphabet, sequence, 3` and `alphabet, shift, 8, sequence, 3`. In this case, the correlation can be established using the instruction `shift, 0` that corresponds to the identity operator (mapping a sequence to itself). A factorization can then be proposed:

```
let(alphabet, shift, ?, sequence, 3)
```

This factorization leads to slight changes in previous code definitions. Such modifications correspond to the rule slipping phase. The modifications are stored in memory, in order to produce a general description method available both for source and target.

```
let(alphabet, shift, ? sequence, 3),
  let(mem,, ?, next_block, mem,, ?, last, increment);
```

4.4.4 Rule Execution

The final instructions are obtained through the following steps:

```
let(alphabet, shift, ? sequence, 3),  // Structure definition
  let(mem,, ?, next_block, mem,, ?, last, increment);  // Rule
  mem,,, next_block, mem,, 8;
```

Executing this instructions, we obtain analogy **ABC : ABD :: IJK : IJL**. The way the system generates each step is an open research problem and will have to be solved in order to obtain an actual analogy solver. Because this solver will rely on storing factorized structures in memory, the found solution will coincide with a local minimum of complexity. No guarantee can be offered in any way that this local minimum corresponds to a global minimum.

4.4.5 Cognitive Interpretation

The procedure proposed by Hofstadter for CopyCat is supposed to mimic human cognition. We would like to conclude this chapter with a brief interrogation on its cognitive plausibility.

The steps described above reveal an order in the interpretation of the analogy. The resolution can be split in two parts. First, the system finds an internal representation in the source domain (rule generation). In this first stage, no consideration on the mapping between the source and the target domains is involved. This link is established in a second step (world mapping and rule slipping). Using the generic notations $a : b :: c : d$, this procedure suggests then to consider first the relation $a : b$ and then the relation $a : c$ in order to infer d. Such procedures are called *project-first*. Such procedures are opposed to the models of structure mapping theory which analyze the mapping $a : c$ first.

Several cognitive studies have addressed the question of the resolution procedure for analogies. A recent analysis (Vendetti, Starr, Johnson, Modavi, and Bunge, 2017) provided a study based on eye-tracking, which points out varying behaviors with respect to the age of the subject. The results of this study show that the project-first approach is the most frequently used by adults, but also the most efficient in terms of correct answers.

These observations tend to indicate that the baseline proposed by Hofstadter is indeed cognitively coherent and should lead to a good accuracy in the resolution process.

4.5 Conclusion

In this chapter, we have presented a particular class of analogies called Hofstadter's analogies. These analogies involve structured character strings but can be extended to less structured strings, such as grammatical inflections (for instance conjugation or declension). We developed a generative programming language to describe objects in this micro-world and a way to convert these instructions into a uniquely decodable binary pseudo-code. Based on this pseudo-code, the *description length* of an object was defined as the lowest number of bits involved in a program generating the object. An experiment performed on human beings confirmed the intuition that human beings tend to prefer analogies of lower description length, which is a confirmation of previous studies on minimum description length analogies.

These very positive results have to be considered with care, though. It is obvious that the language we designed is **one** way among others to describe character strings. We chose this specific description because of the cognitive nature of the task: In order to mimic the way human beings react to Hofstadter's analogies, we propose a language expressive enough to encode the possible given descriptions. This imitation characteristic is essential here but is an *ad hoc* property. A central question in general will be to propose the one description language which corresponds to the nature of problem.

In the next chapter, we take a larger perspective and consider more general classes of analogies. We will show that the preliminary work proposed in the context of alphanumerical analogies can be extended easily and has a simple interpretation in terms of algorithmic information theory.

Chapter 5

Minimum Complexity Analogies

In the previous chapter, we proposed a description language for alphanumerical strings and applied it for the description of analogies of the form "**ABC** is to **ABD** as **IJK** is to **IJL**". We have shown that human beings favor a *minimum description length* strategy to choose the solution of such problems.

The purpose of this chapter is to give a general interpretation of this strategy. To do so, we will introduce the notion of *Kolmogorov complexity* which will play a central role in this thesis. Complexity is the theoretical counterpart of the description length used in previous chapter and measures the shortest description length (in bits) of a Turing machine generating a given object. We will show that a restriction of the space of Turing machines is needed but biases the system toward some results. In order to enhance the readibility of the models, we will propose the notion of *Descriptive Graphical Model* (DGM), largely inspired by probabilistic graphical models, but entirely based on Kolmogorov complexity. These graphical models will be used to define a general class of analogical relations.

The chapter will be organised as follows: In Section 5.1, we present an extension of the language developed for Hofstadter's analogies which applies to any analogy on any domain. We will show that the minimum description length naturally involves Kolmogorov complexity and we will present the base notions of algorithmic information theory. The introduced notions will be used in Section 5.2 to define the Descriptive Graphical Models and present some of their properties. Lastly, Section 5.3 proposes a graphical model for analogical reasoning. This model is similar to the approach of (Cornuéjols and Ales-Bianchetti, 1998). It will be inferred from the description language and will be presented with various examples.

5.1 A General Description Language for Analogies?

A restriction of analogies to Hofstadter's micro-world is obviously not desirable, despite the interesting modeling properties of this domain. From now on, we propose to consider more general analogies, hence analogies in broader domains. The purpose of this section is to relax the descriptive setting exposed above.

5.1.1 Analogies in Structured Domains

A characteristic of Hostadter's micro-world is the fact that it does not have a strict semantic structure. Even if the descriptions based on the developed language can be interpreted in terms of predicates, this structure is built by the system in order to solve the analogies but is not given as an input.

Working in structured domains (e.g. knowledge bases) is a necessity to consider more realistic cases. The same issues apply to such domains as those described for

character strings, but the proposed methodology has to be slightly modified to adapt to such domains.

We consider a domain represented by a propositional networks of nodes and predicates (such as suggested by (Gentner, 1983)). A node represents a concept and is engaged in predicates. The arity of a predicate can be arbitrary high. Predicates of arity 1 are called *attributes* (for instance `red(door)`) and predicates of arity 2 are called *relations* (for instance `attracts(sun, planet)`). The theory of structure mapping (introduced in Chapter 3) is based on this representation and suggests that the domain of analogy is the set of relations, rather than the object description: "The target objects do not have to resemble their corresponding base objects. Objects are placed in correspondence by virtue of corresponding roles in the common relational structure." (Falkenhainer, Forbus, and Gentner, 1989)

In order to describe such a domain, a description language can be found, similar to the one developed for character strings. Several modifications have to be considered though:

- Character strings are structured linear objects, which is not necessarily the case with standard objects.
- Predicates used in Chapter 4 are forced to have arity 2 (since methods have two arguments: the string they apply on and the parameter). Here, we consider more general cases.
- All available properties are not necessarily pertinent in the description of the analogy. It was also the case in the alphanumeric domain, but to a lesser extent. Here, in general, very few properties are engaged in the analogical process.
- A complete description of the four elements engaged in the analogy is not necessarily possible (consider for instance the example "sheep:lamb::cow:x" in the domain of animals).

These differences are well-known and are the core of methods such as SMT or connectionist models. Our purpose is not to give a new or better solutions then those already proposed, but to find a common point in them and exploit it with the same idea as for Hofstadter's analogies.

From the description language, we keep the idea of a memory division and of a factorization of representation based on memory assignments. The hierarchical description of instructions in the form of a tree is kept, as well as the delimited and blank-separated code.

5.1.2 Description Length and Memory Factorization

The structured domains can be described in a Prolog-like way. This approach is consistent with the differences outlined earlier, in particular the non-linearity. In this descriptive framework, an object is then defined by a list of predicates in which it is involved.

Structure Mapping Theory suggests that the analogical mapping has to be done from the general structure of these predicates rather than the semantic of the predicates or the instance of objects. This choice corresponds to a specific use of memory, which imposes the use of more than one parameter (which can be denoted by `?1`, `?2`...). The order of parameters after the declaration is then important.

We observe that in both worlds (Hofstadter's world and structured domains), the memory is used to encode low level inter- and intra-domain structures. In Hofstadter's micro-worlds, these structures were for instance the global generation of the source. In structured domains, they correspond to the set of non-instantiated predicates.

Memory plays a fundamental role of factorization: It avoids the description of redundant features by reusing the same structures. The instantiation of objects from the structures is done in a final step and requires providing extra-information to transform an abstract structure into an object.

These introductory observations provide two indications in the direction of a general model of analogy:

1. Analogical reasoning involves a **factorization of common structures** shared by the problems and the solutions, but also by the source and the target.

2. The description of objects is not directly stored in memory, but reconstructed from the structures stored in memory. Their description relies then on parameters only.

The first idea requires to formalize the notion of "factorization". This formalization will be done in Section 5.2 with the introduction of the notion of Kolmogorov complexity. The second idea will be developed in Section 5.3 with the introduction of the descriptive model initially proposed by (Cornuéjols and Ales-Bianchetti, 1998).

5.2 Descriptive Graphical Models

In this section, we formalize the notion of description length used previously. We will introduce the notion of Kolmogorov complexity and will discuss its relevance in artificial learning. Based on this notion, we will present a class of Turing machines, called *Descriptive Graphical Models*, which can be seen as a generalization of Probabilistic Graphical Models to non-probabilistic settings.

5.2.1 Description Length and Kolmogorov Complexity

The notion of description length described previously has its formal counterpart: *Kolmogorov complexity*. Complexity provides a measure of how complex the generation of an object is. For instance, the sequence `1111111111` is intuitively considered simpler than `1010011101` since it can be generated by a "short" program (repeat 10 times character `1`). We propose to give here an introductory overview of Kolmogorov complexity and some of its properties of interest. For a more precise and complete overview of complexity, we refer the reader to (Li and Vitányi, 2008), from which the main notions, definitions and results given in this subsection are taken.

In the following, we will use the standard asymptotic notation $g(n) = \mathcal{O}(f(n))$ if there are two constants c and n_0 such that $|f(n)| \leq c|g(n)|$ for all $n \geq n_0$.

Consider a countably infinite set of objects $\mathcal{S}$. Object $x \in \mathcal{S}$ can be identified with its index $n(x)$ or by any reordering function $\phi : \mathbb{N} \to \mathbb{N}$. Such a function ϕ can generate the object x if there exists p such that $\phi(p) = n(x)$. The complexity is defined as the length of the shortest p satisfying this property.

A particular case is observed when the specifying method ϕ is *partial recursive*. A function $\phi : \mathbb{N} \to \mathbb{N}$ is called partial recursive if its output $\phi(p)$ corresponds to the

output of a given Turing machine after its execution with input p. In this case, the specifying function corresponds to a Turing machine and p can be interpreted as a program.

We introduce a bijective recursive function $\langle .,. \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, called pairing function. The existence of such a mapping can be shown, for instance using the coding $\bar{x}$ of a string x defined by

$$\bar{x} = \underbrace{11\ldots1}_{l(x) \text{ times}} 0x$$

which corresponds to the concatenation of a '1' repeated $l(x)$ times, a '0' and finally the string x. The mapping is then defined by $\langle x, y \rangle = \bar{x}y$. This pairing will be called standard pairing. It is one among all possible pairing functions, and in particular it is not optimal (in terms of the length of produced outputs).

Based on these notions, we introduce the *plain complexity*:

Definition 6. *Let x, y, p be natural numbers. For any partial recursive function ϕ, we define the complexity C_ϕ of x conditional to y as:*

$$C_\phi(x|y) = \min\{l(p) : \phi(\langle y, p \rangle) = x\} \tag{5.1}$$

and $C_\phi(x|y) = \infty$ if there is no such p. When $y = \epsilon$ (empty input), we simply note unconditional complexity by $C_\phi(x)$.

This definition of complexity is the most intuitive but has several limitations that encourages to consider slight modifications. One of the major drawbacks of plain complexity is its non-subadditivity: Given x and y and the standard bijection $\langle .,. \rangle$, the joint complexity $C(x, y) = C(\langle x, y \rangle)$ does not satisfy the intuitive property $C(x, y) \leq C(x) + C(y) + \mathcal{O}(1)$. The problem is that the joint complexity designates the length of the shortest machine that computes x and y, as well as *a way to tell them apart*. It can be shown that:

$$C(x, y) \leq C(x) + C(y) + 2\log(\min(C(x), C(y))) \tag{5.2}$$

An alternative to this problem (and others that are not discussed here) is the use of *prefix complexity*, denoted $K(.)$. Prefix complexity is defined as a restriction of plain complexity to partial recursive prefix functions:

Definition 7. *A partial recursive prefix function $\phi : \{0,1\}^* \to \mathbb{N}$ is a partial recursive function such that if $\phi(p) < \infty$ and $\phi(q) < \infty$, then p is not a proper prefix of q.*

If ϕ is a partial recursive prefix function, the quantity C_ϕ is denoted by K_ϕ and is called prefix complexity. An alternative definition is often used in practice. Consider a Turing machine made up of three separate tapes such that:

1. The first tape (called *input tape*) is one-way infinite and one-directional (writing is possible only from the left to the right).
2. The second tape (called *work tape*) is two-ways infinite and two-directional .
3. The third tape (called *output tape*) is one-way infinite and one-directional.

Initially, both work and output tapes are empty, and the input tape contains the input sequence. The output corresponds to the content of the output tape when the

Plain complexity C	**Prefix complexity** K
$C(x\|y) \leq K(x\|y) \leq C(x\|y) + 2\log C(x\|y)$	
$C(x) \leq \|x\| + \mathcal{O}(1)$	$K(x) \leq \|x\| + \mathcal{O}(K(\|x\|))$
$C(x,y) \leq C(x) + C(y) + 2\log(\min(C(x), C(y)))$	$K(x,y) \leq K(x) + K(y) + \mathcal{O}(1)$
$C(x\|y) \leq C(x) + \mathcal{O}(1)$	$K(x\|y) \leq K(x) + \mathcal{O}(1)$
$C(x\|y,z) \leq C(x\|y) + \mathcal{O}(1)$	$K(x\|y,z) \leq K(x\|y) + \mathcal{O}(1)$
$C(x\|x,z) = \mathcal{O}(1)$	$K(x\|x,z) = \mathcal{O}(1)$

TABLE 5.1: Comparison of plain and prefix complexities.

machine halts. When the input tape contains only 0's and 1's (no blank), the machine is called *self-delimiting*. It can be shown every partial recursive prefix function is computed by a self-delimiting machine, and that every self-delimiting machine computes a partial-recursive function.

The self-delimited nature of programs used by prefix complexity simplifies the concatenation, since the delimitation between the two descriptions at play is self-contained in the code and does not require to be marked explicitly. Thus, sub-additivity is a direct property of prefix complexity:

$$K(x,y) = K(\langle x,y \rangle) = K(x) + K(y) + \mathcal{O}(1) \tag{5.3}$$

A comparison of some useful properties of plain and prefix complexities is given in Table 5.1.

A major drawback of complexity is its non-computability (which holds for both plain and prefix complexities). The non-computability of complexity is a direct consequence of the non-decidability of the halting problem: in order to compute complexity, it is needed that all programs are tested on the reference UTM. However, it is known that some of them do not halt and that it cannot be known if a program will halt or not. Non-computability of complexity is obviously a major drawback. In practical applications, such as the MML principle exposed in Chapter 2, a restricted set of machines is considered. This restriction, in the context of induction, corresponds intuitively to the choice of an inductive bias, but it makes the invariance theorem unsatisfied[1].

5.2.2 A Key Property: The Chain Rule

The chain rule is a basic property of prefix complexity which states, in short, that $K(x) \leq K(y) + K(x|y)$. The intuition behind this property is that, in order to generate string x, a string y can be generated (term $K(y)$) and used to describe x (term $K(x|y)$).

In the case of prefix complexity, the chain rule expresses as follows:

$$K(x) \leq K(x,y) + \mathcal{O}(1) \leq K(y) + K(x|y) + \mathcal{O}(1) \tag{5.4}$$

In this equation, the constants $\mathcal{O}(1)$ does not depend on variable x but only on the choice of the machine. Moreover, we notice that the status of the variables x and y is not the same. Variable x is defined as the output that has to be described: It is observed and its complexity has to be estimated. On the contrary, string y is used as an intermediate object that is not considered prior to the description of x. In

[1] The invariance theorem (Theorem 2.1.1 in (Li and Vitányi, 2008)) states that there exists a reference UTM for which the complexity is equal to the complexity of any other UTM up to a constant that depends on the machine only.

the following, such variables will be referred to as *latent variables*, as in probabilistic models.

Chain rule can be seen as a major simplification for data compression. The use of an intermediate machine to describe chain x (ie. the machine that computes y first) is the key step in the process: In some cases, generating y has a lower cost than describing x by itself. In the description of x, latent variable y plays the role of a parameter that can be estimated. Inequality 5.4 holds for all y, thus, in particular, it holds for the value of y which minimizes the right-hand size.

In the following, we will use a graphical representation for chain rule: We use a directed graph, the nodes of which correspond to the variables x and y and the edge directed from node y to x displays the orientation of the description (x is described with the help of y, and not the converse). This graph is represented in Figure 5.1. In this representation, the gray node means that the variable is an observation and the white node stands for latent variables.

FIGURE 5.1: Graphical representation of chain rule

The graphical representation of Figure 5.1 corresponds to the description of a set of Turing machines (or equivalently a class of partial recursive functions): All machines in this set are the composition of a first machine that computes y and of a second machine that takes y as input and computes x. This set is parametric and is associated to a global *description length* $K(y) + K(x|y)$. *Descriptive Graphical Models* are a generalization of this class with more complex generative processes.

5.2.3 Defining Graphical Models

A *directed graph* $\mathcal{G}$ is defined as a pair (V, E) where V is a set of elements (called *vertices*, or *nodes*) and $E \subseteq V^2$ is a set of ordered pairs of vertices (called *edges*). A directed graph is called *acyclic* if there is no path of length $l \geq 1$ in the set of edges that link a vertex to itself. Directed acyclic graphs (DAG) are used to define Descriptive Graphical Models.

Definition 8. *Consider $(x_1, \ldots, x_n)$ an ordered set of n variables. Let $\mathcal{G} = (V, E)$ be a directed acyclic graph such that there exists a one-to-one mapping between V and the set of variables. We build a set $T_\mathcal{G}$ of Turing machines by composition of sub-machines as follows: Each node is built by a submachine that takes the parent nodes as input. We call $T_\mathcal{G}$ the descriptive graphical model associated to $\mathcal{G}$.*

In this definition, a machine in $T_\mathcal{G}$ is made up of intermediate sub-machines that compute intermediate values that are taken as input by the others. A descriptive graphical model $T_\mathcal{G}$ is associated to a quantity called *program length* and denoted by $l(T_\mathcal{G})$ (or $l(\mathcal{G})$) defined by:

$$l(T_\mathcal{G}) = \sum_{i=1}^{n} K(x_i|x_{\pi_i}) \tag{5.5}$$

where π_i designates the list of parents of node x_i (if π_i is empty, we define $K(x_i|x_{\pi_i})$ as being equal to $K(x_i)$). From this definition, and using the chain rule, it follows that the program length is an upper-bound of the complexity of observed values for the variables.

Proposition 2. *Consider a DAG $\mathcal{G} = (V, E)$. If $V' \subseteq V$ is a subset of vertices and $x_{V'}$ designates the self-delimited list of x_i with $i \in V'$, then $K(x_{V'}) \leq l(T_{\mathcal{G}}) + \mathcal{O}(1)$.*

In the following, we will use the notations introduced for the chain rule: In the graphical representation, observed variables are displayed with gray nodes and latent variables are displayed with white nodes. Besides, we use the plate notation for the factorization of representations (see Figure 5.2).

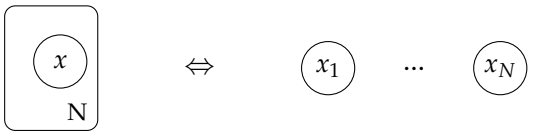


FIGURE 5.2: Plate representation.

5.2.4 Machine Restriction

As mentioned in the presentation of complexity, the function $K(.)$ is not computable and descriptive graphical models offer a valid upper-bound for complexity. A DGM corresponds to a valid description of the observations and its description length is necessarily greater than the length of the shortest description program.

Instead of considering an upper-bound for complexity, we propose to define a **restricted** universal Turing machine that can emulate only a subset of the complete set of Turing machines. In particular, in the context of this chapter, we denote $\phi_{\mathcal{G}}$ the partial recursive prefix function that emulates all Turing machines in $T_{\mathcal{G}}$.

In case the complexities $K(x_i|x_{\pi_i})$ are defined in a simple way (eg. parametric), a procedure **can** exist to find the machine with the lowest description length and the function $K_{\phi_{\mathcal{G}}}(.)$ can be computable. We insist on the fact that the machine $\phi_{\mathcal{G}}$ is **not universal** and that the complexity obtained by this mean is only a computable approximation. In the following, we will refer to this reference machine when considering complexity.

Note however that the choice of a restricted (non universal) Turing machine for the reference of complexity weakens the theory of Kolmogorov complexity, in particular by making the invariance theorem wrong. This theorem states the existence of an additively optimal universal Turing machine, hence a UTM ϕ_0 such that for all x and all UTM ϕ, the complexity $K_{\phi_0}(x) \leq K_{\phi} + c_{\phi}$ where c_{ϕ} is a constant that depends only on machine ϕ. Even if having the invariance theorem not valid makes complexity a weaker theoretical tool, the choice of a restriction is required to get a computable variant. Notice that we will not need the invariance theorem in the remainder of this thesis.

5.2.5 Discussion: DGM and PGM

The definition of DGM is very close to the definition of another class of machine Turing models, the *Probabilistic Graphical Models* (PGM) (Sucar, 2015). Probabilistic graphical models offer a compact representation of probability distributions by taking advantage of probability theory and graph theory.

In PGM, a directed edge is interpreted as an application of Bayes rule and represents conditional probability. While in DGM a graph is associated to a description

length score, the graph used in PGM defines a likelihood function:

$$\log p(x_1, \ldots, x_n) = \sum_{i=1}^{n} \log p(x_i | x_{\pi_i}) \tag{5.6}$$

which is highly similar to the function defined in Equation 5.5.

The analogy between DGM and PGM is not a simple coincidence. Intuitively, a descriptive graphical model can be seen as the description of a process generating strings. A priori, this process is thought to be deterministic, but a random procedure can also be considered, in which case the relative complexities correspond to the description of "randomness" (more formally, a description of the probability distribution).

More formally, an explicit link exists between complexity and probabilities. If μ is an upper semi-computable[2] probability distribution, then $K(x) \leq K(\mu) - \log \mu(x)$. This observation implies that probabilistic graphical models are a particular case of descriptive graphical models, where the data generating machine is based on a probability distribution.

The proximity between these two representations are an invitation to reuse the properties of PGM and interpret them in terms of DGM.

5.2.6 Algorithmic independence

A major interest of Probabilistic Graphical Models is the use of graph theory to get probabilistic properties of random variables, especially regarding independence. In probability theory, two variables are independent if their joint distribution can be factorized as the product of their probability: $\log p(A, B) = \log p(A) + \log p(B)$. An equivalent definition is proposed for *algorithmic independence*, ie. independence of variables in terms of their complexity.

Definition 9. *Variables $x_1, \ldots, x_N$ are said to be algorithmically independent if*

$$K(x_1, \ldots, x_N) = \sum_{i=1}^{N} K(x_i) + \mathcal{O}(1) \tag{5.7}$$

In the same way, variables $x_1, \ldots, x_N$ are said to be algorithmically independent conditionally to variable y if

$$K(x_1, \ldots, x_N | y) = \sum_{i=1}^{N} K(x_i | y) + \mathcal{O}(1) \tag{5.8}$$

This definition means that the value of one of the variables x_i cannot be used (neither entirely nor partially) for the generation of another variable.

Results on independence of random variables in PGMs are now well-known. Despite an apparent relatedness, these results cannot be directly adapted. As an illustration, we consider the following graph as a case study (Figure 5.3):

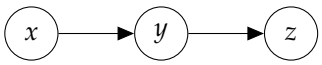


FIGURE 5.3: Elementary graphical model $\mathcal{G}_1$ for independence.

[2] A partial function $f : \mathbb{N} \to \mathbb{R}$ is upper semi-computable if there exists a computable function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that $\lim_{k \to \infty} g(x, k) = f(x)$ and for all $x, k \mathbb{N}^2, \phi(x, k+1) \leq \phi(x, k)$.

In this graphical model, we have conditional independence of Z from X given Y in the probabilistic setting. Do we also have algorithmic independence in the corresponding DGM? The justification for the probabilistic independence is based on Bayes rule which implies that $p(X, Z|Y) = p(X, Y, Z)/p(Y) = p(X|Y)p(Z|Y)$. The same equality does not hold in terms of complexity for a general universal partial recursive prefix function ϕ (which is not the restriction to the graphical model) since the chain rule states that:

$$K(X, Z|Y) \geq K(X, Y, Z) - K(Y) - \mathcal{O}(1) \tag{5.9}$$

In this case, we cannot use the property $K(X, Y, Z) \leq K(X) + K(Y|X) + K(Z|Y) + \mathcal{O}(1)$ since we are considering a lower-bound.

With the universal partial recursive prefix function $\phi_{\mathcal{G}}$ relative to descriptive model $T_{\mathcal{G}}$, the inequality can be simplified since $K(X, Y, Z) = K(X) + K(Y|X) + K(Z|Y)$ and $K(X, Y) = K(X) + K(Y|X)$. It comes that $K(X, Z|Y) \geq K(X|Y) + K(Z|Y) - \mathcal{O}(1)$. On the other hand, it can be shown that $K(X, Z|Y) \leq K(X|Y) + K(Z|Y) + \mathcal{O}(1)$. Thus, $|K(X|Y) + K(Z|Y) - K(X, Z|Y)| \leq \mathcal{O}(1)$, which proves the conditional independence.

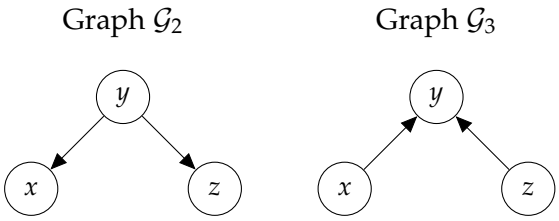


FIGURE 5.4: Two elementary graphical models for independence.

Two other canonical graphs are exposed in Figure 5.4. Such as in the theory of probabilistic graphical models, it can be shown that, in graph $\mathcal{G}_2$, X and Y are independent conditionally to Y, and in graph $\mathcal{G}_3$, X and Z are independent.

Based on these three paradigmatic graphs, relations of algorithmic independence in any graph can be found. Using the results found for the three canonical graphs, it is possible to show that, given a graph, Bayes-ball algorithm (Shachter, 1998) can exhibit the whole list of conditional independence. Other results similar to the ones of PGM can also be stated: d-separation and Hammersley Clifford theorem (Besag, 1974). This theorem states the equivalence between the class $T_{\mathcal{G}}$ and the class of descriptive models satisfying the same conditional independence properties.

5.2.7 Inference

Until now, we have considered that the graph $\mathcal{G}$ was given and the described properties were relative to the fixed graph. It has been shown that a graph is associated to a class of Turing machines $T_{\mathcal{G}}$. The question we investigate in this section regards the choice of a specific machine $\mathcal{M} \in T_{\mathcal{G}}$.

Suppose first that all variables in the graph are observed (no latent variable). The purpose is then to evaluate which partial recursive prefix functions correspond to each of the edges in order to provide the *best* description.

In probabilistic terms, the notion of *good* description is directly given by the likelihood function which measures the probability of observing the variables given the model. In our case, we state that a good quality description corresponds to a minimum complexity description. This assumption is called *minimum description length*

principle (Rissanen, 1978) and corresponds to a formalization of Ockham's razor principle. As exposed in Chapter 2, this principle is used, in its strictest form, to choose a class of models but not to select one model in particular inside this class. A weaker form, called *crude MDL* and refined in the idea of the *minimum message length (MML) principle* (Wallace and Boulton, 1968) offers a way to assess the model directly. In the context of this thesis, we will use either the terms (crude) MDL or MML to refer to this principle.

In a parametric case, we consider that the machine producing variable x_i is parametered by θ_i and is denoted ϕ_{θ_i}. In this case, the inference algorithm is given by:

$$\theta_1^*, \ldots, \theta_n^* = \arg \min_{\theta_1,\ldots,\theta_n} \sum_{i=1}^{n} K_{\phi_{\theta_i}}(x_i|x_{\pi_i}) \tag{5.10}$$

Practical algorithms to effectively solve minimization problem 5.10 depend on the nature of parameters θ_i. The techniques are the same as the likelihood maximization algorithms for completely observed directed probabilistic graphical models.

When latent variables are present in the graph, Equation 5.10 has to be slightly adapted in order to include the minimization over the latent variables. In a way, latent variables can be interpreted as parameters here. For instance, consider the simple graph of Figure 5.1. This graph can be seen as follows: There exists a partial recursive function ϕ such that $\phi_{\theta_1}() = y$ and a partial recursive function ψ such that $\psi_{\theta_2}(y) = x$. It is clear that there also exists a partial recursive prefix function Ψ such that $\Psi_{\theta_2,y}() = x$. Hence, unlike in PGMs, adding latent variables does not modify the way the optimization is performed.

5.3 Minimum Complexity Analogies

In this section, we propose a description of analogies in terms of DGMs. We propose in particular an application to the question of syntactic priming.

5.3.1 A Graphical Model for Analogical Reasoning

Among all methods proposed to solve analogies, one is very similar to the idea proposed in Chapter 4: Using Minimum Description Length (MDL) principle to solve analogical equations. This idea, proposed by (Cornuéjols and Ales-Bianchetti, 1998), is inspired by a comparison of inductive reasoning (in particular with the Empirical Risk Minimization principle, which will be discussed later) and aims to take the specificity of analogy into account. Compared to inductive reasoning, analogies involve one element only from both source and target domains and do not try to perform generalization. Besides the analogical process focuses on the transfer from the source domain to the target domain, and how to apply this transfer to one observation in particular. Based on these observations, the authors propose the use of DGM presented in Figure 5.5.

The model is associated to its description length expressed as follows:

$$K(M_S) + K(X_S|M_S) + K(Y_S|X_S, M_S) + K(M_T|M_S) + K(X_T|M_T) + K(Y_T|X_T, M_T) \tag{5.11}$$

The strength of the model is its use of a *model* as an intermediate object to operate the transfer, rather than transferring the objects' properties directly. It is assumed that these models encode the definition of the domains and are used to describe the objects in a more concise way. Such as suggested in Section 5.1.2, a model is used

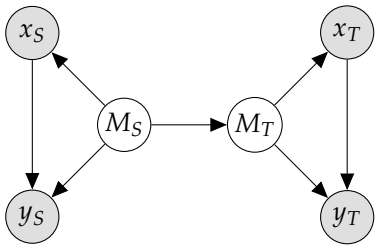


FIGURE 5.5: Model-based DGM for analogical reasoning as suggested by (Cornuéjols and Ales-Bianchetti, 1998).

as a memory factorization tool. Following this observation, we will show that a link exists between our description language and the DGM presented in Figure 5.5.

We consider the analogical equation **ABC:ABD::IJK:x** on Hofstadter's domain. This equation is a very particular case, since source and target domains are equal. We have proposed the following description as an optimal representation of this analogy:

```
let(alphabet, shift, ? sequence, 3),  // Structure definition
  let(mem,, ?, next_block, mem,, ?, last, increment),  // Rule
  mem,,, next_block, mem,, 8;
```

that we decomposed into three steps: Structure definition, rule definition and generation of the analogy. The description length for the global procedure is equal to the sum of the description lengths for each of the three steps.

Step 1 and step 2 provide a definition for the objects in the domain. This definition is incomplete (a parameter has to be given) and stored in memory. Structure definition describes the first term in the analogy (term **A** in the formulation **A:B::C:D**). This structure is stored in memory. It is used to alleviate the description of the association **A:B**: Once instantiated in memory, this source description can be used at any place of the code with lower complexity. Rule definition (step 2) corresponds to the association rule inside a domain (**A:B** and **C:D**). The procedure stored in memory in these two steps can be identified to the *model* M. Their description length is then given by $DL_1 + DL_2 = K(M) + \mathcal{O}(1)$.

In terms of models, two solutions can be identified:

1. The structure described by steps 1 and 2 is considered as a single model. As such, the source and target models are identical and $K(M_T|M_S) = 0$. In this case, terms `mem,,` and `mem,,8` in step 3 are considered to correspond to the construction of observations from the model and $DL_3 = K(X|M) + K(Y|M,X)$, where the variables X, Y and M designate either their source or target equivalents.

2. The structure described by steps 1 and 2 is considered to be a meta-model. In this case, terms `mem,,` and `mem,,8` correspond to instantiations of the meta-model and thus $DL_3 = K(M_S) + K(M_T)$ and data description is direct from the model: $K(X|M) + K(Y|M,X) = 0$.

Based on any of these two solutions, it appears that the proposed code is equivalent to the description of the proposed DGM.

5.3.2 Application: Priming Effect

Syntactic priming is a well-known linguistic behaviour happening when a speaker's syntactic understanding is altered by the prior exposition to a similar structure. As exposed in Section 3.1.2, it has been shown by linguists that this alteration can be observed when the speaker is exposed to the same structure in the same language, in different languages, but also in other domains (for instance music or maths).

This priming effect can be naturally interpreted as a minimum complexity analogy, and especially using the suggested DGM.

Consider the example of syntactic priming from the mathematical domain to the language domain (Scheepers, Sturt, Martin, Myachykov, Teevan, and Viskupova, 2011). An example of such a problem is given in Figure 5.6. The problem is the following. The target problem is a sentence the meaning of which is ambiguous: "I visited a friend of a colleague who lived in Spain". Two structures can be found for this sentence: The relative clause "who lived in Spain" can be attached either to "a friend" (*high attachment*) or to "a colleague" (*low attachment*). The source problem is a mathematical operation, with elementary algebraic operations (additions, substraction and multiplications). Due to the distributive properties of these operations, a simple calculus can be interpreted in the form of a tree. Each node is associated to an operation to be applied on the children nodes. Two operations are detailed in the figure.

A proximity can be found between the source and target problems, which corresponds to the tree representation of the instances. Research in syntactic priming has noticed that a priming is observed in favor of the target having the same tree structure as the source. This observation can be interpreted in terms of analogical reasoning.

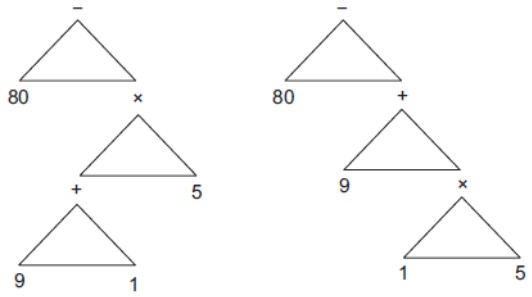
Consider the suggested DGM for analogy. We take as a model the tree structure. In the source domain, the construction of the data X_S (hence the mathematical operation) is obtained by completing the tree, and the output Y_S, which is the result of the operation, is given by simply applying the operations in the order given by the tree. It comes that the complexity $K(X_S|M_S)$ is related to the number of leaves in the tree M_S and the complexity of the numbers involved in the formula.

From the point of view of MDL principle, the model M_T must be easily described by source model M_S but also fit well to the target data X_T, hence the sentence. As seen in Figure 5.6, two trees (hence two models) can explain the sentence equally well, one of them being identical to the source tree. This model is then naturally chosen in a MDL setting, since it leads to the best compression of the description for both source and target data.

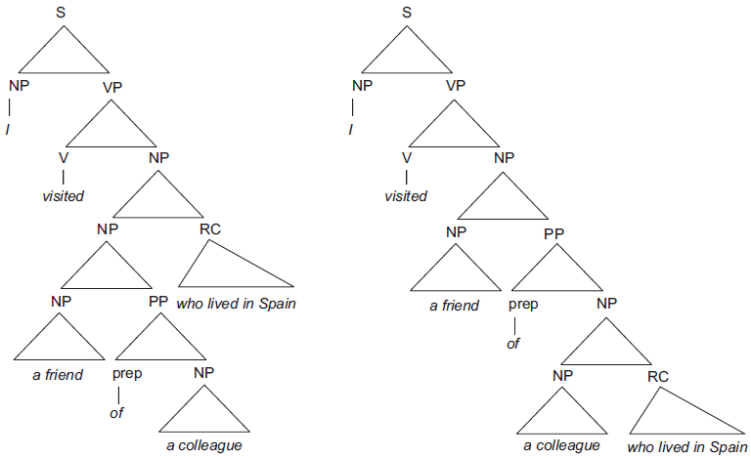
This short analysis of a complex problem has to be seen as an illustrative example of the proposed DGM for analogical reasoning. However, it cannot be seen as a cognitively sound analysis of the phenomenon of syntactic priming. Our model is overly simplified since syntactic priming is quite different from analogy. Syntactic priming cannot be considered as the resolution of a problem, unlike analogy as presented here.

5.4 Conclusion

In this chapter, we formalized the idea of minimum description length analogies that was introduced for Hofstadter's micro-world. We explained that analogy is related to a notion of compression and that compression is measured by a theoretical



(A) Non-ambiguous mathematical structures. The properties of algebraic operators impose a hierarchical structure to the computation.



(B) Ambiguous sentence structure. Two hierarchical structures can be chosen: high-attachment interpretation (on the left) or low-attachment interpretation (on the right).

FIGURE 5.6: Ambiguous and non-ambiguous structures in a mathematical source domain and a language target domain. This figure is taken from (Scheepers, Sturt, Martin, Myachykov, Teevan, and Viskupova, 2011).

tool called Kolmogorov complexity. Complexity measures the length of the shortest program on a Turing machine that can generate an object. Based on this notion and inspired by probabilistic graphical models, we introduced Descriptive Graphical Models (DGM), which can define a general class of Turing machines. We analyzed, in terms of DGM, the model for analogy firstly presented by (Cornuéjols and Ales-Bianchetti, 1998).

In the next chapter, we will consider analogies in geometrical domains. We will explain that minimum complexity analogies require using simple transformations in these domains. When the ambient space is a vector space, the simplest operations are given by additions and substractions, which are naturally defined in the structure of the space. However, we will show that the results obtained in Riemannian manifolds are not that simple and, in particular, are in conflict with the axioms of proportional analogy.

Chapter 6

Geometrical analogies

In previous chapters, we proposed a general model for analogies with the minimum description length principle. The chosen model, inspired by previous works and consistent with the language introduced in Chapter 4, consists in separating the domain description phase and the inter-domain transfer. The transfer is operated through objects called *models*, the role of which is to store the domain knowledge.

In this chapter, we propose to explore the domain of analogies defined on geometric structures such as vector spaces or manifolds. This problem has already been investigated in vector spaces, in particular regarding the theory of analogical proportion. The well-known parallelogram rule states that an analogy can be represented as a parallelogram in the concept space and has been used since the first researches in the domain of analogical reasoning (Rumelhart and Abrahamson, 1973). This rule is particularly simple and satisfies the axioms of analogical proportion. We will give a general expression for it and will think of it in geometrical terms. In particular, we will address the question of what happens when the concept space is curved.

The chapter will be organised as follows: In Section 6.1, we will propose an indirect approach to describe geometrical analogies, inspired by MML principle and we will present parallelogram rule as an application of our approach. In Section 6.2, we will investigate the natural method induced for non-Euclidean spaces and will discuss some interesting properties of such analogies. Lastly, in Section 6.3, we present an application to analogies in Fisher manifolds and discuss a potential application to curved shape spaces.

6.1 Building Analogies in Concept Spaces

In this section, we propose an interpretation of the DGM introduced in Section 5.3.1 for analogies when the four terms are elements of a same space $\mathcal{S}$.

6.1.1 Interpretation of the Parallelogram Rule

Consider an analogy $A : B :: C : D$ where all elements $A, B, C, D \in \mathcal{S}$ are elements of a vector space $\mathcal{S}$. Among all possible relations R on $\mathcal{S}^4$, one of the simplest one is the parallelogram rule, which has been exposed in Section 3.3.1 and represented in Figure 6.1.

How can parallelogram rule be expressed algorithmically? It is clear that a parallelogram can be built given three vectors: the initial position A, the first edge u and the second edge v (it follows that $B = A + u$, $C = A + v$ and $D = A + u + v$). This representation is the most economic representation in terms of number of parameters (no description can be given with 2 parameters or less). In terms of the analogical DGM presented in Section 5.3.1, this description can be expressed as follows:

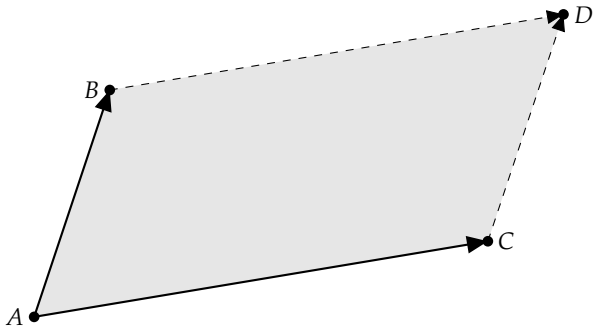


FIGURE 6.1: Illustration of the parallelogram rule on $\mathcal{S} = \mathbb{R}^2$.

- Source model: $M_S = (u, A)$, hence $K(M_S) \leq K(u) + K(A) + \mathcal{O}(1)$
- Source question: $X_S = A$, hence $K(X_S|M_S) = \mathcal{O}(1)$
- Source solution: $Y_S = A + u$, hence $K(Y_S|M_S, X_S) \leq K(+) + \mathcal{O}(1)$
- Model transfer: $M_T = (u, A + v)$, hence $K(M_T|M_S) \leq K(v) + K(+) + \mathcal{O}(1)$
- Target question: $X_T = A + u$, hence $K(X_T|M_T) = \mathcal{O}(1)$
- Target solution: $Y_T = A + u + v$, hence $K(Y_T|M_T, X_T) \leq K(+) + \mathcal{O}(1)$

In this description, the complexity of the addition operator $K(+)$ is a constant (depending on the machine but not on data), but we isolated this term on purpose. We point out the idea that addition is an operation *arbitrarily* chosen in this context, but could be replaced by any other operation. Moreover, it appears that we consider two additions of different nature in the process of drawing the parallelogram: The first addition is intra-domain and is involved in the construction of the solution Y based on the question X and the model M, while the second addition is cross-domain and is used to describe target domain from source domain. In a general context, we will call these operations *transport functions*.

Among all possible transport functions, the choice of the addition operation in vector spaces is motivated by the simplicity of this choice. Addition is the only self-defined operation in a vector space, which makes a perfectly valid candidate to be the minimum complexity transport operator.

6.1.2 General Construction of a Parallelogram

Consider now a general space $\mathcal{S}$. We propose a generalization of the parallelogram method exposed above for vector spaces.

To do so, we consider first the complexity term $K(M_T|M_S)$. We recall that, in DGMs, this term measures the description length of a machine taking M_S as input and returning M_T as output. This machine is associated to a partial recursive prefix function ϕ. With this function ϕ we have:

$$K(M_T|M_S) = K_\phi(M_T|M_S) = \min\{l(p)|\phi(\langle M_S, p\rangle) = M_T\} \tag{6.1}$$

with the same notations as introduced in Chapter 5. The question now is how to evaluate this complexity, and in particular, how to choose a restricted universal partial recursive function which makes this term computable and intuitive.

In the discussed case of a vector space, we defined the description model as the delimited concatenation of the initial position and the direction: $M = \langle P, u \rangle$ (where $P \in \mathcal{S}$ is a point in the vector space). Based on this choice, we proposed to consider that transfer function ϕ is defined by:

$$\phi\left(\langle\langle u, P\rangle, v\rangle\right) = \langle u, P + v\rangle \tag{6.2}$$

where the programs p are indexed by the description of vector v. When the input of ϕ does not fit the proposed format, the machine does not halt. With this definition, we get the complexity term $K(M_T|M_S) = K(v) + K(+)$ defined previously.

In practice, one might wish to consider other operators than the addition, in which case the new position can be defined by a function $\omega(P, u, \pi)$ where $\langle P, u\rangle = M$ and π is a parameter. The transfer function is then defined by:

$$\phi\left(\langle\langle u, P\rangle, \langle \omega, \pi\rangle\rangle\right) = \langle u, \omega(P, u, \pi)\rangle \tag{6.3}$$

which induces a complexity of the form $K(M_T|M_S) = K(\omega) + K(\pi)$. This definition of ϕ is particularly inspired by the parallelogram rule and has a major drawback: It implies that the "direction" (parameter v) is not modified in the transfer process, which is not realistic from a more general point of view. Thus, we propose a more general definition which will be considered in the remainder of this chapter:

$$\phi\left(\langle\langle u, P\rangle, \langle\langle \omega_1, \pi_1\rangle, \langle \omega_2, \pi_2\rangle\rangle\rangle\right) = \langle \omega_1(P, u, \pi_1), \omega_2(P, u, \pi_2)\rangle \tag{6.4}$$

with corresponding complexity $K(M_T|M_S) = K(\omega_1) + K(\pi_1) + K(\omega_2) + K(\pi_2)$. We recall that ω_1 is a transformation operator parameterized by π_1 which depicts the transformation of the problem into a solution. Similarly, ω_2 is a transformation operator parameterized by π_2 which depicts the transformation of one problem into another problem.

The choice of the operators ω_1 and ω_2 depends on the nature of the problem and, in particular on the space of interest. As pointed out, addition is the only natural basic operation of a vector space, which explains that parallelogram rule is the first model appearing in the case of analogies in a vector space. The purpose of this chapter is to investigate a natural transformation in spaces of different nature: Riemannian manifolds.

Before pursuing to these considerations, we would like to make two fundamental remarks on the presented methodology.

The first remark regards the choice of the operators. We claimed that the operators have to be chosen with respect to the nature of the space $\mathcal{S}$ and that the most simple operator defined on $\mathcal{S}$ is the optimal operator. This assumption is not completely accurate: In fact, there is not one single possible operator per space, but a large set of possible operators. Changing from one space to another requires to adapt the definition of Ω, the enumeration of possible operators. For simplicity purposes, we only consider the most straightforward solution here, since other operators might require very large description complexity while elementary operators are naturally given by the structure of the considered space.

Secondly, we would like to point out that, given the partial recursive function ϕ as defined in Equation 6.4, knowing space $\mathcal{S}$ becomes optional. We do not use the space directly, but a way to generate elements in it on the fly through operators ω_1 and ω_2. This property is interesting since it allows one to consider analogies in

spaces that can be built in a procedural way and do not need any extensional definition (hence a definition of all of its elements).

6.2 Non-Euclidean Analogies

In this section, we propose to study natural transfer operators ω_1 and ω_2 when the space of interest is a Riemannian manifold. We first present an intuition with the trivial example of analogies on the sphere, then we present a short reminder of differential geometry that are necessary before presenting the method itself. Finally, we discuss the possibility to define proportional analogies on a differential manifold. Since the space $\mathcal{S}$ is a manifold in this section, we will use the notation $\mathcal{M}$ to designate it.

6.2.1 Intuition: Analogies on the Sphere

In order to understand the ideas at play, we propose to consider the example of analogies on a sphere. We denote by $\mathbb{S}^2$ the sphere defined as the subset of $\mathbb{R}^3$ defined as $\mathbb{S}^2 = \{x | x_1^2 + x_2^2 + x_3^2 = 1\}$. The sphere can be shown to be a differential manifold, and is obviously not Euclidean.

We consider three points A, B and C on the sphere and we try to solve the analogical equation $A : B :: C : x$. In the context of this example, we will consider three specific points, but the conclusions we will draw would be the same for any 3 points which are "not aligned" (in the sense that the third point is not on the shortest path between the two others).

In order to solve this analogy, an intuitive idea would be to apply the same procedure as described by the parallelogram rule. Imagine first that the three points A, B and C are very close. On Earth, it is possible to use the parallelogram rule directly on a small scale: Since Earth is locally flat, we can consider the floor as a vector space and apply a parallelogram rule by walking from A to C by keeping in mind the direction to go to B from A.

The same procedure can be used when the three points are very distant. In mathematical terms, we can formulate this procedure as a three steps method:

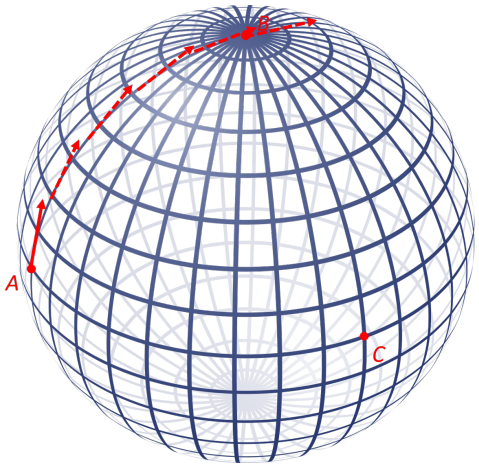
1. **Direction finding:** Estimation of the direction d to reach B from A following a geodesic (ie. a path of minimal length).

2. **Parallel transport:** The direction vector is transported along the geodesic from A to C.

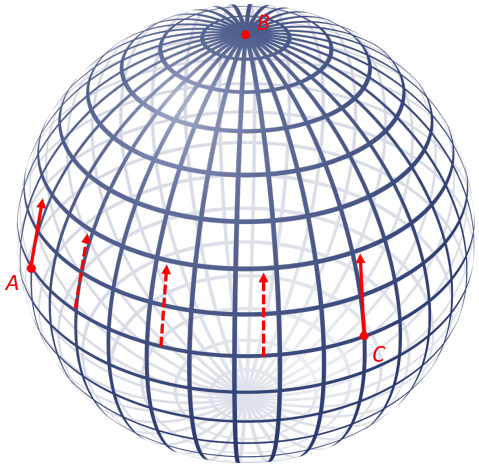
3. **Geodesic shooting:** Point D is reached by following the transported direction d' from point C.

We consider for instance the case where B correspond to the North pole and A and C are located on the equator. For simplicity purposes, we also suppose that the angle between the locations of A and C in the equator plane is $\pi/2$. The solution to this analogy using the proposed method is shown in Figure 6.2.

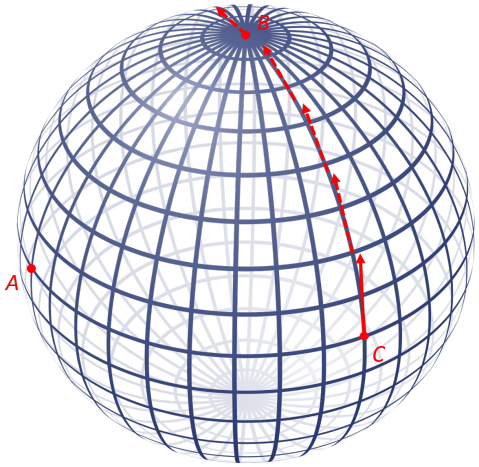
The steps can be intuitively explained as follows. The first step consists in finding the shortest path from A to B: this path is characterized by the initial direction, which is mathematically encoded by a vector in the tangent space. The second step is of a different nature: The idea is to go along the shortest path from A to C while maintaining the initial direction vector "in the same direction" (the exact mathematical terminology will be precised in the next section). As an illustration of this, the



(A) Step 1: Direction finding.



(B) Step 2: Parallel transport.



(C) Step 3: Geodesic shooting.

FIGURE 6.2: Step by step resolution of the analogical equation $A : B :: C : x$ on the sphere $\mathbb{S}^2$. The solution found is $x = B$.

second step can be seen as walking from A to C while maintaining one's nose "parallel" from one position to the other. The shortest path from A to C in our example is the equator and the initial direction is the vector pointing toward the North pole: Hence, step 2 is similar to walking from A to C along the equator with the nose pointing toward the North pole at any time. The third step consists in following the transported initial direction the same way as done to join B from A in step 1.

Using this procedure, the solution of the analogical equation $A : B :: C : x$ is $x = B$. With the same procedure applied to the analogical equation $A : C :: B : x$, we obtain the solution $x = C$, which is in contradiction with the *exchange of the means* property of analogical proportion. However, we can easily verify that the other properties are verified:

- Symmetry of the 'as' relation: $C : B :: A : B$ and $B : C :: A : C$
- Determinism: the solution of $A : A :: B : x$ is $x = B$

In the following, we will call a **Non-Euclidean Analogy** an analogy which satisfies the symmetry of the 'as' relation and the determinism property, but not necessarily the exchange of the means. An analogical proportion is a more constrained case of a non-Euclidean analogy.

6.2.2 Non-Euclidean Analogies

Following the ideas developed in Section 6.2.1, we propose the following definition for a non-Euclidean analogical proportion:

Definition 10. *A non-Euclidean analogy on a set X is a relation on X^4 such that, for every 4-uple $(A, B, C, D) \in X^4$, the following properties are observed:*

- *Symmetry of the 'as' relation:* $R(A, B, C, D) \Leftrightarrow R(C, D, A, B)$
- *Determinism:* $R(A, B, A, x) \Rightarrow x = B$

The second axiom (determinism) is slightly different from the original analogical proportion. For analogical proportions, two possible implications could be used to characterize determinism, the second characterization being the implication

$$R(A, A, B, x) \Rightarrow x = B$$

One being true, the other is a consequence of the first one. In non-Euclidean analogy, these two implications are not equivalent anymore.

Removing the exchange of the means from the definition of an analogy actually makes sense. The symmetry of the means operates in the cross-domain dimension of the analogy: Keeping this observation in mind, the symmetry of the means seems to be a natural property. In practice, it can be observed that the property is perceived as less natural in many examples. Consider for instance the well-known analogy "The sun is to the planets as the nucleus is to the electrons". The symmetrized version of this analogy is "The sun is to the nucleus as the planets are to the electrons", which is less understandable than the original analogy.

Moreover, many examples of common analogies can be found that do not satisfy this property. For instance, the analogy "Cuba is to the USA as North Korea is to China", which is based on a comparison of politics and geographic proximity, while the symmetrized analogy "Cuba is to North Korea as the USA are to China" does not make sense. In this example, the status of the terms is different: In one direction,

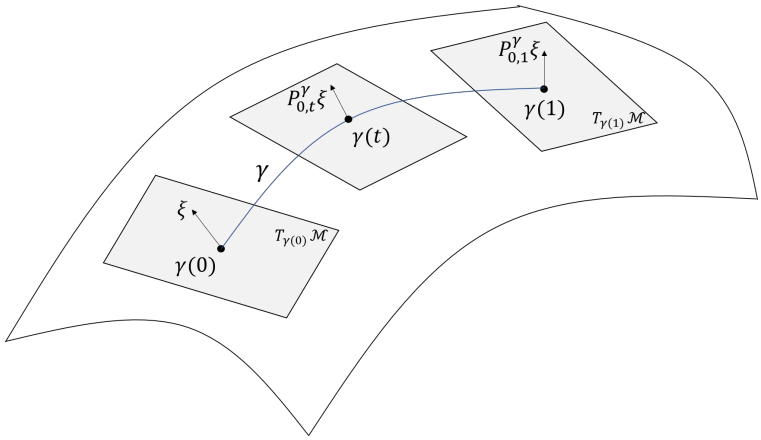


FIGURE 6.3: Illustration of parallel transport on a differential manifold. Vector ξ is transported along a curve γ. At any position t, we have $P^{\gamma}_{0,t}\xi \in T_{\gamma(t)}\mathcal{M}$.

the analogy is based on a political comparison, while in the other direction it is based on a large-scale geographical comparison. The nature of these two domains is not the same and does not have the same weight in the analogy. This intuition of a directional weighting is coherent with the model of non-Euclidean manifolds.

6.2.3 Reminder: Riemannian Geometry

In order to understand our method, we have to introduce some standard definitions of Riemannian geometry. The proposed definitions are not entirely detailed: we refer interested readers to standard references (Boothby, 1986) for more details.

A *topological manifold* of dimension d is a connected paracompact Hausdorff space for which every point has an open neighborhod U that is homeomorphic to an open subset of $\mathbb{R}^d$ (such a homeomorphism is called a *chart*). A manifold is called *differentiable* when the chart transitions are differentiable, which means that the mapping from one chart representation to another is smooth.

A tangent vector ξ_x to a manifold $\mathcal{M}$ at point x can be defined as the equivalence class of differentiable curves γ such that $\gamma(0) = x$ modulo a first-order contact condition between curves. It can be interpreted as a "direction" from the point x (which only makes sense when $\mathcal{M}$ is a subset of a vector space). The set of all tangent vectors to $\mathcal{M}$ at x is denoted $T_x\mathcal{M}$ and called tangent space to $\mathcal{M}$ at x. The tangent space can be shown to have a vector space structure. When the tangent spaces $T_x\mathcal{M}$ are equipped with an inner-product g_x which varies smoothly from point to point, $\mathcal{M}$ is called a *Riemannian manifold*.

We define a connection ∇ as a mapping $C^\infty(T\mathcal{M}) \times C^\infty(T\mathcal{M}) \to C^\infty(T\mathcal{M})$ satisfying three properties that are not detailed here: A connection can be seen as a directional derivative of vector fields over the tangent space. A special connection, called the *Levi-Civita connection*, is defined as an intrinsic property of the Riemannian manifold which depends on its metric g only.

These tools are used to define two notions that are fundamental in our interpretation of non-Euclidean analogies: parallel transport and geodesics. Let $(\mathcal{M}, g)$ be a Riemannian manifold and let $\gamma : [0,1] \to \mathcal{M}$ be a smooth curve on $\mathcal{M}$. The curve γ is called a geodesic if $\nabla_{\dot\gamma}\dot\gamma = 0$. This definition of a geodesic means that the initial direction remains auto-parallel when being transported all along the curve. This notion can be shown to correspond to a minimum length curve between two points.

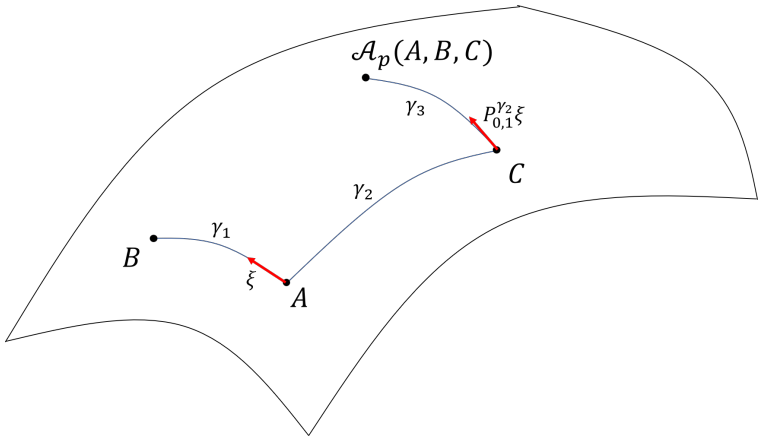


FIGURE 6.4: Parallelogramoid procedure on a Riemannian manifold.

A vector field X along γ is said to be parallel if $\nabla_{\dot{\gamma}} X = 0$. One can define the *parallel transport* as the application $P_{t_0,t}^{\gamma} : T_{\gamma(t_0)}\mathcal{M} \to T_{\gamma(t)}\mathcal{M}$ which maps any vector of the tangent space ξ at point $\gamma(t_0)$ to the corresponding value at $\gamma(t)$ for the parallel vector field X such that $X(\gamma(t_0)) = \xi$ (figure 6.3).

It seems clear that the notions of geodesic and parallel transport are good candidates for operators ω on a Riemannian manifold. This intuition is confirmed by the toy example of the sphere $\mathbb{S}^2$ presented in section 6.2.1.

6.2.4 Non-Euclidean Analogies on Differential Manifolds

Using the notions introduced by differential geometry, we propose now to define a geometric model transfer according to equation 6.4 by defining the models and transfer operators:

- A model is given by a tuple $M = \langle u, P \rangle$ where $P \in \mathcal{M}$ is a point on the manifold and $u \in T_P\mathcal{M}$ is a tangent vector to the manifold at point P.
- We define ω_1 as the parallel transport of vector $u \in T_P\mathcal{M}$ along a geodesic curve of length 1. A geodesic curve is entirely defined by an initial position and an initial celerity vector: Consequently, operator ω_1 is associated to a unique parameter $v \in T_P\mathcal{M}$.
- We define ω_2 as the exponential map at point P with tangent vector v, ie. the parallel transport of a vector from point P along the geodesic drawn with initial direction v. Operator ω_2 is associated to the same parameter $v \in T_P\mathcal{M}$ as operator ω_1.

The transformation induced by these definitions of operators ω_1 and ω_2 will be called the parallelogramoid algorithm. In the following, we present its formal definition and its properties.

Definition 11. *The parallelogramoid algorithm $\mathcal{A}_p : \mathcal{M}^3 \mapsto \mathcal{M}$ is defined as follows. Consider $(A, B, C) \in \mathcal{M}^3$. Let $\gamma_1 : [0,1] \to \mathcal{M}$ be a geodesic curve such that $\gamma_1(0) = A$ and $\gamma_1(1) = B$. Let $\xi \in T_A\mathcal{M}$ such that $\xi = \dot{\gamma}_1(0)$. Consider a geodesic curve $\gamma_2 : [0,1] \to \mathcal{M}$ such that $\gamma_2(0) = A$ and $\gamma_2(1) = C$. Let γ_3 be the geodesic defined by $\gamma_3(0) = C$ and $\dot{\gamma}_3(0) = P_{0,1}^{\gamma_2}\xi$. Then $\mathcal{A}_p(A, B, C) = \gamma_3(1)$.*

Algorithm $\mathcal{A}_p$ corresponds to the procedure used in the case of a sphere. In general, the described procedure is not unique: The unicity of tangent vector ξ is not guaranteed. For instance, in the case of the sphere, if A and B correspond to the North and South poles, there exists an infinite number of such vectors ξ.

Theorem 4. *The relation $R(A,B,C,D) \equiv (\mathcal{A}_p(A,B,C) = D)$ defines a non-Euclidean analogy on $\mathcal{M}$.*

Proof. We would like to show that $C : D :: A : B$ (symmetry axiom) is correct with our construction. We use the tilde notation to describe the curves for this analogy. For instance, $\tilde{\gamma_1}$ is the geodesic from C to D, hence $\tilde{\gamma_1} = \gamma_3$. Similarly, $\tilde{\gamma_2} = -\gamma_2$, where $-\gamma$ designates the "opposite curve" (ie. $\tilde{\gamma_2}(s) = \gamma(1-s)$). Since parallel transport is invertible, $\xi = P_{0,1}^{\tilde{\gamma_2}} P_{0,1}^{\gamma_2} \xi$. Thus, $\tilde{\gamma_3}$ is the geodesic curve such that $\tilde{\gamma_3}(0) = A$ and $\dot{\tilde{\gamma_3}}(0) = \xi$ and consequently $\tilde{\gamma_3}(1) = B$. □

In general, the relation does not define a proportional analogy since symmetry of the means does not hold: $\mathcal{A}_p(A,B,C) \neq \mathcal{A}_p(A,C,B)$. We will show that we have equality only for a specific metric, called *Ricci-flat metric*.

When $\mathcal{M} = \mathbb{R}^n$ endowed with the canonical inner-product, the proposed construction can be shown to be equivalent to the usual parallelogram rule, since a geodesic is defined as a straight line and parallel transport over a straight line is a simple translation of the original tangent vector. It can be shown that the converse is almost true: The manifolds for which $\mathcal{A}_p$ designs an analogical proportion have their *Ricci curvature* vanishing at any point.

Theorem 5. *The only Riemannian metrics g such that the relation*

$$R(A,B,C,D) \equiv (\mathcal{A}_p(A,B,C) = D)$$

is an analogical proportion for any A, B and C are Ricci-flat.

Proof. In this demonstration, we will consider the equivalent problem where we are given $A \in \mathcal{M}$ and $\xi_1, \xi_2 \in T_A\mathcal{M}$. With these notations, $B = \gamma_1(1)$ and $C = \gamma_2(1)$ where γ_1 is the geodesic drawn from A with initial vector ξ_1 and γ_2 is the geodesic drawn from A with initial vector ξ_2. Considering an infinitesimal parallelogramoid as defined in Definition 1.1 of (Ollivier, 2011), where δ is the distance between A and B, and ϵ the distance between A and C. Then the distance between C and $D = \mathcal{A}_p(A,B,C)$ is equal to

$$d = \delta\left(1 - \frac{\epsilon^2}{2}K(v,w) + \mathcal{O}(\epsilon^3 + \epsilon^2\delta)\right)$$

where $K(v,w)$ is the sectional curvature in directions (v,w). In the case of analogical proportion, it can be verified that distance d must be equal to δ. Thus, we have necessarily $K(v,w) = 0$ and, by construction of Ricci curvature $Ric(v)$ as the average value of $K(v,w)$ when w runs over the unit sphere, we have the result. □

Obviously, Euclidean spaces endowed with the canonical vector space are Ricci-flat, but there exists other Ricci-flat spaces. A direct consequence of Theorem 5 is that analogical proportions can be defined on some differential manifolds.

6.2.5 Proportional Analogies on Manifolds

In previous sections, we have shown that the intuition of what an analogy can be in a differential manifold leads to a less constrained definition of analogies than the definition of proportional analogy. However, at this point of the chapter, the existence of proportional analogies on a manifold $\mathcal{M}$ remains an open question. The purpose of this section is to discuss the construction of analogical proportions on a manifold.

Let $\mathcal{M}$ be a differential manifold. Our purpose is to design an algorithm to build analogical proportions. We define an algorithm as a function $\mathcal{A} : \mathcal{M}^3 \mapsto \mathcal{M}$.

Definition 12. *An algorithm $\mathcal{A} : \mathcal{M}^3 \mapsto \mathcal{M}$ is said to design an analogical proportion on $\mathcal{M}$ if, for all $(A, B, C) \in \mathcal{M}^3$, the relation $R(A, B, C, D) = (D = \mathcal{A}(A, B, C))$ satisfies the axioms of analogical proportion.*

Definition 12 can be seen as a reverse way to define solutions of analogical equations. If a relation R is an analogical proportion over $\mathcal{M}$ designed by algorithm $\mathcal{A}$, then $x = \mathcal{A}(a, b, c)$ is the unique solution of equation $R(a, b, c, x)$ where x is the variable.

The following proposition offers an alternative characterization of proportion-designing algorithms based on global characteristics.

Proposition 3. *Algorithm $\mathcal{A}$ designs an analogical proportion if and only if the following three conditions hold true for any $(A, B, C) \in \mathcal{M}^3$:*

1. $\mathcal{A}(A, B, A) = B$ *or* $\mathcal{A}(A, A, B) = B$
2. $\mathcal{A}(A, B, C) = \mathcal{A}(A, C, B)$
3. $B = \mathcal{A}(C, \mathcal{A}(A, B, C), A)$

The set of such algorithms on $\mathcal{M}$ is denoted by $\mathbb{A}_{\mathcal{M}}$.

Proof. The proof is a direct consequence of the axioms of analogical proportion. □

In the case where $\mathcal{M}$ is a vector space, it can be easily verified that the parallelogram rule algorithm $\mathcal{A}(A, B, C) = C + B - A$ designs an analogical proportion.

However, it is not the only algorithm to satisfy this property. In Proposition 4, we exhibit a parameterized class of analogical proportion designing algorithms.

Proposition 4. *If $\mathcal{M}$ is a vector space and $f : \mathcal{M} \mapsto \mathcal{M}$ is a bijective mapping, then algorithm $\mathcal{A}_f$ defined by $\mathcal{A}_f(A, B, C) = f^{-1}(f(C) + f(B) - f(A))$ designs analogical proportion.*

It can be noticed that, when f is linear, algorithm $\mathcal{A}_f$ corresponds to the parallelogram rule. For other values of f, algorithm $\mathcal{A}_f$ can define proportions of another nature. An interesting perspective would be to study if these non-trivial proportions on a vector space can be related to analogical proportions on a manifold.

The result of Proposition 4 can be generalized in the case where two spaces are available.

Proposition 5. *Consider E and F two isomorphic sets with $f : E \to F$ a corresponding isomorphism. If $\mathcal{A}_F : F^3 \to F$ designs an analogical proportion on F then algorithm $\mathcal{A}_E : E^3 \to E$ defined by $\mathcal{A}_E(a, b, c) = f^{-1}\left(\mathcal{A}_F(f(a), f(b), f(c))\right)$ designs an analogical proportion on E.*

Using this result, it can be shown that locally defined analogical proportions on a manifold can be related to analogical proportions on vector spaces.

Corollary 1. *Consider a chart $U \subset \mathcal{M}$ isomorph to an open subset $E \subset \mathbb{R}^n$. We call ψ the isomorphism $U \to E$. If $a : E^3 \to E$ designs analogical proportion on E, then algorithm $\mathcal{A} : U^3 \to U$ defined by $\mathcal{A} = \psi^{-1}(a(\psi(A), \psi(B), \psi(C))$ designs an analogical proportion on U.*

Without lack of generality, we can take $E = \mathbb{R}^n$, in which case we know that $\mathbb{A}_E$ is non-empty. Indeed, if E is an open subset of $\mathbb{R}^n$ at point x, one can consider an open disk $D \subset E$ which is homeomorphic to a neighborhood of x. We notice that an open disk on $\mathbb{R}^n$ is homeomorphic to $\mathbb{R}^n$.

As a consequence, analogical proportions can be defined on manifolds that are globally homeomorphic to $\mathbb{R}^n$.

Consider for instance the sphere $\mathbb{S}^2$ minus a point N, arbitrarily chosen to be the North pole of the sphere. This manifold can be shown to be homeomorphic to $\mathbb{R}^2$, using for instance the stereographic projection. However, it comes from topological properties that there exists no continuous bijective mapping between $\mathbb{S}^2$ and $\mathbb{R}^2$.

This property does not imply that no bijection can be drawn between $\mathbb{S}^2$ and $\mathbb{R}^2$: Such bijections exist but cannot be continuous. Consequently, it is possible to define analogical proportions on a sphere. However, there is no direct way to define continuous (and *a fortiori* smooth) bijections, hence to define analogical proportions which fits the geometry of the manifold.

In general, the reasoning that was presented for the sphere can be extended to any manifold, which proves the existence of valid analogical proportions on any manifold.

Theorem 6. *For any manifold $\mathcal{M}$, the set $\mathbb{A}_{\mathcal{M}}$ is non-empty.*

Proof. Consider a finite atlas $\mathcal{A} = \{(U_\alpha, \psi_\alpha) | \alpha \in \{1, \dots, m\}\}$. Such an atlas exists for m large enough. In this definition, U_α corresponds to a domain on $\mathcal{M}$ and $\psi_\alpha : U_\alpha \to \mathcal{B}_n(0,1)$ is an homeomorphism from U_α onto the unitary ball $\mathcal{B}_n(0,1)$ on $\mathbb{R}^n$ (where n is the dimension of $\mathcal{M}$). If we denote $E_k = \{x \in \mathbb{R}^n | 2(k-1) < x_1 < 2(k+1)\}$, one can equivalently extend the mapping ψ_k to be homeomorphisms between U_k and E_k (figure 6.5).

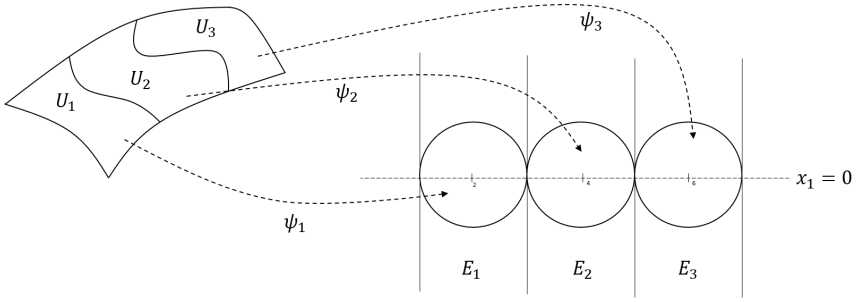


FIGURE 6.5: Construction of a bijective mapping between a manifold $\mathcal{M}$ of dimension n and an open subset of $\mathbb{R}^n$. For simplicity purpose, the subsets U_k are presented as disjoint, which they are not.

We build a function $\psi : \mathcal{M} \to \bigcap_{k=1}^m E_k$ as follows: If $x \in U_k \setminus \bigcap_{i>k} U_i$, then $\psi(x) = \psi_k(x) + e_k$ where e_k is the vector with first component equal to $2k$ and all other components equal to 0. This function defines a bijective mapping. Since

$\bigcap_{k=1}^{m} E_k$ is an open subset of $\mathbb{R}^n$, there exists a bijection $\bigcap_{k=1}^{m} E_k \to \mathbb{R}^n$. The theorem follows from Proposition 5 and the fact that $\mathbb{A}_{\mathbb{R}^n} \neq \emptyset$.

□

Theorem 6 is fundamental since it states the existence of analogical proportions on manifolds, which seems to invalidate the intuitions exposed with the parallelogrammoid method. However, the intuitive "validity" of the existing analogies (and in particular of the analogies produced by the proof) is not clear since they appear to be highly irregular since they are not continuous.

These observations point out a deficiency in the definition of analogical proportion, which comes from its main applicative domains. The definitions of analogical proportion were first designed for applications in character-string domains (Lepage, 2003) and were discussed for applications in other non-continuous domains (Miclet, Bayoudh, and Delhay, 2008) such as analogies between finite sets. Among real continuous applications (hence applications which do not involve a discretization of the continuous space), most are based on parallelogram rule on a vector space. When defining analogical proportions on continuous spaces, a continuity property is also desirable, which is not induced by the definition of analogical proportion. Intuitively, this property makes sense: If two analogical problems are close, it is expected that their solutions will be close as well.

The question of the existence of analogical proportion defining algorithms that are also continuous (in the sense of a function $\mathcal{M}^3 \to \mathcal{M}$) remains open at this step. It is impossible to adapt the proof of Theorem 6 in order to make the mapping continuous. More generally, the result cannot be directly adapted from Proposition 5. The main problem to overcome is the transition from one chart to another in the atlas decomposition of the manifold (see Figure 6.6). The difficulty is to define the result of $R(A, B, C)$ when the three variables do not belong to the same chart.

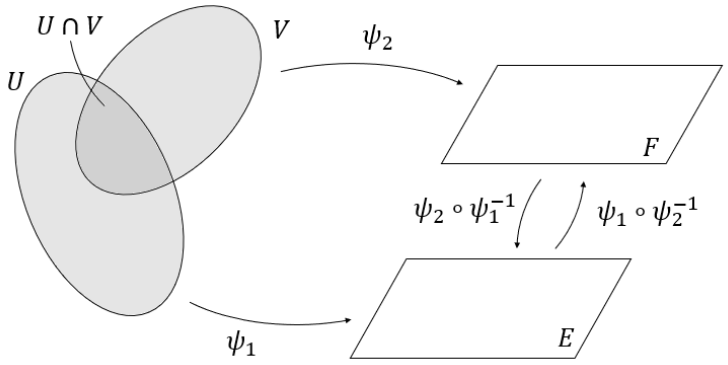


FIGURE 6.6: Chart transition on a manifold.

6.3 Applications

In this section, we discuss applications of the parallelogrammoid procedure in the case of pre-existing manifolds with well-known Riemannian structures.

6.3.1 Non-Euclidean Analogies in Fisher Manifold

6.3.1.1 Fisher Manifold

By definition, a parametric family of probability distributions $(p_\theta)_\theta$ has a natural structure of a differential manifold and, in this context, is called *statistical manifold*. Unless in general a manifold is not associated to a notion of distance or metric, *information geometry* states that there exists only one natural metric for statistical manifolds (Cencov, 2000). This metric, called *Fisher metric*, is defined as follows (Fisher, 1925):

$$g_{ab}(\theta) = \int p(x|\theta) \frac{\partial \log p(x|\theta)}{\partial \theta^a} \frac{\partial \log p(x|\theta)}{\partial \theta^b} dx \tag{6.5}$$

It can be related to the variance of the relative difference between one distribution $p(x|\theta)$ and a neighbour $p(x|\theta + d\theta)$. For a more complete introduction to Fisher manifolds and more precise explanations on the nature of Fisher metric, we refer the reader to (Amari, 2012).

Among all possible statistical manifolds, we focus on the set of normal distributions, denoted by $\mathcal{N}(n)$. A complete description of the geometric nature of $\mathcal{N}(n)$ is given in (Skovgaard, 1984). As mentioned previously, a geodesic curve $(\mu(t), \Sigma(t))$ on $\mathcal{N}(n)$ is described by the following geodesic equation:

$$\begin{cases} \ddot{\Sigma} + \dot{\mu}\dot{\mu}^T - \dot{\Sigma}\Sigma^{-1}\dot{\Sigma} = 0 \\ \ddot{\mu} - \dot{\Sigma}\Sigma^{-1}\dot{\mu} = 0 \end{cases} \tag{6.6}$$

In order to find non-Euclidean analogies on $\mathcal{N}(n)$ by the application of the parallelogrammoid algorithm, a fundamental issue has to be overcome. As explained in the reminder on Riemannian geometry, there exists two equivalent definitions of geodesic curves:

1. A geodesic can be interpreted as a curve of shortest length between two points. It is described by two points A and B.

2. A geodesic can be interpreted as an auto-parallel curve, hence a curve generated by the parallel transport of its celerity. It is described by the initial state: the initial position $A \in \mathcal{M}$ and the initial celerity $\xi \in T_A\mathcal{M}$.

These two definitions are equivalent but switching from the one to the other is a complex task in general. The second definition offers a simple computational model for geodesic shooting, since it corresponds to integrating a differential equation (equation 6.6 in our case), but using it to find a geodesic between two points requires to find initial celerity ξ.

In the scope of this chapter, we consider the algorithm for minimal geodesic on $\mathcal{N}(n)$ proposed by (Han and Park, 2014). The proposed algorithm is based on the simple idea to shoot a geodesic using initial celerity ξ using equation 6.6 and to update ξ based on the Euclidean difference between the endpoint of the integrated curve and the actual expected endpoint. The algorithm is empirically shown to converge for lower dimensions ($n = 2$ or $n = 3$).

6.3.1.2 Experimental Results

We present the results of the parallelogrammoid procedures $D_1 = \mathcal{A}_p(A, B, C)$ and $D_2 = \mathcal{A}_p(A, C, B)$ obtained for various bidimensional multinormal distributions.

We use the classical representation of the multivariate normal distributions by the isocontour of its covariance matrix, centered at the mean of the distribution. The results we display are presented as follows:

- **In black**: Intermediate points in the trajectories γ_1, γ_2 and γ_3.
- **In blue**: Normal distribution A.
- **In green**: Normal distribution B.
- **In cyan**: Normal distribution C.
- **In red**: Normal distribution $D_1 = \mathcal{A}_p(A, B, C)$.
- **In magenta**: Normal distribution $D_2 = \mathcal{A}_p(A, C, B)$.

Case 1: Fixed covariance matrix

For the first case, we fix $\mu_A = (0,0)$, $\mu_B = (1,1)$, $\mu_C = (0,1)$ and $\Sigma_A = \Sigma_B = \Sigma_C = \begin{pmatrix} 1 & 0 \\ 0 & .1 \end{pmatrix}$.

The space of normal distributions with fixed covariance matrix is Euclidean, which implies that algorithm $\mathcal{A}_p$ is equivalent to the parallelogram rule under these conditions and that the defined relation is an analogical proportion. We observe on Figure 6.7 that the trajectories of means in the space correspond to a parallelogram and that the two solutions are identical. This observation can be verified from the differential system 6.6.

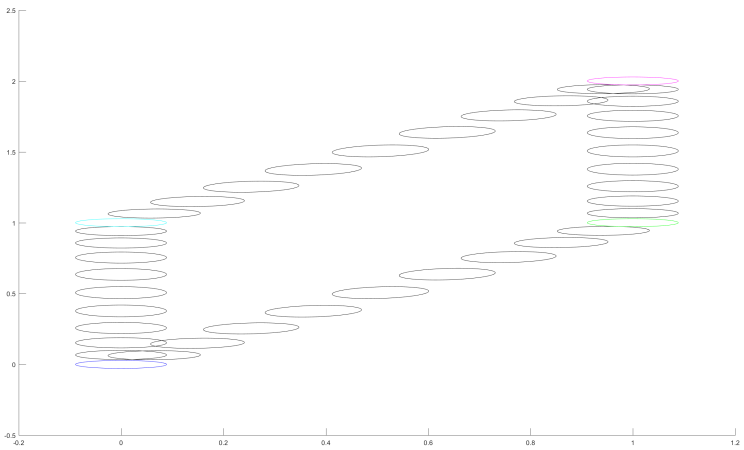


FIGURE 6.7: Results for case 1 (fixed covariance matrix setting).

Case 2: Fixed mean in source domain, fixed covariance from source to target

For the second case, we fix $\mu_A = \mu_B = (0,0)$, $\mu_C = (0,2)$ and, for covariance matrices, $\Sigma_A = \Sigma_C = \begin{pmatrix} 1 & 0 \\ 0 & .1 \end{pmatrix}$ and $\Sigma_B = \begin{pmatrix} .1 & 0 \\ 0 & 1 \end{pmatrix}$.

With these parameters, we observe that the two results are different (Figure 6.8). The result of $\mathcal{A}_p(A, B, C)$ corresponds to the intuition that D will have the same mean as C and the same covariance change as B compared to A. However, for the case $\mathcal{A}_p(A, C, B)$, the results are non-intuitive: the mean of distribution D is different from the mean of C. It can be explained by the fact that the trajectory varies both in μ and Σ. The geometric properties of information require that these two dimensions are related together and that the change in μ depends on the change in Σ.

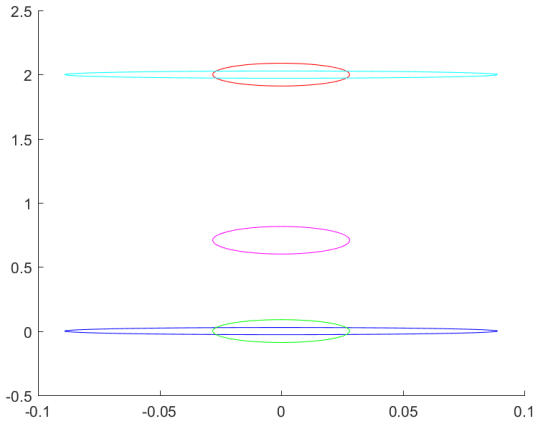


FIGURE 6.8: Results for case 2 (fixed mean in source, fixed covariance from source to target).

Case 3: Symmetric distributions

For the third case, we fix $\mu_A = (0,0), \mu_B = (1,0)$ and $\mu_C = (0,1)$, and, for covariance matrices, $\Sigma_B = \Sigma_C = \begin{pmatrix} 1 & -.5 \\ -.5 & .5 \end{pmatrix}$ and $\Sigma_A = \begin{pmatrix} 1 & .5 \\ .5 & .5 \end{pmatrix}$. We notice on Figure 6.9 that the trajectory leads to a distributions with "flat" covariance matrix (with one large and one very small eigenvalue). No real intuitive interpretation can be given of the observed trajectory (which shows that information geometry cannot explain shape deformations, here ellipse deformations, as expected by human beings).

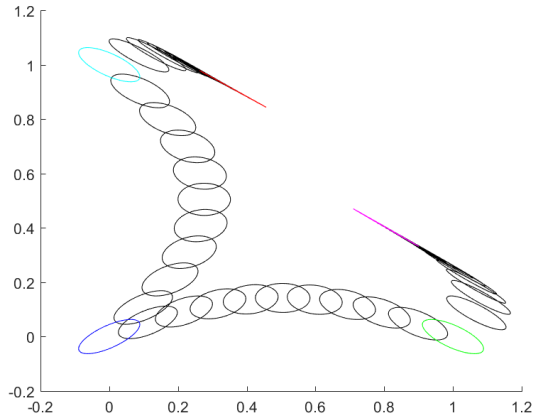


FIGURE 6.9: Results for case 3 (symmetric).

Case 4: Slight perturbation

For the third case, we fix $\mu_A = (0,0), \mu_B = (1,0)$ and $\mu_C = (0,1)$, and, for covariance matrices, $\Sigma_A = \begin{pmatrix} 1 & .5 \\ .5 & .5 \end{pmatrix}$, $\Sigma_B = \begin{pmatrix} 1 & -.5 \\ -.5 & .5 \end{pmatrix}$ and $\Sigma_C = \begin{pmatrix} 1 & .6 \\ .6 & .6 \end{pmatrix}$. Covariance matrix Σ_C is slightly different from Σ_1. If they were equal, the parallelogramoid would be closed. However, the slight modification introduces a perturbation large enough to make $\mathcal{A}_p(A,B,C) \neq \mathcal{A}_p(A,C,B)$ (Figure 6.10). Such artifacts could introduce larger errors in case the distributions are not know with good precision (for instance if they were estimated from data).

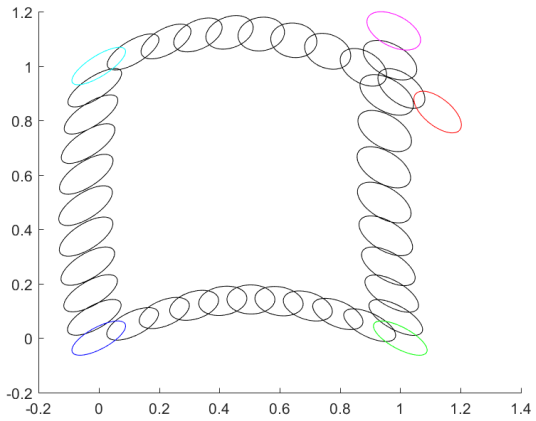


FIGURE 6.10: Results for case 4 (slight perturbation)

6.3.2 Non-Euclidean Analogies in Curved Concept Spaces

The theory of concept spaces (Gärdenfors, 2004) generally involves concept spaces with a Euclidean structure. The vector nature of concepts is the most elementary representation of information where all description features are grouped together in a concatenation. However, there is no formal justification for this simple property. In particular, in the scope of this chapter, it is interesting to wonder if concept spaces could be endowed with a Riemannian structure instead.

A concept space is defined as a geometric structure characterized by a list of *qualities*. When these qualities are independent (ie. it is impossible to express one of them as a function of the others), they form the dimensions of a vector space. Each element in the concept space is then defined by a vector in this vector space. It is not the case when the qualities are independent. For instance, considering a physical space of disks, defined with two qualities (radius and surface), the dependency of the two qualities is straightforward. The concept space is not a vector space in this case.

How to define the metric g of such a space given the dependencies between qualities? This problem is related to the question of *induced metrics* and has explicit solutions in differential geometries, that will not be detailed here.

An interesting perspective for an application would be to consider such a conceptual space and study the influence of the dependency of qualities onto the analogies made in this space.

6.4 Conclusion

In this chapter, we applied the minimum description length principle for analogies in geometric spaces and suggested an operation based on two operations ω_1 and ω_2, which correspond respectively to an intro-domain transformation and to an inter-domain transformation. In vector spaces, the most elementary transformations that can play this role are translations, but in more complex spaces, this is not possible.

We have defined such transformations in the context of Riemannian manifolds and explained that these transformations are an intuitive counterpart to the well-known parallelogram procedure in vector spaces. However, this modified algorithm does not verify the central symmetry axiom of analogical proportion. We presented a simple construction of analogical proportions on manifolds but did not manage to

find a construction for continuous analogical proportions. Lastly, we proposed application cases for our parallelogramoid procedure in order to show that this problem is not merely theoretical but impacts the resolution of analogical equations in very natural spaces.

This chapter concludes the first part of this thesis dedicated to analogical reasoning. Until now, we presented a justification for considering minimum complexity analogies as a *good* approach to analogical reasoning, as well as some issues that might emerge in geometrical settings. Next part will be dedicated to a closely related problem: Transfer learning. As we will show, transfer can be seen as an analogy on the space of data. At this point of our investigation, it seems pertinent to apply the principles we promote to our own research, thus to make an analogy between *analogical reasoning* and *transfer learning*.

Part II

From Analogy to Transfer Learning

Chapter 7

Transfer Learning: An Introduction

The problem of analogical reasoning is a good example of *one-shot learning*: Based on one single example, the agent is able to transfer the knowledge to another problem. This idea is particularly interesting to model human cognition, but becomes irrelevant when example generation is noisy (which is the case in most real world *data mining* problems). In such cases, one single example is not enough and the agent may take advantage of extracting knowledge from group behaviour.

In the context of statistical learning, these group behaviours are described by statistical properties, where groups correspond basically to probability distributions. A single point can be an outlier, but the average information over all points is representative about the actual distribution. This intuition is at play in learning theory, in order to show that the performance on the training dataset is related to the performance on a test dataset of same distribution. However, in real life problems, it is not always possible to assume that data used for training and data on which the learner works have the same distribution. When distributions change after learning, the problem is called *transfer learning*.

The purpose of this chapter is to introduce the general problems of transfer learning as well as an overview of state of the art techniques to deal with changes in distributions. The remainder of this chapter is organized as follows. In Section 7.1, we introduce the general problems related to transfer, in particular multitask learning and the so-called *learning to learning* question. In Section 7.2, we propose an overview of methods used for transfer learning. Finally, in Section 7.3, we address the question of when to transfer: This question is of major importance since it concerns the very possibility of knowledge transfer.

7.1 What is Transfer?

In this section, we propose general notions and notations relative to transfer learning. The section is organized as follows. We first provide some application examples in order to motivate the study of transfer. We then propose a formalization of the notations and expose the two historical directions followed by the research on transfer. Finally, we will present a taxonomy of the various problems associated to transfer learning.

7.1.1 Examples of Transfer Learning Problems

Transfer learning involves any domain involving changing data distributions on a same space, or transfer from one space to another space. Various applications have been found that satisfy these two constraints.

7.1.1.1 Transfer Learning for Computer Vision

The task of *computer vision* naturally involves transfer learning for several reasons as exposed by (Shao, Zhu, and Li, 2015) or (Fei-Fei, 2006). First, from a more conceptual point of view, this ability to transfer prior knowledge of other unrelated domains onto the current task is inspired by human cognitive abilities. (Biederman, 1987) estimated that about 10 to 30 thousand classes of objects can be distinguished in real world, which means that children can acquire a knowledge of about five new classes of objects in a day. This suggests that children are able to learn new categories from very few (even one) instances. Transfer learning offers a good paradigm to mimic this behavior.

The second reason is inherent to the nature of the computer vision task in natural environments and concerns cross-domain and cross-view transfer. From one view to another, various physical parameters might change: image quality, context of the picture, style, orientation of the object etc.

The problem address by (Cao, Liu, and Huang, 2010) is typical of these issues: Their approach consists in transferring knowledge acquired on one dataset, the KTH dataset (Schuldt, Laptev, and Caputo, 2004), and to apply it to another dataset, the Microsoft research action dataset II and TRECVID surveillance data (Smeaton, Kraaij, and Over, 2004). This approach was pioneering in the domain of action recognition. Cross-dataset image categorization has also been used as a solution to the lack of training data for person reidentification (Peng et al., 2016). The problem of person reidentification consists in identifying a same person on the video recordings of different cameras. Traditionally, a fully supervised procedure is used, where labeled data are available for couples of cameras. These methods are obviously extremely demanding in terms of data, and the demand on data cannot be satisfied in real-life situations where hundreds of cameras have to be considered at the same time.

The question of viewpoint changes in images is the core of cross-view transfer. For instance, (Weinland, Ronfard, and Boyer, 2006) proposes the IXMAS dataset for human action recognition recorded from various viewpoints.

7.1.1.2 The Problem of “Small Data"

Apart from the computer vision application, transfer learning is particularly popular in domains where very few training data can be collected. When large amounts of data are available, the training of traditional machine learning algorithms is sufficient to converge to high performance rates. However, getting data is a costly process and, in many situations, it is simply not possible to get a large enough dataset.

In these situations, a solution to overcome the lack of data is to use already known models and adapt them to the considered domain, which is exactly the task of transfer learning. A classifier is learned from a source domain where many labeled data are available, and then transferred to the target domain with very few data.

As an example among many, consider the question of the self-driving car. Acquiring data for this domain is extremely costly, since it requires actual driving of the car in real and various environments, including the rarest. A solution to this problem has been proposed by several methods, including the recent CyCADA (Hoffman et al., 2018), to exploit images from video games and to transfer the information to real life images.

7.1.2 Background and Notations

In this thesis, we will use the now classical notations of (Pan and Yang, 2010).

A **domain** $\mathcal{D} = \{\mathcal{X}, P\}$ is given by a feature space $\mathcal{X}$ on which data are defined and by a probability distribution P over $\mathcal{X}$. The domain designates the space where data are defined as well as the way data are distributed in this space. A **task** $\mathcal{T} = \{\mathcal{Y}, f(.)\}$ defines a label set $\mathcal{Y}$ and a labeling function $f : \mathcal{X} \to \mathcal{Y}$.

The traditional i.i.d. case corresponds to a fixed domain $\mathcal{D}$ and task $\mathcal{T}$, which remain the same at training and testing time. On the contrary, transfer learning is involved when either the domain or the task vary between training and testing.

As a consequence, the notions of training and testing are very different from traditional machine learning, and these terms are usually not used in transfer learning. The training task and domain will be called the *source* and the test will be called the *target*. The idea of transfer learning is to "transfer" the knowledge acquired on the source task in order to solve the target task. In the following, we will use the notation S to designate the source and T to designate the target. In particular, we will consider two datasets: The source dataset, denoted by $D_S = \{(X_i^S, y_i^S)\}_{i=1...n_S}$, and the target dataset $D_T = \{(X_i^T, y_i^T)\}_{i=1...n_T}$. A source hypothesis $h_S : \mathcal{X}_S \to \mathcal{Y}_S$ is learned from data D_S and adapted to a target hypothesis $h_T : \mathcal{X}_T \to \mathcal{Y}_T$ which describes target data correctly. This function h_T can be obtained using inputs X_i^T only or inputs and labels (X_i^S, Y_i^S).

7.1.3 Historical References and Related Problems

This general description of the problem is rather general and can be used for several related problems. These problems, apart from being conceptually close to transfer learning, are also part of the historical development of tranfer learning.

The first researches in transfer learning were held under the name "learning to learn" and were motivated by performance issues. Two main ideas guided this research. First, coordinate computing efforts is supposed to gain time by reducing the number of redundant computations (Pratt, Mostow, Kamm, and Kamm, 1991). Secondly, this coordination had to be integrated into a more ambitious conception of learning: Instead of learning to solve a given task, systems had to acquire the ability to learn any other task, hence the denomination "learning to learn". Encouraged by several seminal works (Thrun and Pratt, 1998), a first workshop was organized at NIPS conference in 1995 to encourage the research on connected domains. This workshop regrouped various works that focus on the idea that previous knowledge must be stored in order to be reused in future situations.

At the same time, another similar paradigm emerged: multitask learning (Caruana, 1997). The principle of multitask diverges slightly from transfer since it is fully symmetrical in terms of the task solving. In this framework, the system faces several different tasks at the same time and proceeds to common computations in order to solve them. One of the main inspirations for such techniques is neural network architecture, in which a first layer could contain commmon information about the various tasks.

The symmetric role of tasks in multitask learning was dismissed in 2005 by the new definition of Transfer Learning established by the Broad Agency Announcement (BAA) 05-29 of Defense Advanced Research Projects Agency (DARPA)'s Information Processing Technology Office (IPTO): Transfer learning is defined as "the ability of a system to recognize and apply knowledge and skills learned in previous

tasks to novel tasks (in new domains)"[1]. This new definition considered the separation of the tasks into source tasks and a target task which is the most important in the learning process. After some knowledge is extracted from the source tasks, this knowledge is chosen and modified by the transfer algorithm in order to apply to the target task. This approach is now the canonical definition of transfer learning.

Among related problems that have been studied but will not be considered in the scope of this thesis, Self-Taught Learning (Raina, Battle, Lee, Packer, and Ng, 2007). In self-taught learning, no labeled data are available in the source domain. The source unlabeled data are used for a classification task in the target domain. It is noticeable that data distributions in source and target domains do not have to be identical. The key point of self-taught learning is that the source data provide knowledge on data representation, which can be shared with target data. For instance, in the domain of image classification, the representation and structure of images does not depend on the object represented in the pictures, and thus a dictionary trained on images from different domains can bring representative information that is used successfully in the target classification task (Wang, Nie, and Huang, 2013).

7.1.4 A Taxonomy of Transfer Learning Settings

In the context we proposed to investigate, several settings can be observed, depending on the existence of labels in source or target datasets. We will omit, on purpose, the case of partially observed labels in the target, which corresponds to semi-supervised domain adaptation. The different tasks are summed up in Figure 7.1.

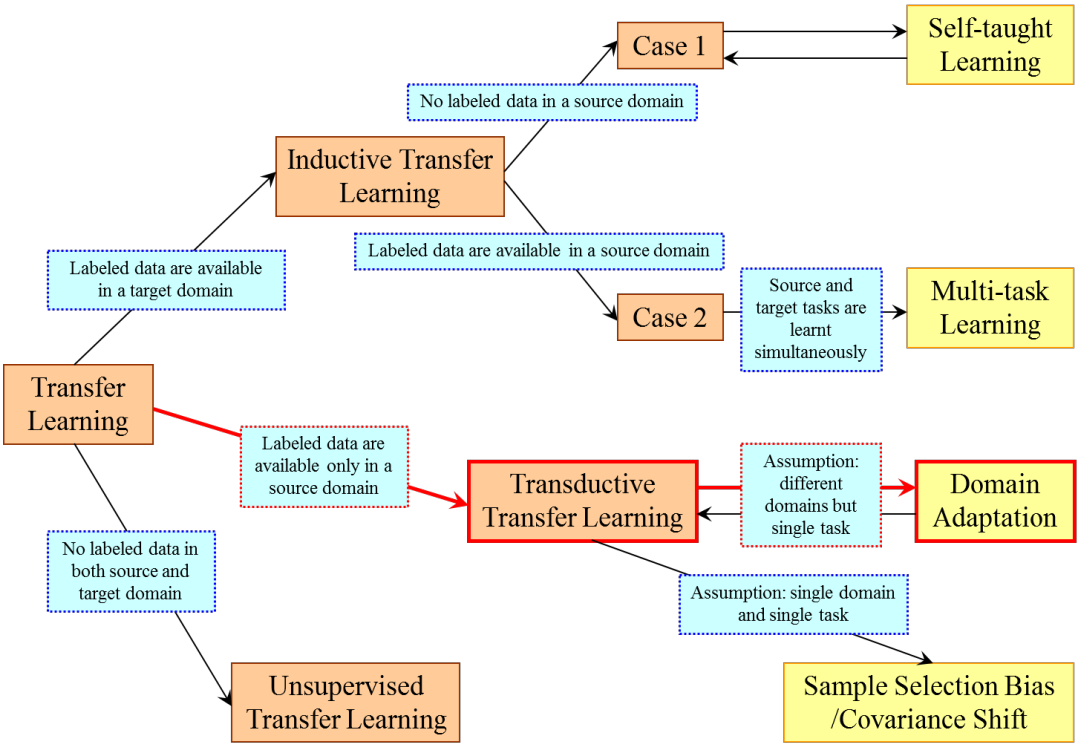


FIGURE 7.1: A Taxonomy of Transfer Learning Settings (Figure from (Pan and Yang, 2010)).

[1]Definition given in the program brief, not available online anymore but archived at http://web.archive.org/web/20110114122026/http://www.darpa.mil:80/i2o/programs/tl/tl.asp

Inductive Transfer problems are characterized by the presence of labels in the target domain, independently of the presence or absence of labels in source data. The term “inductive" refers to the idea that such methods aim to generalize the inferred concept to any other data point outside the observed dataset. As such, the necessity of transfer might not be clear, since the presence of labels in the target makes it possible to apply classical learning methods. The necessity of transfer is often justified by a very limited number of observations in the target, which makes the estimation from the target only very inaccurate. As explained earlier for Self-Taught Learning (Raina, Battle, Lee, Packer, and Ng, 2007) and Multi-Task Learning (Caruana, 1997), the representation knowledge acquired by the first task can be beneficial in the target. Apart from these two learning problems, Hypothesis Transfer Learning (Kuzborskij and Orabona, 2013) is another example of inductive transfer: A hypothesis is estimated in the source domain and has to be transferred to fit the target data.

Transductive Transfer is characterized by the absence of labels in the target domain and the presence of labels in the source domain. Unlike for inductive transfer problems, it is natural in this context that transfer is needed, since it compensates the absence of classification information. In the scope of this thesis, we will mainly focus on Transductive Transfer, even if the proposed model is also relevant for Inductive Transfer.

Unsupervised Transfer, which will not be considered in this thesis, corresponds to the case where no labels are available, neither in the source nor in the target domains. This application remains rare in the state of the art.

7.2 Trends in Transfer Learning

In this section, we propose a brief presentation of the main trends observed in transfer learning. This overview is laconic and only gives general directions. For more details, we refer the readers to surveys (Pan and Yang, 2010; Weiss, Khoshgoftaar, and Wang, 2016).

7.2.1 Importance Sampling and Reweighting

A very simple method that is proposed for transfer is based on the idea of instance reweighting for empirical risk minimization. In this framework, the source data are reweighted according to a weight $\omega(x) = P_T(x)/P_S(x)$. In practice, this weight cannot be computed and is then estimated using various methods that will be described shortly.

Instance reweighting takes its root in the idea of risk minimization, and especially the estimation of empirical risk minimization. Given a loss function $l : \mathcal{Y}^T \times \mathcal{Y}^T \to \mathbb{R}$, the purpose of risk minimization is to find a hypothesis $h : \mathcal{X}^T \to \mathcal{Y}^T$ that minimizes the risk

$$R(h) = \mathbb{E}_{(X,Y)\sim P_T}\left[l(Y, h(X))\right]$$

Since the probability distribution P_T is unknown, this value cannot be assessed and an estimated value is used, called *empirical risk*: $R_n(h) = \sum_{i=1}^{n} P_T(X_i, Y_i) l(Y_i, h(X_i))$ where the (X_i, Y_i) are drawn from distribution P_T. The idea of instance weighting is

based on the following computation:

$$\begin{aligned} R(h) &= \mathbb{E}_{(X,Y)\sim P_T}\left[l(Y,h(X))\right] \\ &= \int_{(x,y)\in\mathcal{X}\times\mathcal{Y}} P_T(x,y)l(y,h(x)) \\ &= \int_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \frac{P_T(x,y)}{P_S(x,y)} P_S(x,y)l(y,h(x)) \\ &= \mathbb{E}_{(X,Y)\sim P_S}\left[\frac{P_T(x,y)}{P_S(x,y)} l(y,h(x))\right] \\ &\simeq \frac{1}{n_S}\sum_{(x,y)\in D_S} \frac{P_T(x,y)}{P_S(x,y)} l(y,h(x)) \end{aligned}$$

Under the hypothesis that $P_S(Y|X) = P_T(Y|X)$, called *covariate shift* (Shimodaira, 2000), the weight becomes simply $\omega(x) = P_T(x)/P_S(x)$.

Several methods are known to estimate the weight $\omega(x)$ based on source dataset D_S and target dataset D_T. A very simple approach consists in estimating the distributions P_S and P_T from a class of distributions. In order to avoid estimating these distributions, (Sugiyama, Nakajima, Kashima, Buenau, and Kawanabe, 2008) proposes to estimate the quantity $\omega(x)$ directly by a linear approximation over basis functions. The choice of the linear coefficient is done based on the minimization of Kullback-Leibler divergence between the corresponding inferred distribution and real distribution. Similarly, (Bickel, Brückner, and Scheffer, 2007) avoids estimating the distribution by relying on a discriminative criterion, where a variable σ measures if the data point is drawn from P_S or P_T. A simple expression of ω is derived from this assumption. A third solution, proposed by (Huang, Gretton, Borgwardt, Schölkopf, and Smola, 2007) and called *Kernel Mean Matching*, aims to find an optimal weight such that the means of the projections of source and target data onto a same Reproducible Kernel Hilbert Space (RKHS) are close. It is shown, in the case where the RKHS satisfies desired properties, the weight provides a good estimation of the actual weight.

The main limitation of these methods is the strong hypothesis of covariate shift. The techniques that will be described below are more general. However, the framework of covariate shift is simple and adapted to many real life problems. For a more general overview of these topics, we refer the reader to (Sugiyama, Lawrence, and Schwaighofer, 2017).

7.2.2 Optimal Transport

Optimal Transport is based on the simple assumption that the target domain $\mathcal{X}_T$ is the image of the source domain $\mathcal{X}_S$ with respect to a function $T : \mathcal{X}_S \to \mathcal{X}_T$ called transport map. In order to apply Optimal Transport to transfer learning, another assumption is imposed on the function T: It is supposed to preserve the conditional distributions: $P_S(Y|X_S) = P_T(Y|T(X_S))$.

In terms of probability distributions, the operation defined by T can be applied to a density μ_S over the source space $\mathcal{X}_S$, which is written $T\#\mu_S$, and corresponds to the distribution of $x \in \mathcal{X}_T$ which are the image of elements of $\mathcal{X}_S$ after transformation by T.

Since researching the transport map T in the space of all transformations is intractable, a restricted search is proposed, that is based on a cost function $c : \mathcal{X}_S \times$

$\mathcal{X}_T \to \mathbb{R}^+$. Monge's formulation of optimal transport is then the following:

$$\begin{aligned} \text{minimize}_T \quad & \int_{\mathcal{X}_S} c(x, T(x)) d\mu(x) \\ \text{s.t.} \quad & T\#\mu_S = \mu_T \end{aligned}$$

In practice, this formulation is rarely used, replaced by its convex relaxation, given by (Kantorovitch, 1958). In this formulation, the purpose is to infer a probability distribution γ over $\mathcal{X}_S \times \mathcal{X}_T$, the marginals of which correspond to distributions μ_S and μ_T. Kantorovitch formulation is the following:

$$\begin{aligned} \text{minimize}_\gamma \quad & \int_{\mathcal{X}_S \times \mathcal{X}_T} c(x, y) d\gamma(x, y) \\ \text{s.t.} \quad & \int_{\mathcal{X}_T} \gamma(x, y) dy = \mu_S(x) \\ & \int_{\mathcal{X}_S} \gamma(x, y) dx = \mu_T(y) \end{aligned}$$

Optimal transport applies in a natural way to transfer learning (Courty, Flamary, Tuia, and Rakotomamonjy, 2017): It can be shown that the Kantorovitch problem has a very simple formulation when the distributions are discrete (which is the case in the problem of transfer, since the distributions are estimated simply by a sum of Dirac functions on the data points). A problem with this technique is that it can lead to overfitting. As a soluion, (Cuturi, 2013) suggests to use a penalization based on entropy, which speeds up the computations and improves the accuracy of the transport for some problems. Another family of regularizations, called class-based regularizations, aims to preserve the label information during the transport. We will not discuss these regularizations nor their applications here. Another different technique proposed by (Courty, Flamary, Habrard, and Rakotomamonjy, 2017) suggests to apply optimal transport on joint distributions over $\mathcal{X} \times \mathcal{Y}$ rather than on the domain space $\mathcal{X}$ only.

The real strength of transfer methods based on optimal transport is the explicit mapping that is a natural output, as well as the empirical validation that show excellent results on transfer learning problems.

7.2.3 Mapping and Learning Representations

An intuition that is the core of several transfer learning techniques is to assume that the direct representation given by the source and target domains $\mathcal{X}_S$ and $\mathcal{X}_T$ is not the most appropriate representation. Based on this idea, the problem of transfer is seen seen as the learning of an optimal representation. The main characteristics of a good representation space is that it has to represent shared characteristics between two domains.

A typical example is given by (III, Hal Daume, 2007) in the domain of Natural Language Processing. The authors consider the case where $\mathcal{X}_S = \mathcal{X}_T$ and propose to define a new input space $\tilde{\mathcal{X}}$ of higher dimensionality than the original space. This new input space is obtained by concatenation of 3 vectors of the original space (hence, $dim(\tilde{\mathcal{X}}) = 3 \times dim(\mathcal{X})$). The transformation from the original input space to the extended input space is given by a transformation $\Phi : \mathcal{X} \mapsto \tilde{\mathcal{X}}$ defined as

follows: $\Phi(x) = \langle x, x, 0 \rangle$ for x in source, and $\Phi(x) = \langle x, 0, x \rangle$ for x in target. This extremely simple transformation is shown to be sufficient to address the problem of domain adaptation.

Another well-known example in the domain of NLP is Structural Correspondence Learning (Blitzer, McDonald, and Pereira, 2006), a technique which uses unlabeled instances of source and target domain to build a common feature representation. This representation is based on *pivot features*, features that are frequent and diverse enough in both source and target domains. Such features can be associated to projections onto $\mathbb{R}$ (through a linear classifier determining if a point x has the pivot feature present or absent), the original features can be then represented as a common low-dimensional feature space.

The deep learning methodology being particularly well-adapted to the question of feature representation, it has been widely used for representation mapping in transfer learning. (Glorot, Bordes, and Bengio, 2011) propose to use stacked denoising auto-encoders to extract characteristic features from unlabeled source and target data. A classifier is then learned on this common representation space, from the projected labeled data only. This property of transferability of representation, which has been studied from a general point of view by (Yosinski, Clune, Bengio, and Lipson, 2014), has been used for the purpose of transfer learning by various authors, following the paradigm of feature representation and mapping (Long, Cao, Wang, and Jordan, 2015; Luo, Zou, Hoffman, and Fei-Fei, 2017). Among these methods, the algorithm proposed by (Ganin and Lempitsky, 2015) is a bit different, in the sense that it uses back-propagation directly to incorporate the multi-domain transfer inside a feed-forward architecture. The proposed network aims to determine, for an element $x \in \mathcal{X}$, both its label $y \in \mathcal{Y}$ and its origin (source or target distribution).

Recent improvements in the domain of computer vision rely on the use of Generative Adversarial Networks (GANs). These techniques aim to project the target images into source style images, hence images that share similar distribution with source images. Among such methods, CyCADA (Hoffman et al., 2018) proposes to use a GAN to transpose the target image into the style of the source, respecting a principle of cycle consistency. This idea was developed in parallel by (Murez, Kolouri, Kriegman, Ramamoorthi, and Kim, 2018).

7.3 A Central Question: When to Transfer?

In the previous section, we presented existing techniques used to solve transfer learning problems. These problems occur in practice in contexts where few or no labels are available in target domain. However, there is *a priori* no guarantee that the transfer that will be done will actually lead to correct results. In this section, we expose some ideas about performance of transfer. First, we will introduce the notion of negative transfer, that is inherent to transfer learning. We will then present the major theoretical framework used to evaluate the quality of transfer. Finally, we will present an evaluation of the notion of task relatedness.

7.3.1 Introducing Negative Transfer

Negative transfer is a phenomenon that is inherent to transfer learning but goes against the intuition that more learning material (for instance more data) necessarily means better performance. The mathematical justification of this intuition is provided by the law of large number: When the number of observed points tends to

infinity, the empirical distribution converges to the actual distribution. Obviously, this result does not hold when the data distribution varies.

In transfer learning, the performance depends, in a large extent, on the relatedness between source and target tasks. When no correlation can be found between these two tasks, it is expected that transfer fails and gives poorer results than learning from scratch.

This result has been observed empirically by (Rosenstein, Marx, Kaelbling, and Dietterich, 2005) in the context of a simple transfer task between two populations, the relatedness of which can be controlled. The transfer algorithm used by the authors is based on Naive Bayes, and does not take task relatedness into account. It is then shown that the performance of the algorithm highly depends on the relatedness of source and target. Two major effects can be observed. If the source and target tasks are correlated, an improvement is observed for transfer over learning from target data only. This improvement can be explained by the fact that information is brought by the (similar) source data. On the contrary, when the two tasks are not correlated, the performance is penalized by the use of source information: In this case, better performances are achieved by training the learner on the target data only. A restriction can be noticed: The paper focuses on a small number of observations in target. When the number of target data increases, the amplitude of the difference between negative transfer and no transfer decreases, whereas “positive" transfer still gives significantly better results.

These observations show the necessity to consider the problem of negative transfer. In the following, we will present two ideas that are relative to this notion. The first idea is an adaptation of learning theory to the case of transfer learning. The second idea, which follows directly, consists in measuring the relatedness between two tasks.

7.3.2 Guarantees with Small Drifts

A first theoretical analysis of domain adaptation was provided by the seminal works of (Ben-David, Blitzer, Crammer, Kulesza, Pereira, and Vaughan, 2010) in the context of small differences between source and target tasks. The main purpose of the paper is to bound the target error of a hypothesis h:

$$\epsilon_T(h, f) = \mathbb{E}_{x \sim \mathcal{D}_T} \left[|h(x) - f(x)| \right]$$

where $\mathcal{D}_T$ designates the target distribution and f is the target labeling function. This error measures the expected difference between the prediction of hypothesis h and actual labeling function f.

The article opens with an observation: L^1 divergence between distributions

$$d_1(\mathcal{D}, \mathcal{D}') = 2 \sup_{B \in \mathcal{B}} |Pr_{\mathcal{D}}[B] - Pr_{\mathcal{D}'}[B]|$$

where $\mathcal{B}$ is the set of measurable subsets under $\mathcal{D}$ and $\mathcal{D}'$, can be used to estimate a first upper-bound of the target error, but this bound has two major drawbacks: It cannot be accurately estimated and it provides a too large bound (since it considers a supremum over all possible sets). As a solution, the authors suggest using another measure, called $\mathcal{H}$-divergence, defined as follows:

$$d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}') = 2 \sup_{h \in \mathcal{H}} |Pr_{\mathcal{D}}[I(h)] - Pr_{\mathcal{D}'}[I(h)]|$$

where $I(h) = \{x \in \mathcal{X} | h(x) = 1\}$. This quantity can be estimated from finite samples and the corresponding estimator will be denoted by $\hat{d}_{\mathcal{H}}(\mathcal{D}, \mathcal{D}')$. The notion of $\mathcal{H}$-divergence can be applied in particular to the symmetric difference hypothesis space $\mathcal{H}\Delta\mathcal{H}$, defined as the set of hypotheses g such that there exists $h, h' \in \mathcal{H}$ such that $g(x) = h(x) \oplus h'(x)$, where $\oplus$ is the XOR operator. It is involved in a fundamental theorem:

Theorem 7. *(Ben-David, Blitzer, Crammer, Kulesza, Pereira, and Vaughan, 2010) Let $\mathcal{H}$ be a hypothesis space of VC dimension d. If $\mathcal{U}_s, \mathcal{U}_T$ are unlabeled samples of size m' each, drawn from $\mathcal{D}_S$ and $\mathcal{D}_T$ respectively, then for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the choice of the samples, for every $h \in \mathcal{H}$:*

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_s, \mathcal{U}_T) + 4\sqrt{\frac{2d\log(2m') + \log\left(\frac{2}{\delta}\right)}{m'}} + \lambda \tag{7.1}$$

where $\lambda = \min_{h \in \mathcal{H}} \epsilon_S(h) + \epsilon_T(h)$

The key hypothesis in this framework is the idea that the source and target tasks are close. This hypothesis is reflected at several levels in Equation 7.1:

- The same hypothesis h is involved in the source and target errors. If the two tasks are uncorrelated, it is expected that h is not proficient in both domains. In particular, an optimal h for the target can lead to large error in the source, which makes the theorem non-informative.
- The term λ can be high if there is no classifier in $\mathcal{H}$ that performs well for the combined error $\epsilon_S(h) + \epsilon_T(h)$, which can be the case when the tasks are too different.
- The divergence term $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_s, \mathcal{U}_T)$ is expected to be high if the two domains do not match with regards to the hypothesis class $\mathcal{H}$.

These ideas reflect in particular an interesting aspect of this theory: The hypothesis space $\mathcal{H}$ is involved here in a trade-off between the accuracy of some hypothesis $h \in \mathcal{H}$ for classification in the source, the VC dimension of $\mathcal{H}$ and its incapacity to discriminate the source from the target data.

In practice, this strong assumption is not necessarily a limit. For instance, the mapping-based methods, as well as optimal transport, are motivated by the idea of "projecting" points from the target domain onto the source model in such a way that the distributions are close. The error bound proposed by (Ben-David, Blitzer, Crammer, Kulesza, Pereira, and Vaughan, 2010) applies perfectly well to such methods.

7.3.3 Characterizing Task Relatedness

As seen with the theory of (Ben-David, Blitzer, Crammer, Kulesza, Pereira, and Vaughan, 2010), a major notion in the theory of transfer learning is the notion of task relatedness (called domain divergence in the context of domain adaptation).

We have seen that (Ben-David, Blitzer, Crammer, Kulesza, Pereira, and Vaughan, 2010) proposes two measures for domain divergence: L^1 divergence, discarded for its lack of precision when used in error bounds, and $\mathcal{H}$-divergence, which measures the relatedness of two tasks with respects to a given hypothesis class $\mathcal{H}$.

Other divergence measures have been proposed in the literature. (Mansour, Mohri, and Rostamizadeh, 2009) proposes an extension of the $\mathcal{H}$-divergence to a

more general class of loss functions $\mathcal{L}$. This divergence is given by:

$$disc_{\mathcal{L}}(\mathcal{D}_S, \mathcal{D}_T) = \sup_{(h,h') \in \mathcal{H}^2} |\mathbb{E}_{X_T \sim \mathcal{D}_T}[\mathcal{L}(h(X_T), h'(X_T))] - \mathbb{E}_{X_S \sim \mathcal{D}_S}[\mathcal{L}(h(X_S), h'(X_S))|$$

This quantity is involved in a theorem close to Theorem 7, but which can be more precise in some cases.

Another measure is proposed by (Germain, Habrard, Laviolette, and Morvant, 2013) in a PAC-Bayesian approach. Consider a hypothesis class $\mathcal{H}$ and a posterior distribution ρ on $\mathcal{H}$. The authors define the domain disagreement $dis_\rho(\mathcal{D}_S, \mathcal{D}_T)$ as:

$$dis_\rho(\mathcal{D}_S, \mathcal{D}_T) = \left| \mathbb{E}_{(h,h') \sim \rho^2} \left[\epsilon_{\mathcal{D}_T}(h, h') - \epsilon_{\mathcal{D}_S}(h, h') \right] \right|$$

where $\epsilon_{\mathcal{D}}(h, h') = \mathbb{E}_{X \sim \mathcal{D}} \mathbb{I}(h(X) \neq h'(X))$.

In a different direction, (Zhang, Zhang, and Ye, 2012) considers that the domain divergence should not be the only source of divergence considered to measure task relatedness. The authors propose to combine a measure of the divergence in the distributions and a measure of the divergence in the labeling functions, given by the entropy number of the class of labeling functions.

Finally, we would like to mention the work of (Mahmud, 2009) which is particularly interesting in the context of this thesis, since it characterizes task relatedness with help of Kolmogorov complexity. The author defines the transfer learning distance of two tasks (associated to the semi-measures $\mathcal{D}_S$ and $\mathcal{D}_T$) as $E_1(\mathcal{D}_S, \mathcal{D}_T)$, where $E_1(x, y) = \max\{K(x|y), K(y|x)\}$. (Mahmud and Ray, 2008) exploits this idea in a bayesian setting, following the theory of universal distribution (Solomonoff, 1964; Hutter, 2003) (see end of Section 9.3.2 for more details).

7.4 Conclusion

In this chapter, we presented the general problem of transfer learning, from its applications to the main trends that are observed in its algorithmic treatment. Our review of the existing techniques is far from being complete, but oriented toward the ideas that will be developed in this thesis. We also discussed the major question of the necessity of transfer, not from the practical point of view but from a more theoretical point of view. This question is related to the problem of negative transfer, ie. situations where transfer brings negative information to the target and deteriorates the performance. Several theoretical models have been developed in order to measure task relatedness and prevent bad transfer performances.

In the following chapter, we will follow a simple intuition, knowing that analogy and transductive transfer are very similar tasks. We will present a general framework based on Kolmogorov complexity and discuss simple applications in the context of a prototype-based model.

Chapter 8

Transfer Learning with Minimum Description Length Principle

We have seen in previous chapter that transfer learning is a machine learning task which consists in *transposing* the concept learned on a source domain onto a target domain. Among all the problems presented in our short review, we will focus on transductive transfer, hence a transfer learning problem where the learner is given labeled data in the source domain and unlabeled data in the target domain. Formally speaking, knowing source data $\mathcal{D}_S = \{(X_i^S, Y_i^S)\}_{i=1...N_S}$ and target data $\mathcal{D}_T = \{X_i^T\}_{i=1...N_T}$, the purpose is to estimate a decision function $h_S : \mathcal{X}_S \mapsto \mathcal{Y}_S$ and to transpose it into a decision function $h_T : \mathcal{X}_T \to \mathcal{Y}_T$ on the target domain.

Formally speaking, we notice a large similarity between this description of transductive transfer learning and analogical reasoning as presented in the previous part of this thesis. The purpose of this chapter is to investigate this relationship and to discuss the potential consequences onto the description of transductive transfer.

The remainder of the chapter is organized as follows: In Section 8.1, we propose a discussion of the similarities and dissimilarities between analogical reasoning and inductive transfer. Based on the conclusions of this discussion, we suggest to use the same graphical model as introduced for analogical reasoning. In Section 8.2, we propose two families of models for complexity that can be used in machine learning. These models are elementary and will be used in all the following chapters. Lastly, in Section 8.3, we propose some experimental validation of the proposed methodology.

This chapter develops and extends the ideas and results presented in (Murena and Cornuéjols, 2016).

8.1 Transductive Transfer Learning with Minimum Description Length Principle

In this section, we explore the similarities between transductive transfer and analogical reasoning. Based on the observation of these similarities, we propose to use the principle suggested for analogical reasoning in Equation 5.11 in order to solve transductive transfer problems.

8.1.1 Transductive Transfer and Analogy: Two Related Tasks?

The problem of analogical reasoning, as exposed in Part I, considers problems of the form "A is to B as C is to x" where x is to be found. With this formulation, the task of transductive transfer can be expressed as "X_S is to solution vector Y_S as X_T is to solution vector Y_T", where Y_T is the solution to be found.

The links between analogy and transfer has been already discussed in (Wang and Yang, 2011). The authors suggest to use the principle of Structure Mapping Theory (see Section 3.2.3) and to apply it on transfer learning. The suggested method relies on a projection of both source and target domains onto a common Reproducing Kernel Hilbert Space. The key idea that is exploited by this article is that the similarities between the source and target domains cannot be apparent, but must be observed in the inherent structure of the domains.

Despite some apparent similarities, some points have to be discussed regarding the relation between transductive transfer and analogical reasoning:

- **Cardinality of elements:** Analogy is characterized by a relation involving four elements, while transfer learning might involve many elements in both source and target domains. When the i.i.d. hypothesis holds inside a domain, the elements have the same "structure" (ie. are produced by the same distribution) but are not related together in a structural way.
- **Role of the source:** In analogical reasoning, the source plays a prominent role, since it is involved both in the characterization of the intra-domain transformation and of the inter-domain transfer. This role is not necessarily as strong in transfer learning: for instance, the setting of *hypothesis transfer* makes the source completely unused by assuming that the source hypothesis is already known.

8.1.2 What Analogy Suggests

As shown above, an analogy can be drawn between analogical reasoning and transductive transfer. Since the two problems are similar, it makes sense to consider that their solutions have to be similar too. Following this intuition, we propose to use the DGM introduced for analogical reasoning and to interpret its consequences in terms of transductive transfer.

The problem of transductive transfer can be described as the following analogical equation: $X_S : Y_S :: X_T : y$ where y is the unknown variable. This equation (and its solution) can be described using the DGM presented in Figure 5.5 and its solution can be obtained by minimizing the following objective (already presented in Equation 5.11) over the models M_S and M_T:

$$K(M_S) + K(X_S|M_S) + K(Y_S|M_S, X_S) + K(M_T|M_S) + K(X_T|M_T) \tag{8.1}$$

We postpone the term-by-term analysis of this equation for this specific case of application to Section 8.1.3.

In practice, transfer learning based on this principle involves the following steps:

- **Definition of the models**: A restricted class of models has to be chosen for the source and the target. This choice plays the same role as the choice of a hypotheses class.
- **Estimation of the complexities**: Given a model, an upper-bound for the complexity terms $K(M)$, $K(X|M)$ and $K(Y|X, M)$ is required. Approximations of such terms on simple models will be proposed in Section 8.2.
- **Computing** Y_T: Once the models M_S and M_T have been evaluated, the solution Y_T is inferred. For this purpose, it is needed that the model defines an *intrinsic* transformation $h_M : \mathcal{X} \to \mathcal{Y}$.

8.1.3 Interpretation: A General Principle?

As exposed in Section 5.2, Kolmogorov complexity is not computable. In order to obtain computable values, the admitted solution in the scope of this thesis is to consider an upper-bound of complexity obtained for a fixed class of machines. The choice of the Turing machines defines an inductive bias for the method. It seems important at this point to comment on the chosen bias and the underlying approximations. These hypotheses that are made (which are related to the presented DGM, hence are shared by both analogical reasoning tasks and transductive transfer tasks) will be interpreted in terms of classical approximations in statistical learning and in comparison with state of the art methods in transfer learning.

The first hypothesis at play in our model is the **division of the model**: We suppose that two models are used, one for the source domain, one for the target domain. These two models are not necessarily equal or close. They do not even belong necessarily to the same space. As a complement, we assume a complete **separation of source and target**, which means that source points and labels are described with help of source model only and target points and labels are described with help of target model only.

These two hypotheses together are the major specificity of our framework. They ensure that the transfer is managed at the level of models and not of data.

A closer look at the different terms shows that they can be interpreted in terms of usual machine learning notions.

The complexity $K(X|M)$ measures the descriptive quality of model M for data D. Intuitively, this value corresponds to the notion of *likelihood* that is used in bayesian methods. As we will see later, when the model M is a probability distribution, the term $K(X|M)$ corresponds exactly to the definition of likelihood.

The term $K(Y|X, M)$ measures the complexity to build the labels from the inputs and the model. When the model is perfect, it gives the exact expected solution Y when applied to data X, which corresponds to a zero complexity. In case where one mistake is produced, it has to be corrected, and thus increases the complexity. We will show later that this term is upper-bounded by the empirical risk (up to a multiplicative coefficient).

The terms $K(M_S)$ and $K(M_T|M_S)$ constraint the exploration of the model space by favoring source models of low complexity and target models similar to the source model. These terms are similar to penalization terms that are commonly used in machine learning.

As a conclusion, Equation 8.1 can be seen as an extended version of the common inductive principles used in machine learning. As such, it seems to be very general, while encompassing the characteristics of well-known approaches.

8.2 Defining Models

The proposed framework relies on the use of *models*, which have been interpreted as information factorization in the context of analogical reasoning. In this section, we propose two general classes of models for transfer learning.

8.2.1 Probabilistic models

In machine learning, it is often assumed that data are observations of random variables. Our point is that probability distributions (generating these random variables) can be considered as information factorization.

In order to get an intuition, we can consider first the simple case of a normal distribution on $\mathbb{R}$. Such a distribution is entirely defined given its mean value μ and its variance σ. With such a distribution, the observation $X = \mu$ is the most typical element of the distribution. On the contrary, an observation very far from mean value might be considered as less typical, hence more complex element. Moreover, it is reasonable to imagine that the description of the observation can be estimated in terms of number of standard deviations to the mean value: $(X - \mu)/\sigma$.

This intuition can be generalized based on complexity theory. A link between complexity and probability theory has been established. Let p be a semi-computable distribution over a set $\mathcal{X}$, then the complexity of an element $x \in \mathcal{X}$ is upper-bounded by $K(x) \leq K(p) - \log p(x) + \mathcal{O}(1)$, and in particular:

$$K(x|p) \leq -\log p(x) + \mathcal{O}(1) \tag{8.2}$$

In particular, the intuited formula presented for the normal distribution corresponds to the actual upper-bound value given in Equation 8.2:

$$K(x|p) \leq \frac{1}{2}\log(\sigma^2) + \frac{1}{2}\frac{\|x-\mu\|^2}{\sigma^2} + \mathcal{O}(1) \tag{8.3}$$

The strength of this representation is its link with classical machine learning methods, where models are evaluated by their (log)likelihood regarding observed data, which is exactly the quantity described in Equation 8.2.

The complexity of the models depend on the chosen probability distribution. For parametric distributions, the complexity of the distribution corresponds to the complexity of the parameters, up to an additive function which encodes in particular the choice of the distribution family. For a fixed machine, this family of distribution can even be supposed to be fixed.

8.2.2 A prototype-based model

Another basic way to model data on a space relies on the use of prototype-based models, such as done in Self-Organizing Maps (Kohonen, 1990), Learning Vector Quantization (Kohonen, 1997) or even K-means in unsupervised setting. The principle of such methods is to describe data points in $\mathcal{X}$ as attached to artificial points $P \in \mathcal{X}$ (*a priori* not in the observed dataset) called *prototypes*. The way the coordinates of the prototypes is estimated depends on the training method.

The prototype-based models can be interpreted as compression models: Instead of describing each data point by its absolute position, prototype models describe them by their position relatively to the closest prototype (Figure 8.1). Doing so, the model factorizes the common information shared by multiple data points, and thus is supposed to compress the representation of points. In terms of the language described in Chapter 4, the prototypes could be understood as position vectors stored in memory. The prototypes would be declared inside a `let` instruction and would be called by the `mem` operator to be modified on the fly.

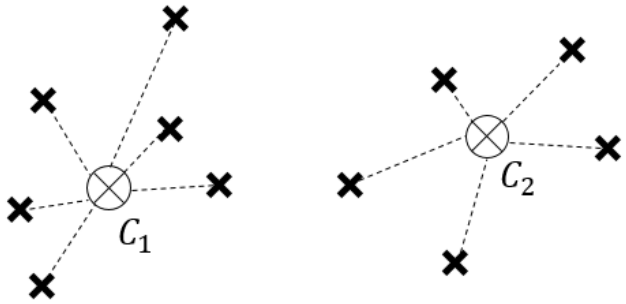


FIGURE 8.1: Representation of a prototype-based model. The circled cross corresponds to a prototype and the standard crosses are data points. Data points are attached to their closest prototype.

We denote by P the matrix of all prototypes, where the i-th line P_i corresponds to the transpose of vector representation for i-th prototype. We will denote by P_i^j the $j-th$ coordinate of P_i (and in general X^j the j-th component of any vector X).

8.2.2.1 Model Complexity

Concerning complexity, we assume that it is possible to consider each data point individually (as in the DGM described in Figure 5.2). This hypothesis, which can be seen as an equivalent of the independence property used in probabilistic settings, will be employed at several other occasions in the following. We formalize it as follows:

Hypothesis 1. *The chosen class of Turing machines considers that the data are algorithmically independent, ie. it requires that the description of points is done independently from each other. As a consequence, if X is the design matrix of a dataset, then the complexity of $X \in \mathbb{R}^{n\times d}$ relative to the chosen class of Turing machines is equal to:*

$$K(X) = \sum_{i=1}^{n} K(X_i) + \mathcal{O}(1) \tag{8.4}$$

The actual description of the prototype model requires two extra pieces of information:

1. The prototypes have to be designated in order to be used in data description. Each of the prototypes is associated with a unique id, which is an integer between 0 and the total number of prototypes. We designate by id_i the id of i-th prototype.
2. In supervised setting, each prototype is associated to a class y_i.

Applying Hypothesis 1, and using these two extra information, it comes that the prototype model complexity can be given by:

$$K(P) = \sum_{i=1}^{p} K(id_i) + K(y_i) + K(P_i) \tag{8.5}$$

8.2.2.2 Data Complexity

Without using the prototype model, data have to be described by setting the absolute positions of the n points in the set and by the labels of each point individually, which corresponds to a complexity equal to $K(X,Y) = \sum_{i=1}^{n} K(X_i) + K(Y_i)$. A compression can be obtained by factorizing common representations into the prototypes.

A point is associated to a prototype, which means that the index of the chosen prototype is explicitly given in the description of the point. Once the prototype is given, the description can be obtained by indicating the relative position of the point to the prototype. This leads to the following complexity for the data:

$$K(X|P) = \sum_{i=1}^{n} \left(K(id_i) + \min_{c \leq p} \left(K(id_c) + K(X_i - P_c) \right) \right) \tag{8.6}$$

Here, the choice of the machine to encode the indexes (id_i and id_c) is crucial. These values being integers, it is tempting to use the basic coding of integers[1] and thus $K(id) = \log id + \log\log id + \mathcal{O}(1)$, but this coding favors prototypes with smaller index. This solution requires then a specific ordering of prototypes, which is not necessarily desired. Another solution, that we will use in the following, consists in taking a non-optimal coding that encodes each index with a fixed number of bits. The minimal complexity of such a coding is then $K(id) = \log n$, where n is the total number of elements in the considered set of ids.

In the considered supervised setting, the class Y of the points has to be specified too. In a binary case, this value can be described by one bit (by $\log |\mathcal{Y}|$ bits for a finite set $\mathcal{Y}$). However, given a prototype, the class of the point can be also available for free, given the assumption that the class of points associated to a prototype is the same as the class of the prototype. Of course, this assumption is not always satisfied, so in practice this prediction needs to be corrected. The conditional complexity of the actual Y relative to the prototype class $Y^{(p)}$ is the defined by:

$$K(Y|Y^{(p)}) = \log |\mathcal{Y}| \times \mathbb{I}(Y \neq Y^{(p)}) \tag{8.7}$$

where $\mathbb{I}$ designates the indicator function (equal to 1 when the argument is true and to 0 otherwise).

8.3 Validation of the Framework: A Prototype-based Algorithm

In this section, we propose to test our framework using a prototype-based model. We propose the subsequent approximations of complexity as well as a description of the used algorithm and the obtained results.

8.3.1 Measuring Complexity

The presented algorithm is based on a prototype-based model and aims to solve problem 8.1. Since this objective function only implies complexity terms, the machine of interest for our use case remains to be defined. This definition is the purpose of the following sections.

[1] With this basic coding, the complexity of $n \in \mathbb{N}$ is upper-bounded by $\log n + \log\log n$

8.3.1.1 Complexity of real numbers

A straightforward encoding of the real numbers is based on a subdivision of the real line in ordered portions of fixed length Δx. A real number $X \in \mathbb{R}$ is described (with a precision Δx) by the index of the portion it belongs to, which corresponds to a discretization of the space. This encoding leads to the following definition of the complexity:

$$K(X) = \log \left\lfloor 1 + \frac{|X|}{\delta} \right\rfloor + \mathcal{O}(1) \tag{8.8}$$

In this definition, the constant includes in particular a bit that gives the sign of the real number.

The parameter δ controls the precision of the encoding: two numbers can be distinguished only if their distance is greater than δ. The limit case where $\delta \longrightarrow 0$ corresponds to maximal precision, hence continuous case. This parameter is necessary in our method since Kolmogorov complexity is defined only for discrete objects. Its choice will necessarily affect the results.

In order to make computation easier, we will often use the continuous approximation of this term obtained by ignoring the flooring step:

$$K(X) \simeq \log \left(1 + \frac{|X|}{\delta}\right)$$

As we will show later, this approximation offers a very simple tool to design simple minimization algorithms. Besides, we observed empirically that this approximation does not affect the results much.

We can notice that this complexity value is upper-bounded by the L^1 norm on $\mathbb{R}$. This observation can be generalized to any vector, if Hypothesis 1 is assumed to be true. The Kolmogorov complexity of a vector X is then upper-bounded by the L^1-norm, up to the precision parameter δ. This upper-bound could be used especially in the algorithms we present. We will discuss their impact in Section 8.3.2. Another remark is that this value of complexity corresponds to the case where the complexity is not prefix. A possibility to make it prefix would be to use a doubling code for instance, which would increase the complexity of x of a constant $\log C(x)$ where $C(.)$ designates the plain complexity.

8.3.1.2 Complexity of vectors

We consider two categories of vectors in our model: absolute position vectors and relative position vectors.

Absolute position vectors encode a position on the whole vector space $\mathcal{X}$. We consider that these vectors are encoded on a fixed number of bits, which means that their complexity is constant. This assumption is motivated by two arguments. Firstly, from a description point of view, this hypothesis is a way to make the description length translation invariant. This invariance property is desirable since it prevents from the results to be influenced by the choice of the origin. Secondly, this assumption can be interpreted in terms of Bayesian prior. Using the probabilistic equivalent of complexity $p(x) = 2^{-K(x)}$, it comes that a non-constant complexity of absolute position vectors would correspond to a non-uniform prior over the input space, which has to be avoided for the arguments evoked above.

Relative position vectors encode a position relative to another point of reference. These vectors encode a difference between two positions, hence are not subject to

the same limitations as absolute position vectors. Thus, using the complexity of real numbers as defined in Equation 8.8 and Hypothesis 1, we consider a machine such that the complexity of a d-th dimensional vector v is defined by:

$$K(v) = \sum_{i=1}^{d} K(v^i) + \mathcal{O}(1) \tag{8.9}$$

As noted in the case of real numbers, this complexity term is upper-bounded by the L^1 norm of vector v.

8.3.1.3 Complexity of prototype model transfer

In order to transfer the prototype model from the source domain to the target domain, we aim to characterize the transformation of the source model returning the target model. We propose a multi-level approach to describe these changes, based on global transformations, class transformations and local transformation:

- **Global transformation**: A first transformation affects all prototypes in the model, independently of their class.
- **Class transformation**: For each class, a transformation is then applied, that affects all prototypes belonging to the class.
- **Local transformation**: The final transformation depends on each prototype individually.

The use of such a multi-level approach enables one to take global effects into account, such as a global translation of the data, or effects shared by points of the same class. Global and class transformations are used as a factorization of common transformations that affect all or groups of prototypes. The solution to describe a transformation for each of the prototypes in the model is tempting at first sight but would actually increase the complexity by repeating shared information.

The choice of three levels here is arbitrary: It could be possible to work with an arbitrary number of levels. Our choice is mainly motivated by the simple interpretation which can be done of the results with this point of view ("global" and " by class" having a strong semantic meaning). In the general case, determining the number of levels could be done by applying the MDL principle. A large number of levels will make the movement descriptions more compact but will require a larger description, and thus are not necessarily optimal. On the contrary, when the prototype model consists of a low number of prototypes, the shortest description can be provided by only one or two levels. We do not propose any investigation on this question here.

We propose to restrict our study to the simple case where $\mathcal{X}_S = \mathcal{X}_T$ but the global methodology can apply to the case $\mathcal{X}_S \neq \mathcal{X}_T$ as well. Having a common representation space makes it possible to choose translations as simple transformations for this study. Other families of transformations could be studied (for instance affine transformation[2], and in particular, in the two-dimensional case, rotations). The three-step transformation process with translations is depicted on a simple example in Figure 8.2.

[2] An interesting property of affine transformations is that they can be used also for cases where source and target spaces have different dimensions.

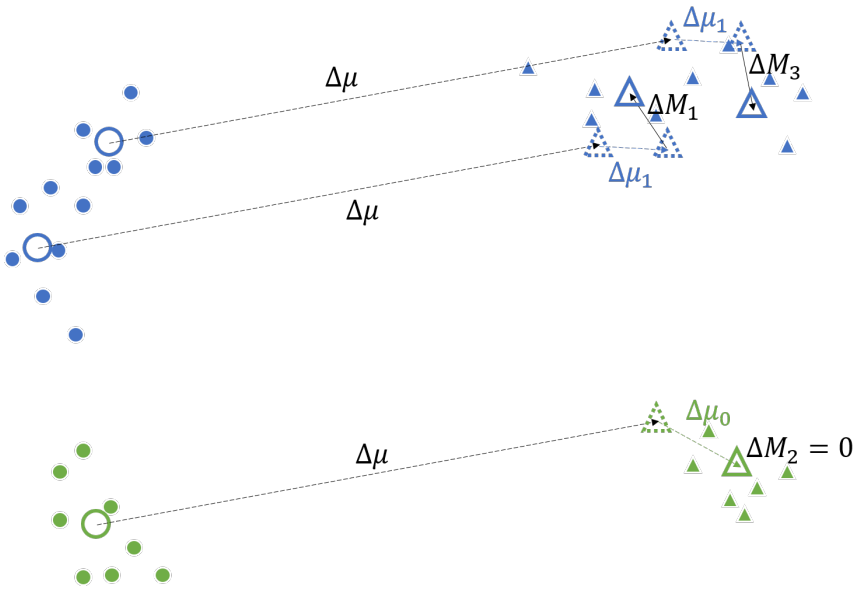


FIGURE 8.2: Transfer of a prototype model from source domain (circles) to target domain (triangles) for a two class problem (class 0 in green and class 1 in blue). Prototypes are represented by large empty symbols. The transformation is in three step. First, all prototypes are translated with a common vector $\Delta\mu$. Then prototypes inside a same class c are translated with a vector $\Delta\mu_c$. Then prototype i is translated with a vector ΔM_i. Intermediate prototypes are dashed.

The global transformation affects the whole prototype structure and corresponds to a general translation of all prototypes. This transformation is meant to model the global change of means that can be observed between source and target distributions. In the case where the data are centered during pre-processing, this translation vector is supposed to be equal to zero. In general, the global translation of the prototypes is described by the coordinates of the translation vector $\Delta\mu$.

In practice, the vector $\Delta\mu$ can be initialized as the difference of means between target data points and source data points. However, this assumption might be wrong in some cases. Consider for example the case in which only one class is translated from the source to the target. It costs more in terms of Kolmogorov complexity to translate all points by a vector $\mu^{(T)} - \mu^{(S)}$ and then to translate each class individually to its position, than to only translate the right class to its new position.

Once the global transformation has been applied to the model, we may wish to characterize the common changes shared by a whole class of points. Such a class transformation is defined relatively to the global transformation. If l is a class label, we designate by $\Delta\mu_l$ the class translation vector. The complete class transformation is given by the set of vectors $\{\Delta\mu_1, \ldots, \Delta\mu_{|\mathcal{Y}|}\}$ where $|\mathcal{Y}|$ is the total number of classes.

The local transformation is the residual transformation to completely describe the position of a target prototype given the source prototype model M_S. This transformation is applied to each prototype after the first two transformations.

In practice, unlike global and class transformations, a local transformation can consist of three actions:

- **Move a prototype**: This action, which is by nature a translation, is encoded by the relative position vector.

- **Create a prototype**: This action is encoded by the class index and the relative position vector in the class.
- **Remove a prototype**: This action is encoded by the index of the prototype to remove.

In terms of complexity, the creation of a prototype costs more than its suppression: This observation is consistent with the intuition that it is easier to simplify the model than to make it more complex.

The relative position vectors are put together in the local transformation matrix ΔM. For each prototype i of class l, the local transformation ΔM_i of prototype i is defined as:

$$\Delta M_i = M_i^{(T)} - M_i^{(S)} - \Delta\mu_l - \Delta\mu \tag{8.10}$$

In the proposed experiments, we will not consider the question of prototype creation or removal which is postponed to future researches. Algorithmic solutions to this problem already exist in other frameworks, such as proposed by (Grbovic and Vucetic, 2009) for Learning Vector Quantization.

Given the source model M_S, the transfer to M_T with the previously described transformations is described by a vector $\Delta\mu$, a set of vectors $\{\Delta\mu_1, \ldots, \Delta\mu_{|\mathcal{Y}|}\}$ and a matrix ΔM corresponding to the individual local transformations. The index of prototypes to delete and the coordinates of prototypes to create would be required if these operations were considered.

In practice, the global construction of the target models can be summed up by the following procedure (including prototype creation and removal):

1. Delete the specified prototypes.
2. Apply a translation of vector $\Delta\mu$ to all prototypes.
3. For each class l, apply a translation of vector $\Delta\mu_l$ to all prototypes in the class.
4. Apply a translation of vector ΔM_i to all prototypes i.
5. Concatenating the new prototypes.

The corresponding Kolmogorov complexity (not considering creation and removal) is given by:

$$K(M_T|M_S) = K(\Delta\mu) + \sum_{l=1}^{|\mathcal{Y}|} K(\Delta\mu_l) + K(\Delta M) \tag{8.11}$$

8.3.2 Algorithm

Based on the complexity values described in Section 8.3.1, we propose an algorithm to minimize objective function 8.1.

8.3.2.1 A Class of Functions

As a tool for the algorithm, we propose to study a simple class of functions defined as follows: Let $n > 0$ and $A = (a_1, \ldots, a_n) \in \mathbb{R}^n$ be n real numbers. We define function Λ_A as follows:

$$\Lambda_A(x) = \sum_{i=1}^{n} \log\left(1 + |x - a_i|\right) \tag{8.12}$$

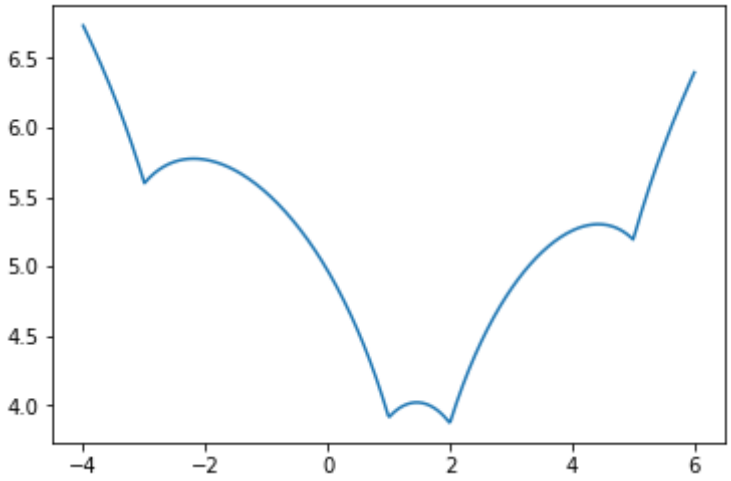


FIGURE 8.3: Plot of function $\Lambda_{(-3,1,2,5)}$

In Figure 8.3, we display the plot of function Λ_A for $n = 4$ and $A = (-3, 1, 2, 5)$. We can observe on the plot that the local minima of Λ_A seem to be reached at points $x \in \{a_1, \ldots, a_n\}$. This conjecture is demonstrated in the following proposition.

Proposition 6. *Let Λ_A defined as in Equation 8.12. The set of local minima of Λ_A is a subset of A.*

Proof. We first consider that the a_i are distinct two by two. We assume, without loss of generality, that $a_1 < a_2 < \ldots < a_n$.

On the portion $(-\infty, a_1]$, we have $\Lambda_A(x) = \sum_{i=1}^{n} \log(1 + a_i - x)$. The function is derivable and its derivative is equal to

$$\Lambda'_A(x) = -\sum_{i=1}^{n} \frac{1}{1 + a_i - x} < 0$$

hence Λ_A is decreasing on this portion. In the same way, it can be shown that it is increasing on $[a_n, \infty)$.

On the segment $I_k = [a_k, a_{k+1}]$, we observe that $\Lambda_A(x) = \sum_{i=1}^{k} \log(1 + x - a_i) + \sum_{i=k+1}^{n} \log(1 + a_i - x)$. The function is twice derivable on the segment and its second derivative Λ''_A is equal to

$$\Lambda''_A(x) = -\sum_{i=1}^{k} \frac{1}{(1 + x - a_i)^2} - \sum_{i=k+1}^{n} \frac{1}{(1 + a_i - x)^2} < 0$$

which means that Λ_A is concave on I_k. Based on this property, we have that the local minima of Λ_A on I_k are included in $\{a_k, a_{k+1}\}$.

These two observations prove the proposition in the case where the a_i are distinct two by two. When the a_i are not distinct, the corresponding terms in the sum are multiplied by a positive constant (the arity of the corresponding a_i), which does not modify the result. □

Corollary 2. *If $\delta > 0$, $A \in \mathbb{R}^n$ and function $\Lambda^{\delta}_A : \mathbb{R} \to \mathbb{R}$ is defined by $\Lambda^{\delta}_A(x) = \sum_{i=1}^{n} \log(\delta + |x - a_i|)$, then the set of local minima of Λ^{δ}_A is a subset of A.*

This property will be used to improve the performance of the subsequent algorithms for the minimization of complexity.

8.3.2.2 Unlabeled Data Description without Transfer

As a first step in the direction of minimum complexity transfer, we propose to study the optimal description of an unlabeled dataset before any transfer. This problem can be reduced to the following optimization problem:

$$\text{minimize} \quad K(P) + \sum_{i=1}^{n} K(X_i|P) \tag{8.13}$$

where the complexities are defined as presented above. Assuming that the number of prototypes is fixed, we can rewrite the objective function of Equation 8.13 as:

$$K(P) + \sum_{i=1}^{n} K(X_i|P) = \sum_{i=1}^{n} \min_{p \in P} \sum_{j=1}^{d} \log\left(1 + |X_i^j - p^j|\right) + \mathcal{O}(1) \tag{8.14}$$

This objective function is similar to the objective function of K-means and can be minimized in a same way. We propose a minimization algorithm in two steps: In a first step, we associate each datapoint to the prototype that minimizes the description complexity of the point. In a second step, we optimize the position of the prototypes with fixed point-prototype association. These steps are alternating until convergence.

Step 1 consists in a simple comparison of the possible complexity values for each prototype. The time complexity of this step is $\mathcal{O}(n \times |P|)$, proportional to the number of points and the number of prototypes. We denote by $I(p)$ the set of indexes of points attached to prototype p.

In **step 2**, it is assumed that prototypes are associated to a set of points. We denote by $X(p)$ the subset of data points associated to prototype p. The objective function for step 2 is the following:

$$\sum_{p \in P} \sum_{j=1}^{d} \sum_{i=1}^{|X(p)|} \log\left(\delta + |p - X(p)_i^j|\right) = \sum_{p \in P} \sum_{j=1}^{d} \Lambda^{\delta}_{X(p)^j} \tag{8.15}$$

where Λ designates the function defined in Equation 8.12 and $X(p)^j$ is the vector made up of the j-th coordinates of points in $X(p)$. The form of this equation suggests that the minimization can be done prototype by prototype and coordinate by coordinate, by simply evaluating the $\Lambda^{\delta}_{X(p)^j}$ function on points in $X(p)^j$ (an evaluation of this function has a linear complexity). Consequently, the complexity of this step is $\mathcal{O}(n^2 \times d)$.

Algorithm 1 sums up the procedure detailed above. It consists in alternating the two steps until convergence. The time complexity is linear in the number of data. In practice, the computation time can be reduced by using parallel computations over the data points in step 1 and over the prototypes and dimensions in step 2. Besides, the proximity to K-Means algorithm allows one to prove that the algorithm converges in a finite number of steps, as well as to apply all the improvement tricks developed for K-Means, including better initialization (Pena, Lozano, and Larranaga, 1999) or massive parallel versions (Zhao, Ma, and He, 2009). These variants have not been used in the context of this work.

Algorithm 1: Unlabeled prototype-based data description with MDL

Data: Data points X, Number of prototypes
Result: Prototypes P
Initialize prototypes P;
do
  Initialize prototype-data association: $I(0) = \ldots = I(p) = \emptyset$;
  for $i = 1 \ldots n$ **do**
    $\hat{p} \leftarrow \arg\min_p \sum_{j=1}^{d} \log\left(\delta + |X_i^j - p^j|\right)$;
    Add i to $I(\hat{p})$;
  for $p \in P$ **do**
    for $i = 1 \ldots d$ **do**
      $p^j \leftarrow \arg\min_{k \in I(p)} \sum_{i=1}^{n} \log\left(\delta + |X_k^j - X_i^j|\right)$;
while *convergence not reached*;

8.3.2.3 Labeled Data Description without Transfer

The second algorithm we propose, based on a prototype-based model for data description, provides a description of labeled data. It is very similar to the algorithm proposed for unlabeled data.

Unlike in previous algorithm, the prototypes are now associated to a label $y \in \mathcal{Y}$. Data points are expected to be attached to a prototype with the same label. As suggested in Equation 8.7, when a point is attached to a prototype with a different label, this results in a penalty of $\log |\mathcal{Y}|$ bits.

The algorithm we propose for this problem is similar to algorithm 1 but has an additional step in order to update the label of the prototype. When the set $I(p)$ of points attached to a prototype p is determined, the label of the prototype is determined by minimizing the number of corrections needed in $Y_{I(p)}$ (or in other words the number of errors). The resulting algorithm is presented in Algorithm 2.

Algorithm 2: Labeled prototype-based data description with MDL

Data: Data points (X, Y), Number of prototypes
Result: Prototypes P
Initialize prototypes P;
do
  Initialize prototype-data association: $I(0) = \ldots = I(p) = \emptyset$;
  for $i = 1 \ldots n$ **do**
    $\hat{p} \leftarrow \arg\min_p \sum_{j=1}^{d} \log\left(\delta + |X_i^j - p^j|\right)$;
    Add i to $I(\hat{p})$;
  for $p \in P$ **do**
    $Y^{(p)} \leftarrow$ majoritary label in $Y_{I(p)}$;
    for $i = 1 \ldots d$ **do**
      $p^j \leftarrow \arg\min_{k \in I(p)} \sum_{i=1}^{n} \log\left(\delta + |X_k^j - X_i^j|\right)$;
while *convergence not reached*;

8.3.2.4 Prototype-based Transductive Transfer with Simple Transformation

We now consider the case of transductive transfer with the model described in section 8.3.1.3. For simplicity purposes, we consider here only the first step of the transformation, ie. a global translation of vector $\Delta\mu$. The global objective function, to minimize under P, $Y^{(P)}$ and $\Delta\mu$ is then:

$$\sum_{i=1}^{n_S} \min_{p\in P} \left(\sum_{j=1}^{d} \log\left(\delta + |X_i^j - p^j|\right) + \mathbb{I}(Y_i \neq Y^{(p)}) \right) + \sum_{j=1}^{d} \log\left(\delta + |\Delta\mu^j|\right) + \sum_{i=1}^{n_T} \min_{p\in P} \sum_{j=1}^{d} \log\left(\delta + |X_i^j - p^j - \Delta\mu^j|\right) \tag{8.16}$$

where n_S is the number of source data and n_T the number of target data. This objective is made up of three parts: the first sum corresponds to the description length of source data, the second sum to the description length of the translation vector $\Delta\mu$ and the last sum to the description length of target data. The first term is the same as the objective minimized in Section 8.3.2.3.

We propose an algorithm of the same nature to minimize objective function 8.16. In a first step, we fix the point-prototype association in both source and target domains, using the current value of P and $\Delta\mu$. The second step consists in finding the corresponding optimal values for the variables. The chosen strategy we propose is an alternating minimization: minimization over P with fixed $\Delta\mu$ and minimization over $\Delta\mu$ with fixed P.

For the minimization over P, we simply observe that the objective is a Λ function defined with a concatenation of two vectors: The position of the source data points $X^S(p)$ and the position of the target data points after a translation of vector $\Delta\mu$ (which will be written $X^T(p) - \Delta\mu$ for simplicity purposes). If we denote the concatenation operator with symbol $\oplus$, the j-th component of prototype p is the minimum of function $\Lambda^{\delta}_{X^S(p)^j \oplus (X^T(p)^j - \Delta\mu^j)}$. We notice that the set $X^T(p) - \Delta\mu$ can be interpreted as a projection of target data onto source domain since the transfer is supposed to be a translation. The minimization over $\Delta\mu$ is simpler and simply consists of a minimization of function $\Lambda^{\delta}_{(0)\oplus(X^T(p)^j - p^j)}$. The whole algorithm is summed up in Algorithm 3.

Algorithm 3: Prototype-based unsupervised domain adaptation with MDL

Data: Source data points (X^S, Y_S), Number of prototypes
Result: Source prototypes P, Translation vector $\Delta\mu$
Initialize prototypes P;
Initialize translation vector $\Delta\mu$;
do
  Initialize source prototype-data association: $I^S(0) = \ldots = I^S(p) = \emptyset$;
  for $i = 1 \ldots n_S$ **do**
    $\hat{p} \leftarrow \arg\min_p \sum_{j=1}^{d} \log\left(\delta + |(X^S)_i^j - p^j|\right)$;
    Add i to $I^S(\hat{p})$;
  Initialize target prototype-data association: $I^T(0) = \ldots = I^T(p) = \emptyset$;
  for $i = 1 \ldots n_T$ **do**
    $\hat{p} \leftarrow \arg\min_p \sum_{j=1}^{d} \log\left(\delta + |(X^T)_i^j - \Delta\mu^j - p^j|\right)$;
    Add i to $I^T(\hat{p})$;
  for $p \in P$ **do**
    $Y^{(p)} \leftarrow$ majoritary label in $Y_{I(p)}$;
  do
    for $p \in P$ **do**
      for $j = 1 \ldots d$ **do**
        $p^j \leftarrow \arg\min_{x \in X^S(p)^j \cup (X^T(p)^j - \Delta\mu^j)} \Lambda^{\delta}_{X^S(p)^j \oplus (X^T(p)^j - \Delta\mu^j)}(x)$;
    for $j = 1 \ldots d$ **do**
      $\Delta\mu^j \leftarrow \arg\min_{x \in \{0\} \cup (X^T(p)^j - p^j)} \Lambda^{\delta}_{(0) \oplus (X^T(p)^j - p^j)}(x)$;
  while *convergence not reached*;
while *convergence not reached*;

The general case with the transformation proposed in Section 8.3.1.3 is a direct adaptation of Algorithm 3 with one more step corresponding to the class translation. We do not present the complete algorithm for simplicity purposes and due to its high similarity to the case of global translation only.

8.3.3 Measuring the quality of transfer

Unsupervised domain adaptation is not a well-posed problem; consequently, even if it is possible to define a *classification error rate* over a labeled target set, this quantity does not measure exactly the efficiency of a transfer method.

A transfer learning problem has multiple solutions, our approach consisting in selecting the most simple solution in terms of algorithmic complexity. In some problems, even human experts cannot make the distinction between two solutions and, in this sense, penalizing an inversion of two classes in the result of a method would seem to be arbitrary.

The misclassification rate (or error rate) expresses how far the classification results are from the actual labels. This rate can be calculated for source and target data (as the source model and the drift are learned simultaneously). Given a set of points $\{X_1, \ldots, X_n\}$ and their respective labels $\{Y_1, \ldots, Y_n\}$, the misclassification rate of a classifier h is defined as:

$$R_n(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(Y_i \neq h(X_i)) \tag{8.17}$$

Because of the previous observation, this quantity has to be considered carefully as a high misclassification rate does not necessarily imply a bad transfer.

As seen above, non-normalized empirical risk can be seen as an upper-bound of complexity $K(Y|X, M)$, since each classification error has to be corrected. As a consequence, we propose another quality index for transductive transfer, that we call *quality of transfer*, and which measures the quality of the transferred target model, after the target labels Y_T are available:

$$Q_n(M_T) = 1 - \frac{K(Y_T|M_T, X_T)}{K(Y_T)} \tag{8.18}$$

In the case where Hypothesis 1 holds, the complexities are equal to the sum over the dataset: $K(Y_T) = \sum_{i=1}^{n_T} K(Y_{T,i})$ and $K(Y_T|M_T, X_T) = \sum_{i=1}^{n_T} K(Y_{T,i}|M_T, X_{T,i})$.

Unlike empirical risk, the quality of transfer does not measure directly the quality of the solution in terms of number of errors, but rather the complexity to produce the actual solution based on the inferred solution. This difference is particularly interesting in cases where $\mathbb{P}_S[X] = \mathbb{P}_T[X]$ but $\mathbb{P}_S[Y|X] \neq \mathbb{P}_T[Y|X]$. An example of this situation is the problem of class inversion, when the source labeling function $f_S : \mathcal{X} \to \mathcal{Y}$ is transformed into a target labeling function $f_T : \mathcal{X} \to \mathcal{Y}$ such that $f_T(x) = 1 - f_S(x)$. This simple transformation cannot be seen without labels in the target domain.

8.3.4 Toy examples

We test our method on two-dimensional toy examples built artificially. We consider two parameterized problems:

- **Class translations**: The input points are generated by two normal distributions. The drift consists of a translation of the means of each of the distributions. The transformation is parameterized by the translation vector.
- **Class deformation** (Figure 8.4): one of the class is continuously deformed from a vertical line to a circle surrounding the second class. The deformation is parameterized by a real number $\theta \in [0, 1]$.

The class translation problem seems particularly adapted to our method since it relies on a global translation of the distribution, which is the natural bias of our algorithm. However, class deformation and half-moons are more complex but provide a good way to parameterize the difficulty of the transfer. Depending on the value of the parameter (θ for class deformation and α for the inter-twinning moons), the transfer requires more complex adaptation of the model.

8.3.5 Results and discussion

The **class translation** problem has been tested on automatically generated sets of 200 points in $\mathbb{R}^2$. In the source, the first class is generated by a normal distribution centered on $(0, 0)$ and the second class by a normal distribution centered on $(2, 0)$. Both distributions have identity covariance matrix. In the target, the same distribution is used for the first class, but the second class is derived from a normal distribution centered on $(t, 0)$.

The results obtained for the transfer from source to target highly depend on the parameter $t \in \mathbb{R}$ (Figure 8.5).

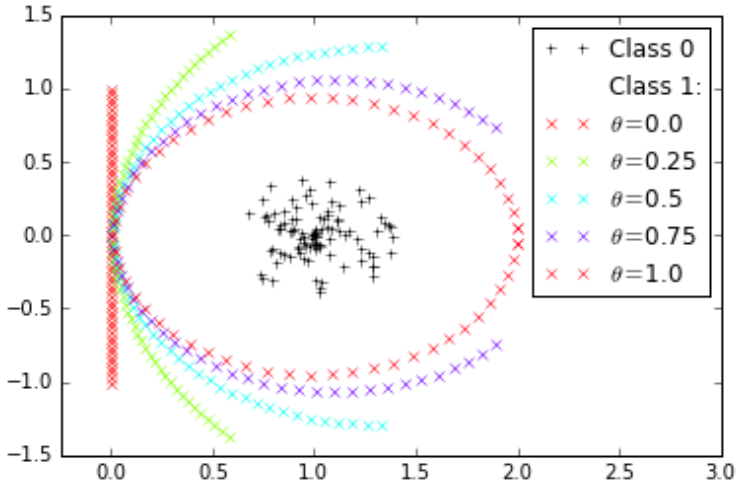


FIGURE 8.4: Toy problem for transfer learning with various difficulty levels. The distribution of class 0 (plotted as black +) doesn't change. The distribution of class 1 (plotted as colored crosses) is parameterized by a real number θ. When $\theta = 0$, the points are aligned on a vertical line; when $\theta = 1$, the points are distributed on a circle surrounding class 0.

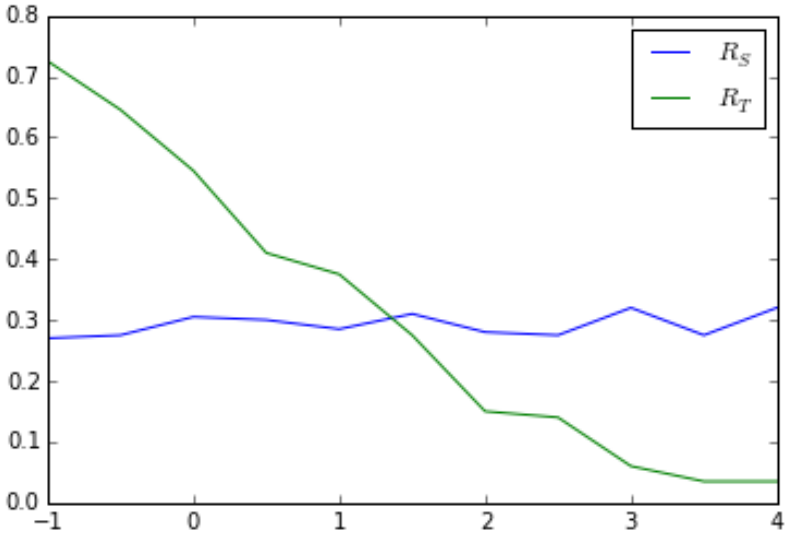


FIGURE 8.5: Evolution of the classification error over the source (R_S) and target (R_T) with the translation parameter t (x-axis).

The source error remains approximately constant for all values of the parameter: the value of the error is due to the noise. When $t < 0$, the situation consists basically in an inversion of the order of the centers along the x-axis. Such an inversion cannot be deduced by any method without further instruction; our method relying on a simplicity principle, it avoids the class inversion (which would be far too complicated), and thus leads to a high target error. However, using our quality index Q_n, we would observe a large value of Q_n, due to the fact that the model could correctly describe the structure of the target but permuted the labels.

When t is close to 0, the two classes are not separable and the high target error

TABLE 8.1: Misclassification rate for transfer (left: in source; right: in target) with source data generated with a parameter θ_S and target data generated with a parameter θ_T.

	$\theta_T = 0$	$\theta_T = 0.2$	$\theta_T = 0.4$	$\theta_T = 0.6$	$\theta_T = 0.8$	$\theta_T = 1$
$\theta_S = 0$	0%, 0%	0%, 0%	0%, 8.99%	0%, 20.2%	0%, 43.8%	0%, 50.6%
$\theta_S = 0.2$	0%, 0%	0%, 0%	0%, 2.25%	0%, 21.3%	0%, 5.62%	0%, 52.8%
$\theta_S = 0.4$	6.74%, 6.74%	5.62%, 0%	0%, 0%	0%, 0%	0%, 5.62%	1.12%, 34.8%
$\theta_S = 0.6$	0%, 0%	4.49%, 1.12%	8.99%, 7.87%	0%, 0%	0%, 1.12%	7.87%, 11.2%
$\theta_S = 0.8$	8.89%, 77.5%	11.2%, 15.7%	8.99%, 10.1%	1.12%, 1.12%	7.87%, 6.74%	8.99%, 44.9%
$\theta_S = 1$	6.74%, 77.5%	13.5%, 58.4%	7.87%, 19.1%	10.1%, 0%	10.1%, 11.2%	14.6%, 14.6%

rate is due primarily to this non-separability. For values of t larger than 2, the transfer is done as expected and the target error rate is quite low. We can note that this value keeps decreasing: the target problem becomes more and more separable.

For **class deformation**, the misclassification error has been calculated for various transfer situations: the source data are generated by our predefined process with parameter θ_S and the target data are generated with parameter θ_T. After the learning step, we calculate the misclassification rate on the source and on the target. The obtained results are summed up in Table 8.1.

As in the class translation problem, the misclassification rates are lower in the source: this is because misclassification is penalized directly in the learning for source data. The results show that the method has difficulties adapting to topologically different situations: when θ_S is low and θ_T is high, many errors occur due to the method which essentially preserves the position of the prototypes from source to target. The cost of the structure adaptation for the model in difficult transfers is too high.

8.4 Conclusion

In this chapter, we have shown the similarity between analogical reasoning and transductive transfer learning. These similarities led us to considering that the same DGM developed for analogy can also be used to describe transductive transfer problems. We have presented a couple of experimental results that tend to justify our principle from an empirical point of view. For these examples, we considered a very simple application framework with a naive data description.

We have also shown that the proposed objective function is consistent with other objective functions which are typical of machine learning: In particular, we have seen that maximum likelihood principles or empirical risk minimization scheme can be interpreted in terms of complexity minimization. However, the purpose of transductive transfer is not to provide a generalization of the observations, unlike for induction. The link with inductive reasoning, and in particular the question of generalization, will be explored in next chapter.

Chapter 9

Beyond Transfer: Learning with Concern for Future Questions

In the previous chapter, as well as in the part related to analogical reasoning, we have focused on a very specific task, which consists in transferring the learned concept onto a given target problem, in particular onto a given dataset. This problem is called *transduction*. However, two questions arise from these preliminary results. The first question regards the feasibility of such a transductive transfer: Is it possible to determine that two datasets are related? The second question concerns an extension of the framework: How are transductive transfer and inductive transfer related?

The solutions we propose to these problems are closely related to the framework of complexity-based transfer. The key notion developed in this chapter is the notion of problem transferability. Intuitively, a model is transferable to a target problem if it helps compressing this problem. Two alternative definitions will be given, and we will show that all models cannot be transferred to a target problem. Based on this observation, we expose that complete generalization is not possible and propose a simple inductive framework based on this idea of impossible transfer.

The remainder of this chapter is organized as follows: In a first section, we discuss quickly how the previously introduced model for transductive transfer can adapt to many other transfer learning settings. Then, we discuss the possibility of a universal transfer based on our complexity-based framework. Finally, in Section 9.3, we propose an inductive framework that is based on strong assumptions on the future.

9.1 Supervised and Semi-Supervised Problems with Transfer and without Transfer

In this section, we extend the methods developed in Chapter 8 to cases where some or all labels are available in target domain. We also provide evidence that this method can be used in the absence of transfer.

9.1.1 Supervised and Semi-Supervised Domain Adaptation

The methods presented in Chapter 8 are designed to deal with transductive transfer, also referred to as unsupervised domain adaptation. In this framework, no labels are available in the target domain. Another problem of interest in transfer learning is the case where labels are available in target domain. In this case, called *supervised domain adaptation*, the learner is given a source dataset $\mathcal{D}_S = \{(X_i^S, Y_i^S)\}_{i=1,\dots,n_S}$ and a

target dataset $\mathcal{D}_T = \{(X_i^T, Y_i^T)\}_{i=1,\dots,n_T}$ and aims to infer the classifiers $h_S : \mathcal{X}_S \to \mathcal{Y}_S$ and $h_T : \mathcal{X}_T \to \mathcal{Y}_T$.

The supervised domain adaptation problem can be assessed by the same graphical model as transductive transfer. The objective function to minimize, in the general case, is the following:

$$K(M_S) + K(X_S|M_S) + K(Y_S|M_S, X_S) + K(M_T|M_S) + K(X_T|M_T) + K(Y_T|M_T, X_T) \tag{9.1}$$

Compared to the objective of transductive transfer, this objective includes the complexity term $K(Y_T|M_T, X_T)$ which models the number of errors done by the classification on the target dataset.

In fact, this representation of transfer does not imply that the labels are available. It can be considered that the labels Y are empty, which counts for a zero complexity. It comes that the label vector Y_T can be sparse. This situation corresponds to semi-supervised transfer. In this case, the presence of labels in target helps the transfer compared to the pure transductive situation.

When applied to the previously described prototype-based model, the objective function of Equation 9.1 can be evaluated as presented in Section 8.3.1.

9.1.2 Absence of Transfer

In all the contexts presented until now, no assumption is made on the nature of data, and in particular on their distribution. The absence of assumption is central in the context of transfer, since transfer learning differs from classical supervised learning by ignoring the assumption of identically distributed data (from source to target). However, the i.i.d. case is a particular case of transfer, where no transfer is needed and, intuitively, $M_S = M_T$.

We propose here to study the particular case where the distribution is the same in source and target tasks.

Theoretically speaking, supposing that (X_S, Y_S) and (X_T, Y_T) are i.i.d. imposes a symmetry in the objective function of Equation 9.1: For a given model M, the terms $K(X_S|M)$ and $K(X_T|M)$ are i.i.d. too. The same observation holds for the terms $K(X_S|M, X_S)$ and $K(X_S|M, X_S)$. From there, it comes that the distributions of models M_S and M_T should be close, under a condition of continuity on the complexity (if two objects are close, their complexity is close). This idea gives a good intuition that, in this case, $M_S = M_T$.

When $M_S = M_T$ (denoted simply by M), Equation 9.1 can be simplified: The complexity term $K(M_T|M_S)$ is clearly equal to zero. If we denote by $\oplus$ the concatenation of two vectors or matrices, the objective function can simplify into:

$$K(M) + K(X_S \oplus X_T|M) + K(Y_S \oplus Y_T|M, X_S \oplus X_T) \tag{9.2}$$

which corresponds to a simple objective for MDL principle in a supervised setting.

However, traditional machine learning differs from what is done in this example. In supervised learning, only one dataset is available at training time, and the model learned from it is supposed to apply to future data that are generated by the same distribution. In the case of transductive transfer, the target dataset is known and the purpose is to find a target model that works well on this dataset. The nature of the two learning problems is then very different: In the first case (transductive transfer), the learned model is supposed to apply to a given set of data, while in the

second case (supervised learning) the model is supposed to work well for a potentially infinitely-many datasets that are not observed yet.

Applying the transductive transfer model to solve supervised learning problem is not intended to work well for generalization, but we observe empirically that it is the case. The good performances that can be observed show that generalization is possible using a merely descriptive approach. Intuitively, this idea makes sense: The observed data are supposed to be representative of the distribution.

These observations raise two new questions. The first question concerns the problem of **generalization**, which is central in machine learning. The main purpose of most learning methods is to infer a decision function from a given dataset that is well-adapted to *any* new data. It is well-known, since the no-free-lunch theorem (Wolpert, 1996), that no learning algorithm can generalize well on all datasets. We propose to revisit this result in next section. The second question is a direct consequence of this observation. Since it is not possible to learn a universal function, is it possible to know if a learned model is **transferable** to a new problem?

9.2 Impossibility of Transfer?

In supervised learning, it has been shown, in a probabilistic setting, that a learning algorithm cannot perform well on any kind of problem. The purpose of this section is to define some notions that could be of interest in the perspective of finding a similar result for transfer learning.

In this section, we consider that the system attempts to describe pairs of problems $X \in \mathbb{X}$ and solutions $Y \in \mathbb{Y}$ based on intermediate objects $M \in \mathcal{M}$, called models. The space $\mathbb{X}$ is called *input space* (or *problem space*); the space $\mathbb{Y}$ is called the *output space*; and the set $\mathbb{M}$ is called the *class of models*. In the following, we might omit to mention these sets when it is obvious.

9.2.1 Two Notions of Transferability

The transfer learning principle based on MDL principle and presented in Equation 8.1 suggests that the transfer does not need to operate at the level of instances but at the level of underlying models. A model is an object factorizing information about observed entities.

In order to define transferability, we propose two similar but not equivalent notions of learnability based on this theoretical tool. These notions will be then extended to define transferability.

9.2.1.1 Learnability from Source Model

The first definition of learnability considers that a problem $X \in \mathbb{X}$ is learnable by a model $M_S \in \mathbb{M}$ if giving M_S as a parameter in the estimation of optimal model for the description of X has a gain in compression:

Definition 13. *Given a model $M_S \in \mathbb{M}$ and an integer $\eta > 0$, a problem X is called weakly (M_S, η)-learnable if the following property holds:*

$$\min_M \{K(M) + K(X|M)\} \geq \min_M \{K(M|M_S) + K(X|M) + \eta\} \tag{9.3}$$

This notion of learnability means that providing the model M_S to the learner will help finding a new description of the problem shorter to the optimal description by η bits. In particular, we have necessarily that a problem will be weakly learnable with respect to its optimal model (ie. the model that minimizes $K(M) + K(X|M)$). A weakness in this definition is the fact that the models M defined in the left-hand side and in the right-hand side of inequality 9.3 are different. This problem is solved by defining an alternative notion of learnability.

The second definition of learnability suggests that the optimal model that can be used for the description of problem X can be compressed when the source model M_S is given. Unlike previous definition, M_S is not directly involved in the description of X.

Definition 14. *Given a model $M_S \in \mathbb{M}$, a problem $X \in \mathbb{X}$ is called strongly (M_S, η)-learnable if the following property holds:*

$$M = arg\min_M \{K(M) + K(X|M)\} \Longrightarrow K(M) \geq K(M|M_S) + \eta \tag{9.4}$$

These two notions of learnability based on a source model have a couple of interesting properties. In particular, it can be shown that they are correlated.

Proposition 7. *For any model $M_S\mathbb{M}$, any problem $X\mathbb{X}$ and any parameter η, if X is strongly (M_S, η)-learnable, then X is weakly (M_S, η)-learnable.*

Proof. Consider a problem X that is strongly (M_S, η)-learnable and call M^* the optimal model: $M^* = \arg\min_M \{K(M) + K(X|M)\}$. We have then:

$$\begin{aligned}\min_M \{K(M) + K(X|M)\} &= K(M^*) + K(X|M^*) \\ &\geq K(M^*|M_S) + \eta + K(X|M^*) \\ &\geq \min_M \{K(M|M_S) + K(X|M) + \eta\}\end{aligned}$$

which proves the proposition. □

A priori, the converse is not true: weak learnability does not imply strong learnability. This is the consequence of the fact that the models implied in Equation 9.3 are not the same on the right hand side and on the left-hand side. However, we have no counter-example and we have not formally proved that weak learnability does not imply strong learnability.

9.2.1.2 Properties of Learnability

In the following proposition, we group a couple of direct properties of learnability (either weak or strong). Their proofs are trivial and are not given.

Proposition 8. *Let $M_S \in \mathbb{M}$ be a source model and $X\mathbb{X}$ a problem. The following properties are true:*

1. *If X is (M_S, η)-learnable, then X is (M_S, η')-learnable for all $\eta' \leq \eta$.*
2. *If M minimizes $K(M) + K(X|M)$, then X is (M, η)-learnable for all η.*
3. *If M_S is empty ($M_S = \langle\rangle$), then no problem X is (M_S, η)-learnable for $\eta > 0$.*

As a last property of learnability, we show in the following proposition that the notion of learnability is related to a notion of information factorization, in the sense that no bit of information can be added or removed.

Proposition 9. *For any model $M_S \in \mathbb{M}$, if there exists a parameter $\eta > 0$ and a problem $X \in \mathbb{X}$ such that X is strongly (M_S, η)-learnable, then $K(M_S) \geq \eta - c_{\mathcal{M}}$, where $c_{\mathcal{M}}$ is the machine constant in the chain rule (ie. for machine $\mathcal{M}$ and for any objects x and y, $K(x) = K(y) + K(x|y) + c_{\mathcal{M}}$).*

Proof. If X is such a problem and M minimizes the objective $K(M) + K(X|M)$, then by definition $K(M|M_S) \leq K(M) - \eta$. Using chain rule on $K(M)$, we obtain that

$$K(M|M_S) \leq K(M|M_S) + K(M_S) + c_{\mathcal{M}} - \eta$$

□

We notice that Proposition 9 is not informative if the price of the chain rule (hence the constant $c_{\mathcal{M}}$ is too high). This constant is necessary yet and cannot be ignored as done in other applications: Unlike our previous considerations which focused on comparing several programs, we consider here one program only, which implies that the constants have to be held. In practice, this restriction is important but does not affect the intuition behind the notions at play. For a first interpretation, it is possible to ignore the constant ($c_{\mathcal{M}} = 0$). If a problem is (M_S, η)-learnable under this hypothesis, then there exists $\eta' \geq \eta$ such that the problem is (M_S, η')-learnable when considering a positive chain rule constant.

9.2.1.3 Transferable Problems

The notion of learnability we proposed in definitions 13 and 14 are not directly applicable to measure transferability. The property of transferability measures the ability to transfer knowledge from a solved (labeled) problem (X_S, Y_S) to apply it on the unsolved target problem X_T. It can be seen as an extension of learnability where the source model is determined from the source solved problem.

Definition 15. *Let $(X_S, Y_S) \in \mathbb{X} \times \mathbb{Y}$ be a solved source problem. The problem (X_S, Y_S) is said to be strongly (resp. weakly) η-transferable to the problem $X_T \in \mathbb{X}$ if the set of feasible models $\{M_S \in \mathbb{M} | K(M_S) + K(X_S|M_S) + K(Y_S|M_S, X_S) < K(X_S) + K(Y_S|X_S)\}$ contains an element M_S^* such that X_T is strongly (resp. weakly) (M_S^*, η)-learnable.*

This notion of transferability is interesting to consider in the perspective of task relatedness as defined in Section 7.3.3. The introduced notions are in contrast with the presented measures of task relatedness in the idea that it does not provide a measure of divergence between two distributions in a domain, but between a model (that can be a distribution) and a dataset. Our notion is more general in the sense that it does not rely on any probabilistic setting, and focuses on the description of input data, and not on labels. In this sense, it differs from (Zhang, Zhang, and Ye, 2012). The choice of not describing the item will be justified below, in a discussion on negative transfer.

Finally, the difference between our definitions and the divergence introduced by (Mahmud, 2009) is complicated. The first main difference is the purpose. Our approach does not focus on defining a measure of task-relatedness, but to what extent a learned model can be applied to a future problem. Another major difference is the symmetry of the considered divergence, that is not observed in our framework. For us, the order of the task (source or target) plays a role in the notion of relatedness.

9.2.2 Non-Transferability and Negative Transfer

The notion of negative transfer is inherent to transfer learning. As exposed in Section 7.3.1, negative transfer designates a situation where the source problem brings *negative* information to the resolution of the target problem. A simple example of negative transfer in everyday life is the case of false friends in language learning: The minimum complexity transfer principle at play implies that, if one word in source language is similar to a word in target language, they must have the same meaning. In practice, it is often not the case, which corresponds to negative transfer.

A first conjecture to define the notion of negative transfer would consist in **associating non-transferability to negative transfer**. This association would imply, in particular, that negative transfer comes from a forced transfer when no transfer is actually feasible. This situation is the context of the analysis of (Rosenstein, Marx, Kaelbling, and Dietterich, 2005).

Another more accurate conjecture to define negative transfer makes use of the problems' solutions that are completely ignored by the notion of transferability. In this conjecture, a negative transfer corresponds to a problem onto which the source model is transferable but does not give good results. The example of false friends is a perfect illustration of this phenomenon: the transfer is feasible but is not correct.

Another example in the prototype-based model can be found in the class translation problem given in Section 8.3.4. When a permutation of the two classes is observed, the problem is clearly transferable using a source prototype-model describing the source distribution. The transfer is described by a simple translation and offers a very good compression of data. As observed, the empirical risk is very high for such situations, whereas the quality of the model, as defined in Equation 8.18, is very low.

9.3 Learning with Concern for Future Questions

In this section, we explore a generalization of transfer in a context of multiple targets. We will show how such problems can be solved and are related to the fundamental question of induction. The solution we propose is called *learning with concern for future questions* and offers larger possibilities than statistical models of learning.

9.3.1 Transfer to Multiple Targets

In Chapter 7, we described two distinct learning problems: multitask learning and transfer learning. On the one hand, in multitask learning, the system addresses several tasks in parallel and exploits their relatedness in order to speed up the computation. On the other hand, transfer learning introduces an asymmetry, by isolating a *source task* which is supposed to contain the meaningful knowledge, and transferring it to a *target task*.

Multitask learning can be summed up by the DGM given in Figure 9.1: In this DGM, the compression of each ordered pair (X_i, Y_i) (where X_i does not represent only one instance but a batch of data) is done by a model M_i. Common information about all tasks, hence all models M_i is factorized into a *meta-model* M. A similar DGM will be presented with more details in Chapter 14 for the problem of multi-source clustering.

In this DGM, the meta-model M plays a prominent role: It influences all other local models. It is specifically designed to this purpose (in the training phase, the

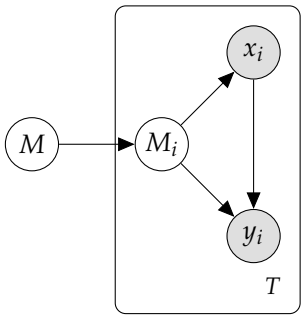


FIGURE 9.1: Model-based DGM for multitask learning.

meta-model is trained to minimize the global description length). In multi-target transfer, this model is chosen in order to describe a source task (see DGM in Figure 9.2). This corresponds to a situation where a source problem is available, as well as multiple target problems. The corresponding objective function is:

$$K(M_S) + K(X_S|M_S) + K(Y_S|M_S, X_S) + \sum_{i=1}^{T} K(M_i|M) + K(X_i|M_i) + K(Y_i|X_i, M_i) \tag{9.5}$$

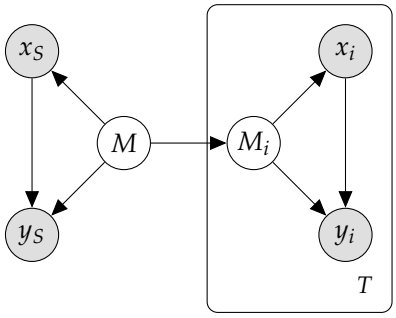


FIGURE 9.2: Model-based DGM for multitask learning.

The main weakness of this representation is the lack of impact of the source over numerous targets. In particular, there is no sequential aspect in which the source model is learned first, then it is used for the transfer to target problems. Here, the representation of the source problem is dictated by the representation of the target problems.

9.3.2 Transfer, Transduction and Induction: Which Links?

In the presented approach of transfer, the purpose was to solve a specific task. This task is known in advance, at learning time. A learning problem of this kind is called *transductive*, after the concept of transduction introduced by (Vapnik, 2006). The idea of transduction, initially developed for SVMs (Gammerman, Vovk, and Vapnik, 1998), is to solve a simple task directly, instead of solving a larger and more difficult task as an intermediate step. In transductive transfer, the problem is to learn a correct representation for the target data. Knowing if this representation can be generalized is absolutely not the main concern.

In many real-life problems, the tasks are not that well-posed, and especially the target problem may not be available at learning stage. This is the case in particular in the traditional supervised learning framework where the learning algorithm is completely blind to the test data which are used only for the evaluation of the

learned classifier. This means that the learning criterion expressed in Equation 8.1 cannot be used directly in these contexts. Whereas the aim of Transfer Learning is to find an answer to a specified problem (the *target* task), the major issue of induction is *generalization*: The system has to find a representation (or, more specifically, a decision function) which will work well not on a a given problem but on **any** problem *similar* to the source problem. The purpose of this section is to define properly what is meant by "any" and "similar".

Even if these two notions seem to be contradictory, there exist clear links between the two of them.

- **From induction to transduction:** The link is direct, induction being the acquisition of general knowledge, while transduction is the acquisition of task-specific knowledge. Transduction could be seen as a direct application of induction. However, this is not a correct way to reason: Generalization should not be an intermediate step to solve simple specific problems (Vapnik, 1995).
- **From transduction to induction:** Generalization from one situation is equivalent to applying a task-specific knowledge to any problem. From this point of view, induction might be interpreted as transduction applied to multiple targets.

The concept of Kolmogorov complexity is fundamental in the understanding of induction. It has been introduced in the seminal works of Ray Solomonoff on induction (Solomonoff, 1964). These works are based on the theory of Bayesian learning and focus on infinite sequences. Given an observed infinite sequence x, the question of induction consists in predicting the next elements in the sequence. Knowing that the observed phenomenon is given by a distribution μ, Bayes rule is expressed by $\mu(y|x) = \mu(xy)/\mu(x)$. Using this formalism, the question of induction consists in finding a good approximation of μ which is not computable. The strength of Solomonoff's theory of the universal induction is to show that there exists a universal distribution M that is a good estimator of *all* computable distribution μ. This distribution is related to Kolmogorov complexity: $M(x) \simeq 2^{-K(x)}$ (see Chapter 4 in (Li and Vitányi, 2008) for more details on the construction of M).

The convergence theorem, proved in various contexts by (Solomonoff, 1978) or (Hutter, 2003), compares the quality of the estimation based on universal distribution to the actual distribution and proves that the estimation converges to the actual quality when the size of the input x increases.

This theory of induction gave birth to another theory, inspired by both the theory of the universal induction and reinforcement learning. AIXI (Hutter, 2001) is a sequential approach which considers that the events faced by the system are distributed with an unknown distribution μ. This distribution is estimated with the universal distribution M. Despite its promise to be the first universal intelligent agent, AIXI has some limitations, such as its dependency on rewards (hence on a teacher) and its convergence to suboptimal states (Orseau, 2010). In order to overcome these difficulties, the Knowledge-Seeking Agent (Orseau, 2014) is defined in an active setting (its actions affect the environment) but does not adapt its behaviour with regards to rewards.

Our point of view is very different from these methods, but illustrates in an alternative way that complexity is a good candidate to model intelligence. The main difference between our approach and the described models (Solomonoff's theory,

AIXI and Knowledge-Seeking Agent) is the probabilistic point of view. These methods rely on an approximation of distributions (posterior distributions in Bayes rule or true distributions in a reinforcement learning environment), while we consider general and a priori non probabilistic environments. Besides, our approach is more cognitively inspired, since it does not aim to assess the actual state of the word but relies on an *a priori* knowledge of the future. We will discuss later the possibility to consider a universal distribution for the future states.

9.3.3 Learning with No Future in Mind

The first solution that is usually chosen in machine learning to solve the problem of induction with the MDL principle consists in considering a non-existing target, hence $X_T = \langle\rangle$. This assumption is consistent with the idea that target data are not available at learning stage.

When choosing $X_T = \langle\rangle$, the complexity term $K(X_T|M_T)$ is equal to 0, which leads to a general objective function:

$$K(M_S) + K(X_S|M_S) + K(Y_S|M_S, X_S) + K(M_T|M_S) \tag{9.6}$$

In this objective function (to be minimized over models M_S and M_T), the target model is only present in the transfer term $K(M_T|M_S)$ which is minimal for $M_S = M_T$. From this observation, it comes that not considering a target at learning stage is similar to considering a stationary process in which the target can be modeled with the same representation as the source. When models are probability distributions, this corresponds to the assumption that source and target data are identically distributed.

9.3.4 Including Priors over the Future

Supposing that the i.i.d. hypothesis holds, as suggested by the empty-target hypothesis, is a very strong assumption. It implies that the system expects the target data to be chosen inside a very limited class of data. Other classes of problems can be chosen, which would correspond to other assumptions over the future. In order to allow these classes, we propose a methodology inspired by the remarks exposed in Section 9.3.2.

We denote by $Cr(M_S, M_T; X_S, Y_S, X_T)$ the quantity defined in Equation 8.1. Function Cr corresponds to the learning criterion to be minimized over source and target models M_S and M_T when observations (X_S, Y_S, X_T) are fixed. In this notation, we split the arguments of the criterion function Cr in two: the models (which are variables in the optimization process) and the data. In cases where the source data are obvious, we may abusively omit them in the notation of the criterion and thus write $Cr(M_S, M_T; X_T)$ for simplicity reasons.

Consider that the numberé N of possible target questions is finite and positive. This hypothesis makes sense in the context of a discrete machine, hence in any computer-based approach. We denote by $X_T^{(i)}$ the i-th possible target problem. Each of them is described by a model $M_T^{(i)}$.

Unlike in multi-target transfer as described earlier, the system does not have access to the future target problem, but has to choose among various possible targets. For this reason, induction is different from multi-target transfer. The solution we propose considers all the transfer problems (transferring from source data (X_S, Y_S) to target data X_T) as possible problems to solve, instead of considering all possible

target data X_T as future states. Based on this idea, learning a generalization consists in finding the models M_S and $M_T^{(i)}$ which are optimal *for all potential target questions*. In mathematical terms, this coincides with the following **multi-objective optimization** problem:

$$\underset{M_S, M_T^{(1)}, \ldots, M_T^{(N)}}{\text{minimize}} \quad \left[Cr(M_S, M_T^{(1)}; X_T^{(1)}), \quad \ldots \quad , Cr(M_S, M_T^{(N)}; X_T^{(N)}) \right] \tag{9.7}$$

Because single-objective optimization and multi-criterion optimization are fundamentally different (Ehrgott, 2000), the notion of optimality of a solution is different in both domains. The notion of Pareto optimality corresponds to a state where it is impossible to improve on one objective without losing on the others. More formally, if $f_1, \ldots, f_n$ are n objective functions defined on a set $\mathcal{X}$ and $S \subset \mathcal{X}$, a feasible solution $x^* \in S$ of the problem:

$$\underset{x \in S}{\text{minimize}} \quad [f_1(x), \quad \ldots \quad , f_n(x)]$$

is called:

- *Pareto optimal* if there exists no $x \in \mathcal{X}$ such that $f_k(x) \leq f_k(x^*)$ for all k and $f_i(x) < f_i(x^*)$ for some i.
- *weakly Pareto optimal* if for all $x \in S$, $f_k(x) < f_k(x^*)$ for all k.

It can be shown (see for example Proposition 3.9 in (Ehrgott, 2000)) that the solution x^* of the scalarized single objective optimization problem

$$\underset{x \in S}{\text{minimize}} \quad \sum_{i=1}^{n} \lambda_i f_i(x)$$

is weakly Pareto optimal if $\lambda_k \geq 0$ for all k, and Pareto optimal if $\lambda_k > 0$ for all k.

Applied to the problem of Equation 9.7, this proposition implies that, if we defined some positive weights λ_i for $i = 1, \ldots, N$, then the models minimizing

$$\sum_{i=1}^{N} \lambda_i Cr(M_S, M_T^{(i)}; X_T^{(i)}) \tag{9.8}$$

form a Pareto optimum. In the following, we will consider that such solutions are actual solutions of the generalization problem.

The choice of the vector parameter λ is arbitrary in the sense that any positive vector will lead to a Pareto optimal solution, however in practice the value of λ works as a weighting of the objective functions. Intuitively, it is reasonable to give a higher weight to target problems which are *more likely* to be encountered. This choice of parameter λ corresponds to a prior over the future. Considering a normalization of λ (hence $\sum_{i=1}^{N} \lambda_i = 1$), and the two-parts expression of Cr, the first part being shared by all the $Cr(M_S, M_T^{(i)}; X_T^{(i)})$, we finally obtain the following objective function:

$$K(M_S) + K(X_S|M_S) + K(Y_S|M_S, X_S) + \sum_{i=1}^{N} \lambda_i \left[K(M_T^{(i)}|M_S) + K(X_T^{(i)}|M_T^{(i)}) \right] \tag{9.9}$$

We consider the minimization of this objective function as an inductive principle that we call *Learning with Concern for Future Questions* (LCFQ). This framework is guided by the idea that the learning system may have some prior over the future.

Two remarks can be done on Equation 9.9. Firstly, the function does **not** correspond to a complexity value and does not make any sense in terms of algorithmic information theory. The initial elements are complexities but the weighted sum (corresponding to the Pareto) is not. Secondly, the weighted sum over target terms can be seen as an expected value, where the terms λ_i correspond to the probability over target dataset $X_T^{(i)}$. Finally, the role played by the source in Equation 9.9 is completely different from its role in Equation 9.5. In multi-target transfer, the description of source data is influenced equally by all targets. Here, the source complexity remains the prominent term of the objective, and all target elements do not have the same weight in the description, which makes this objective both more general and more adapted to induction.

9.3.5 Some Priors for Future Questions

The learning objective defined in Equation 9.9 relies on the definition of a distribution defined by the parameters λ_i. As explained, this distribution can be interpreted as a prior of the learner about the future problem he might face. This prior is defined at the learning time and biases the interpretation of data. A first possibility consists in having these coefficients explicitly given. For instance, the future state can be given in advance to the learner with some uncertainty. Transfer is a particular case of this situation, since it corresponds to the case where $\lambda_i = 1$ for the target dataset and $\lambda_i = 0$ for all other datasets.

More interestingly, we can consider that the target points are independent and identically distributed with a probability distribution q. We also consider that the future dataset will contain exactly n points. The expected value of Kolmogorov complexity for the future can be then computed, using Hypothesis 1:

$$\begin{aligned}
\mathbb{E}_{X_1,\dots,X_n\sim q}\left[K(X|M)\right] &= \int_{X_1}\dots\int_{X_n} p(X_1,\dots,X_n)K(X_1,\dots,X_n|M)dX_n\dots dX_1 \\
&= \int_{X_1}\dots\int_{X_n}\left(\prod_{j=1}^{n} p(X_j)\right)\left(\sum_{i=1}^{n} K(X_i|M)\right)dX_n\dots dX_1 \\
&= \sum_{i=1}^{n}\int_{X_i} p(X_i)K(X_i|M)\left(\int_{X_{-i}} p(X_{-i})dX_{-i}\right)dX_i \\
&= \sum_{i=1}^{n}\int_{X_i} p(X_i)K(X_i|M)dX_i \\
&= n\mathbb{E}_{X\sim q}\left[K(X|M)\right]
\end{aligned}$$

In particular, when M is a probabilistic model (associated to distribution p), we have $K(X|M) = -\log p(X) + \mathcal{O}(1)$ and $\mathbb{E}_{X\sim q}\left[K(X|M)\right] = D_{KL}(q||p) + H(q) + \mathcal{O}(1)$. Based on this expression, and using the definition that $K(Y_S|X_S, M_S) = n_S R_{n_S}(h_p)$, the inference problem becomes:

$$\underset{p}{\text{minimize}} \quad -\sum_{i=1}^{n_S}\log p(X_i) + n_S R_{n_S}(h_p) + n_T D_{KL}(q||p) \tag{9.10}$$

We observe that the estimated distribution is naturally biased by the prior toward the future distribution. When the number of source samples is very low compared to the number of target points, the solution of the minimization problem is $p = q$, which means that the source data are too few to have an actual impact onto the learning.

In practice, however, we can imagine that such a distribution is unknown. In the traditional i.i.d. setting, the solution consists in inferring the distribution, by considering that future data will have the same distribution as observed data. This assumption is very strong, but is of the same nature as any prior over the future: It consists in weighting the possible futures according to the probability that they are generated using the current distribution. Since the distribution is unknown, it is not possible to translate this idea directly into our framework. Future works on this idea must include a solution on this problem. In this direction, the idea of Solomonoff's theory seems particularly promising: Is it possible to use a universal distribution as an estimation of the prior over the future? However, in this case, the solution becomes incomputable, which is not admissible from a cognitive point of view nor in practice. Approximations of the universal distribution could be used.

9.3.6 Discussion: A general learning paradigm?

LCFQ appears to be very similar to other learning paradigms. In particular, we exposed that it corresponds to a more general version of machine learning with i.i.d. hypothesis over data. Moreover, we can see that the various terms in the source description correspond to objective functions that are commonly used in machine learning, such as log-likelihood or empirical risk. We want to conclude the chapter dedicated to LCFQ on a couple of remarks relative to learning in general.

In situation of designing a learning system, three major questions necessarily emerge, an answer to which is expected to get the system acting as desired. These questions are very general and apply for both human learning and machine learning.

Question 1: On which data and knowledge does the learner rely in order to learn? The question of the nature of data is rarely addressed, even if it is of major importance. Depending on the problem of interest, the data can be vectors, character strings, categorical values, binary values... Besides, data can be labeled, unlabeled, partially labeled. Apart from the choice of data, the knowledge of the learner also includes prior knowledge of the model and of the domain. Such knowledge are of major importance in human cognition (Chi, Glaser, and Farr, 2014), but are also present in machine learning with the notion of prior which is inherent in Bayesian learning. Finally, some additional information can be accessible to the learner, as depicted in the paradigme of Learning Using Privileged Information (Vapnik and Izmailov, 2015).

Question 2: What does the learner aim to learn? The question of the objective of learning has a main influence on how the system will effectively perform its learning task. In human cognition, it appears to be straightforward that the objective directs the learning process. Several objectives may be identified, such as rote learning (learning by heart: the learner has access to a set of solved problems and knows that he will be asked about problems among this learned set), transductive learning (solving on precise problem known in advance) or inductive learning (learning general concepts).

Question 3: On which machine does the learner try to learn? Even if the word "*machine*" is part of the expression "*machine learning*", the question of the machine used in the learning process is rarely addressed. However it is of great importance to know the strengths and limitations of the machine which is used to learn. Two understandings of the word machine can be considered here. On the one hand, the *computer* itself may have some importance in the learning process. For example, a machine with bounded-memory can be chosen for the learning. On the other hand,

the *machine* can be understood more formally as the Turing machine, ie. the program itself. The assumption of choosing a specific set of programs for learning is well-accepted in the machine learning field.

Thinking of a global learning theory would necessarily lead to addressing the question of building a learning framework which would be as general as possible and would be consistent with the questions considered above:

How to define a theory which would offer a *machine-dependent* approach of learning valid for *any kind of knowledge* and *any learning goal*?

To our knowledge, no existing framework offers such a full description of learning. Despite its simplicity, LCFQ seems to be a good candidate:

- The use of Kolmogorov complexity makes the principle agnostic to the representation of data. The only limitation is that data must be **representable** on a Turing machine, which is a very reasonable restriction in the domain of machine learning.
- The goal of the learner is directly present in the prior over the future. A goal corresponds to an expected future.
- The choice of the machine is present in the definition of complexity. We remind that complexity is defined for a fixed universal Turing machine. Physical and logical limitations can be added in the form of penalties inside the definition of function $K(.)$

Further research is needed in order to provide better guarantees for the good performance of this principle. In particular, it seems important to determine rigorously how to define the priors depending on the desired learning goals.

9.4 Conclusion

In this chapter, we have presented extensions of the minimum complexity transfer inspired by analogical reasoning. We exhibited a couple of interesting properties, including a possible extension of our method to targets where labels are available. From a more general point of view, we also introduced preliminary results on learnability and transferability. These results were used in particular to define negative transfer. Finally, we presented a general learning framework, extending transfer learning but including multiple potential targets. This framework, called Learning with concern for future questions, has been shown to be extremely general.

This chapter concludes the work we propose on transfer learning. Our purpose was to extend the procedure of minimum complexity analogies to machine learning problems where independence is not observed between training time and test time. In the following part, we will extend this approach and consider not only two time steps (training and test) but an arbitrary number of time steps. Such problems are typical of the domain of data stream mining, incremental learning and online learning.

Part III

Incremental Learning

Chapter 10

From Transfer Learning to Incremental Learning

Time is not necessarily a fundamental notion in transfer learning. Obviously transfer can be needed when the targeted concept evolved over time and is changed at test step. However, in most use cases of transfer learning, the target task is simply chosen from a completely different domain.

The domain of data stream mining is a perfect example of a context reuniting time dynamic and knowledge transfer. The increasing number of automatically generated data (for instance data generated by sensors, Internet of things or social media) leads to the emergence of new issues that offline learning cannot cope with. In non-stationary environments, the process generating these data may change over time, hence the learned concept becomes invalid. Adaptation to this non-stationary nature, called *concept drift*, is an intensively studied topic and requires mechanisms close to transfer.

The purpose of this chapter is to adapt the DGM proposed for analogical reasoning and transfer learning and to show its efficiency in data stream mining. We will show that this approach is consistent with state of the art techniques and has a valid probabilistic counterpart.

The remainder of this chapter is organized as follows. We first present the problem of online learning with concept drift: We provide the useful theoretical notions as well as the main trends to cope with concept drift. In Section 10.2, we propose a DGM for incremental learning and interpret the state of the art methods with regards to our framework. Lastly, we propose an application of our framework with two naive algorithms.

This chapter presents and extends the ideas introduced in (Murena, Cornuéjols, and Dessalles, 2017).

10.1 Introduction: Learning in Streaming Environments

In this section, we propose an overview of the problem of data stream mining (which will also be referred to as *incremental learning* or *online learning*). This introduction will not present the algorithms into more details, which will be done in next session.

10.1.1 A Recent Problem: Stream Mining

We presented transfer learning and domain adaptation as a first break of the traditional model of learning with two identically distributed datasets, one for the training and one for the testing. Data stream mining is another sub-domain of data mining that goes against this conception of learning.

The rise of data stream mining accompanies the emergence of new data generation processes. With the quick development of the Internet of Things (IoT), social media and mobile devices, the data generation rate keeps increasing. In 2012, (Gantz and Reinsel, 2012) estimates that over 2.8ZB of data were generated and processed (hence 2.8×10^{21} bytes) and that this number should be multiplied by 15 over 2020. This enormous and ever-growing amount of automatically generated data raises new challenges in the domain of data mining (Council, 2013). One of the challenges is the emergence of *data streams*: Instead of being generated in batches, data are produced one by one at very high and uneven rates, in a continuous and potentially unbounded fashion. As exposed by (Babcock, Babu, Datar, Motwani, and Widom, 2002), data stream mining differs in multiple ways from traditional data mining:

- **Online nature:** Data arrive one by one and are not accessible in batches. This forces the system to adapt and learn *online*. The very high rate requires the learning system to be able to handle new data on the fly and in real time.
- **Absence of control:** In batch learning, most methods perform multiple passes on the batch of data. Besides, the recent field of curriculum learning (Bengio, Louradour, Collobert, and Weston, 2009) encourages to reorganize the data for the system to learn more efficiently. Neither of these two ideas can be used with data streams since there is absolutely no control over the order of data.
- **Memory limitation:** Data streams can be unbounded or, in general, produce volumes of data that cannot be stored in memory. Elements of the streams are generally discarded after they are processed. Even if the system stores some data in memory, the size of the memory is necessarily small compared to the whole stream.

Another problem that is inherent to data stream mining is the temporal evolution of the data distribution. The stationarity hypothesis does not apply in most real-life situations, because of changes in users' behaviors, seasonality effects or physical changes (including aging effects). Seasonality can affect either physical measures (for instance by affecting sensors) or users' behaviors (which is taken into account in some recommendation appliations for instance (Hidasi and Tikk, 2012)). Existing applications of data stream mining with concept drift can be categorized into these two remaining categories of causes:

- **Changes in users' behaviors:** In several real-world scenarios, data are generated by human users in interaction with a service (a social media, a website, a connected object...), which implies that the data are affected by the changes that might happen in the users' behaviors. For instance, (Widyantoro, Ioerger, and Yen, 2003) studied online tracking of users' preferences. Such an adaptation process has direct applications in the field of recommender systems, as proposed in (Kuo, Chen, and Liang, 2009) for location-based mobile commerce, or in (Cao, Chen, Yang, and Xiong, 2009). The change in users' behaviors can also be taken into account in the analysis of web usage data, as presented by (Da Silva, Lechevallier, Rossi, and Carvalho, 2007).
- **Physical changes:** Sensors can be affected either by alterations of their capacities or by global changes in the environment. For instance, (Bessa, Miranda, and Gama, 2009) describes the importance of considering concept drift in wind speed in wind power forecasting (ie. predicting the output of wind parks). The same kind of evolution can also be observed in most applications to monitoring and control, including mass flow detection (Pechenizkiy,

Bakker, Žliobaitė, Ivannikov, and Kärkkäinen, 2010) and chemical activity prediction in a multitube reactor (Kadlec and Gabrys, 2011). A last applicative field is the biomedical domain. For instance, regarding the evolution of antibiotic resistance (Tsymbal, Pechenizkiy, Cunningham, and Puuronen, 2006; Tsymbal, Pechenizkiy, Cunningham, and Puuronen, 2008), it is observed that new pathogens can develop, against which previously effective antibiotics are ineffective. Clinical studies are also affected by such phenomenons, and by changes in human demographics (Kukar, 2003).

For a more complete overview of applicative fields, we refer the reader to the survey by (Žliobaitė, Pechenizkiy, and Gama, 2016).

10.1.2 Introducing Concept Drift

As introduced previously, concept drift corresponds to a change in the data distribution that happens during the streaming. We presented several possible factors that can cause a concept drift in different domains. We will now introduce the main ideas necessary to describe concept drift. These notions are now classical, and can be found in reference papers such as (Gama, Žliobaitė, Bifet, Pechenizkiy, and Bouchachia, 2014; Ditzler, Roveri, Alippi, and Polikar, 2015; Tsymbal, 2004; Webb, Hyde, Cao, Nguyen, and Petitjean, 2016).

In order to characterize concept drift, we adopt the perspective of statistical learning. A concept drift corresponds to a change in the joint distribution $p_t(X, Y)$ over time:

$$p_t(X, Y) \neq p_u(X, Y)$$

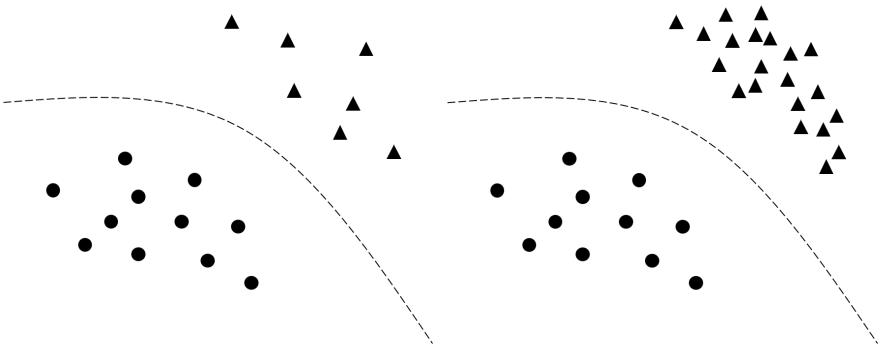
for two time steps $t \neq u$. Following the decomposition $p_t(X, Y) = p_t(X)p_t(Y|X)$, a concept drift can be the consequence of changes in either the posterior distribution $p_t(Y|X)$ or the distribution of non-class attributes $p_t(X)$. This leads to the following taxonomy (illustrated in Figure 10.1):

- **Real concept drift** characterizes a change in the posterior distribution $p_t(Y|X)$. This change can be accompanied by a change of the non-class attribute distribution $p_t(X)$. Less formally, real concept drift can be interpreted as a change in the class boundary.

- **Virtual concept drift** characterizes a change in the non-class attribute distribution $p_t(X)$. If this change is not accompanied by a change in posterior, the drift is called *pure covariate shift* by (Webb, Hyde, Cao, Nguyen, and Petitjean, 2016).

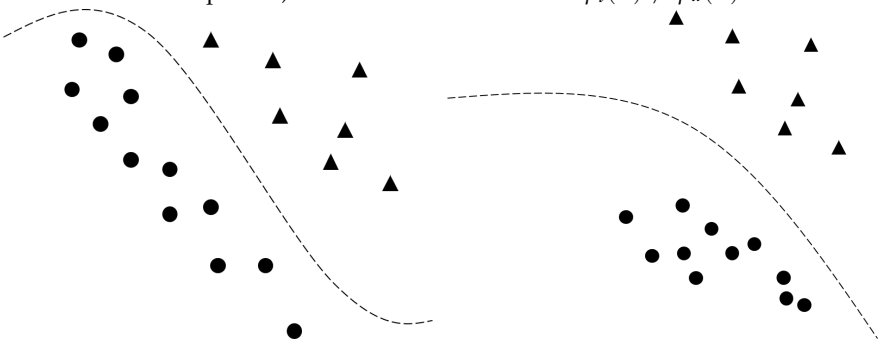
At first sight, virtual drift might seem to have less impact than real drift since it does not affect the class boundary directly. In practice, this idea is obviously incorrect and the distinction between real and virtual drifts is only formal. As pointed out by (Hoens, Polikar, and Chawla, 2012), the change in $p_t(X)$ may change the error of the learned model and thus require a retraining of the model. Equivalently, (Delany, Cunningham, Tsymbal, and Coyle, 2005) states that the learned boundary will change depending on the non-label data point distribution, which will obviously lead to a change in the error. Consequently, the distinction between real and virtual drifts is never taken into account in practical situations.

A special but noticeable kind of real drift is the case of novel class appearance. This problem appears in situations where new labels are observed, that have not appeared yet in the stream (Masud, Gao, Khan, Han, and Thuraisingham, 2011; Mu, Ting, and Zhou, 2017). This problem is particularly challenging when the feedback



(A) Source distribution $p_t(X, Y)$ (before concept drift)

(B) Distribution $p_u(X, Y)$ after real drift: $p_t(Y) \neq p_u(Y)$



(C) Distribution $p_u(X, Y)$ after real drift: $p_t(Y|X) \neq p_u(Y|X)$

(D) Distribution $p_u(X, Y)$ after virtual drift: $p_t(X) \neq p_u(X)$

FIGURE 10.1: Real and virtual drifts.

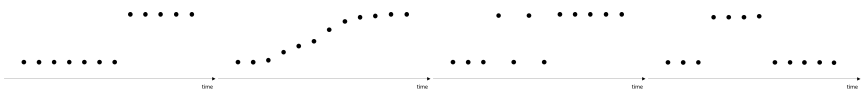
on the labels are not systematically given to the learner. From a theoretical point of view, it can be related to real concept drift, where the probability of emerging class c before the drift is $p_t(Y = c|X) = 0$.

Regarding the temporality of the process, the transition from one concept to the other can take several forms, as illustrated in Figure 10.2.

Abrupt drifts correspond to a very short transition from one concept to another. As suggested by (Webb, Hyde, Cao, Nguyen, and Petitjean, 2016), a threshold can be used on the duration of the transition in order to define if a drift is abrupt or extended. Such a threshold depends naturally on the application. A particular case of abrupt drift is the **blip drift**, ie. abrupt drifts accompanying a very short-lasting concept. Blips are most often considered as outliers and are thus ignored: The difficulty of handling concept drift is to distinguish between what is a new concept and what is only an outlier (Gama, Žliobaitė, Bifet, Pechenizkiy, and Bouchachia, 2014). However, (Webb, Hyde, Cao, Nguyen, and Petitjean, 2016) makes a distinction between outliers and blips: Outliers are isolate points, while blips are short-lasting sequences of equally distributed points.

Extended drifts (ie. non-abrupt drits) are mainly separated in two categories: incremental and gradual drifts. The distinction between these two categories is subtle and might differ from one author to the other. A strict distinction is given by the formalism of (Webb, Hyde, Cao, Nguyen, and Petitjean, 2016). An **incremental drift** designates a transition during which each new encountered concept is closer to the target concept and further from the initial concept. A **gradual drift** involves intermediate steps such that the distance between one concept and a later successor remains low. Following (Hoens, Polikar, and Chawla, 2012), this difference can be related to smoothness: the authors do not make a distinction between gradual and

incremental drifts, but we propose to interpret the distinction between them as a consequence of smoothness, incremental drifts being smoother than gradual drifts.



(A) Abrupt (B) Incremental (C) Gradual (D) Recurrent

FIGURE 10.2: Characterization of concept drift transition.

A last category of drifts are relative to the repetition of concepts. When a past concept is reused in the stream, the drift is said to be **reocurring** (or recurrent). These drifts are typical of seasonality effects. We will discuss them in Section 11.3.1. Reoccurrence is not necessarily periodic. A periodic recurrent drift is also called **cyclical drift**.

10.1.3 Passive and Active Methods

In practice, the main general setting for data stream mining, called *test-then-train* scenario, divides the process in three step:

1. **Prediction:** A new example X_t is received from the stream and classified into $\hat{y}_t$ using the up-to-date model.
2. **Diagnosis:** The true label y_t is received (in general at the same time as next example X_{t+1}) and the loss $l(y_t, \hat{y}_t)$ can be evaluated.
3. **Update:** The current model is updated according to the computed loss.

In this general schema, two questions arise: How to predict the label, and how to update the model? The first question depends on the chosen model. In general, the models are the same as classical supervised methods, hence the prediction simply consists in applying a pre-trained classifier on the observation.

The question of the update is more interesting and leads to the distinction between so-called passive and active methods.

For **passive algorithms**, the model is updated for each observation, without actively seeking for a change in data distribution. The main advantages of passive algorithms is that they are supposed to maintain an up-to-date model all along the stream.

Among the passive methods, methods based on decision trees are popular. In particular, Very Fast Decision Trees (Domingos and Hulten, 2000), an adaptation of Hoeffding trees to *stationary* data streams, have been modified to be used in *non-stationary* streams (Hulten, Spencer, and Domingos, 2001). This methods, called CVFDT, relies on instance selection with an adaptive sliding window strategy. Other similar methods rely on other tree constructions, such as the McDiarmid tree algorithm (Rutkowski, Pietruczuk, Duda, and Jaworski, 2013).

Among other popular approaches, passive methods are often based on ensemble models (Gomes, Barddal, Enembreck, and Bifet, 2017). In such approaches, a pool of classifier is maintained in memory and adapted according to the observations. In offline learning, ensemble methods are used mainly to improve the precision of classifiers. In online settings, ensemble methods has several advantages since they are particularly flexible regarding the addition or deletion of concepts. When a new concept is detected, they can adapt to it directly by adding a new model in the pool;

at the same time, they can easily discard and remove outdated models that are not used anymore. Moreover, they can continuously adapt the weights for the majority voting and hence be consistent with the current data distribution. These advantages have been observed theoretically (Ditzler, Rosen, and Polikar, 2014) with performance bounds inspired by the results of (Ben-David, Blitzer, Crammer, Kulesza, Pereira, and Vaughan, 2010), and in practice with algorithms such as Streaming Ensemble Algorithm (Street and Kim, 2001), Online Nonstationary Boosting (Pocock, Yiapanis, Singer, Luján, and Brown, 2010), Dynamic Weighted Majority (Kolter and Maloof, 2007), Learn^{++}.NSE (Elwell and Polikar, 2011), or Dynamically Expanded Ensemble Algorithm (Pietruczuk, Rutkowski, Jaworski, and Duda, 2017).

Unlike passive methods, **active algorithms** update the model only when necessary. Consequently, active approaches face two main challenges: detecting the concept drift and adapting the model. Following (Gama, Žliobaitė, Bifet, Pechenizkiy, and Bouchachia, 2014), drift detection techniques can be classified in several categories.

Sequential analysis relies on sequential statistical tests in order to detect a change in the distribution. The statistical tests used for sequential analysis follow similar directions as the Sequential Probability Ratio Test (Wald, 1973). This test evaluates the ratio of probabilities for data to be generated by a target distribution p_1 rather than a source distribution p_0. The cumulative sum (CUSUM) test is employed to track the deviation of the mean of a sequence from the value 0: this sequential test is frequently employed in data stream mining (Alippi and Roveri, 2006; Muthukrishnan, Berg, and Wu, 2007).

Control charts (or *Statistical Process Control*) groups together several statistical methods usually employed for quality control (Wheeler and Chambers, 1992). In their original case of use, these methods are supposed to track variations, and in particular deterioration, in processes. The idea of using SPC appears in particular in (Lanquillon, 2001) which suggests the use of Shewhart and CUSUM Control Charts, or (Bouchachia, 2011).

Approaches based on *sliding windows* store past information in fixed-sized or sliding windows and use them either to compare the past and present distributions or to detect changes directly in the current window (Bifet and Gavalda, 2007). Among these approaches, some use two windows, one for the past distribution and one for the present distribution. The data in both windows are then compared in order to determine if they are generated by the same distribution, with Chernoff bound in (Kifer, Ben-David, and Gehrke, 2004), entropy (Vorburger and Bernstein, 2006) or Kullback-Leibler divergence (Sebastião and Gama, 2007). Unlike these methods, ADaptive sliding WINdow (ADWIN) (Bifet and Gavalda, 2007) is based on one single window of increasing size: A drift is detected inside the window when it can be split into two sub-windows of very distinct mean values.

Since passive approaches are designed to adapt slightly at every time step, they are best suited for adaptation to gradual or incremental drifts. On the contrary, active approaches work well for streams with abrupt drifts that are easier to be detected. Active methods tend to detect drifts with a delay, which is higher in the case of incremental drifts for instance.

10.2 Minimum Complexity Transfer for Incremental Learning

In this section, we propose an extension of the DGM previously proposed for transfer learning. This extension is very inspired by Hidden Markov Models and can be used in various situations.

10.2.1 Notations for Online Learning

Let $\mathcal{P}$ be a problem space and $\mathcal{S}$ a solution space. At a time step t, the system receives a problem $X_t \in \mathcal{P}$ and aims at predicting the solution $\widehat{Y}_t \in \mathcal{S}$. After giving its prediction, the system may receive the actual solution Y_t to the given problem.

This formalism is consistent with several usual situations. In particular it covers the two cases described by (Read, Bifet, Pfahringer, and Holmes, 2012): *instance-incremental* and *batch-incremental learning* .

In *instance-incremental learning*, the system receives data one by one. At a step t, the learner receives a point $x \in \mathcal{X}$ where $\mathcal{X}$ is an input space (typically, $\mathcal{X} = \mathbb{R}^d$) and has to predict an output $y \in \mathcal{Y}$ (where the output space $\mathcal{Y}$ can be either continuous in regression, or finite in classification).

In *batch-incremental learning*, the learner receives a batch of data $x_1, \ldots, x_p$ and has to attribute a label $y_1, \ldots, y_p$ to each of the input points.

Using our notation is direct in both cases. In instance-incremental learning, the problem space and the input space are the same. In batch-incremental learning, a problem consists of a batch of instances of the input space $\mathcal{X}$.

In both cases, the problem at time t is denoted by X_t. In instance-incremental learning, $X_t \in \mathcal{X}$ is directly an element of the input space. If $\mathcal{X}$ is a vector space, we denote by X_t^i the i-th coordinate of the vector X_t. More generally, we will use the upper-script index to designate the coordinate of a vector. In batch-incremental learning, X_t is given in form of a list and we will designate by $X_{t,n}$ the n-th element of X_t. In particular, $X_{t,n}$ is an element of the input space $\mathcal{X}$.

10.2.2 A Graphical Model for Incremental Learning

We now propose a DGM for incremental learning. The proposed solution is based on the idea that online learning might be seen as successive steps of transfer learning. We will discuss why this assumption is not exact and what consequences these differences impose onto the model.

In order to deal with incremental learning, and inspired by the model introduced in Figure 5.5, we will consider that each time step t will be associated to a model M_t. This model is used to describe both the problem X_t and the solution Y_t. The problem X_t is described directly with the model M_t. The solution Y_t can be described either by itself or with the help of a decision function β_t (the classifier in a classification problem or regressor in a regression problem), induced by the model.

When the observations are entirely independent, the underlying models are independent, which means that the general process could be described by the DGM of Figure 10.3. The corresponding objective function is then:

$$\sum_{t=1}^{T} K(M_t) + K(X_t|M_t) + K(Y_t|M_t, X_t) \tag{10.1}$$

This hypothesis is very restrictive yet, and ignores completely the chronological aspect of online learning. In particular, models are estimated at each time step, which is not efficient in terms of computation time, of precision (complete models are learned from very few data, which might be a problem in some cases), and which loses all the information due to continuity (in the absence of drift or in the presence of incremental drift).

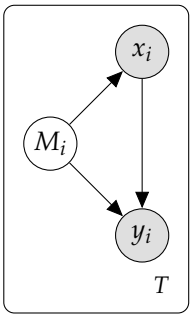


FIGURE 10.3: Model-based DGM for data stream with complete data independence.

In the opposite case, the models are entirely interdependent and the term associated to the model in the description length, $K(M_1, \ldots, M_T)$, cannot be simplified. This hypothesis is not satisfying either: Not only does it not exploit the temporal aspect of online learning, but, more importantly this expression is not allowed by the restrictions of online learning. Indeed, the computations would require too many resources as well as an access to the complete stream.

The approach we propose is intermediate. We adopt a Markovian point of view on the model description. At time step T, we consider that all the models M_t with $t \leq T$ are described using previously defined models. For any $t \leq T$, we define an association function

$$\Delta_t : \{1, \ldots, t-1\} \mapsto \{0, 1\} \tag{10.2}$$

such that $\Delta_i(j) = 1$ if model j is supposed to be involved in the description of model i, and $\Delta_i(j) = 0$ otherwise.

Using these association functions, we can express the complexity of the models up to time T as:

$$K(M_1, \ldots, M_T) = \sum_{t=1}^{T} K(M_t | M_{\Delta_t^{-1}(\{1\})}) \tag{10.3}$$

In this equation, the notation $\Delta_t^{-1}(\{1\})$ designates the set of indices i such that $\Delta_t(i) = 1$.

In practice, the association functions Δ_t can be either fixed by the system or learned online.

The choice of the functions Δ_t is of major importance in theory and in practice, because it offers to the system the possibility to store previously acquired knowledge and thus to memorize states of interest. A constant effort for obtaining this property has been deployed in recent techniques Hosseini, Ahmadi, and Beigy, 2013. Several choice scenarios can be considered in our case, which all correspond to state of the art methods (figure 10.4):

- When $\Delta_i(j) = 0$ for all i, j: all models are *a priori* independent in terms of description. The model is learned completely at each time step. This case corresponds to the model of Figure 10.3.

- When $\Delta_i(j) = 1$ for all i, j: the whole past models are taken into account to describe the present model.
- When $\Delta_i(j) = 1$ only for $j \geq i - h$ with a fixed h: the present model can be described with the last h models, which correspond to a sliding window of fixed size. The fixed sliding window is used in several algorithms such as FLORA.
- When $\Delta_i(j) = 1$ only for $j \geq i - h$ with a size h estimated by the system: the size of the sliding window is not fixed anymore but heuristically adapted to the current problem.

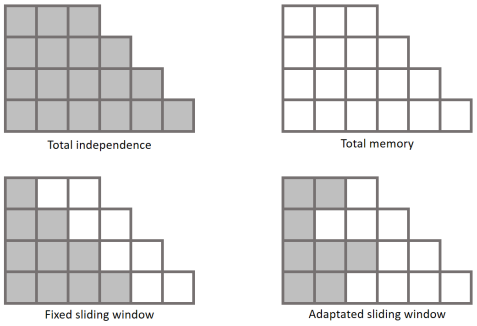


FIGURE 10.4: Possible choices for Δ function. The color of the square at line i and column j indicates the value of $\Delta_i(j)$. If the square is dark, $\Delta_i(j) = 0$; If the square is white, $\Delta_i(j) = 1$

Besides the choice of the association functions Δ_t have consequences over the minimization objective at each time step. Indeed, 1 values of these functions impose to add terms relative to previous observations in the complete Kolmogorov complexity. In practice, this is not always possible, in particular because older data are not stored in memory anymore. Hence, considering old states in the complexity is possible only with simplifying assumptions.

Based on this model description, we obtain an objective function of the form:

$$\sum_{t=1}^{T} K(M_t | M_{\Delta_t^{-1}(\{1\})}) + K(X_t | M_t) + K(Y_t | M_t, X_t) \tag{10.4}$$

which corresponds to the graphical model displayed in Figure 10.5.

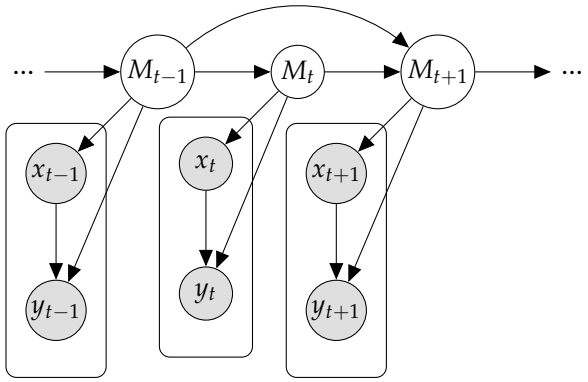


FIGURE 10.5: Model-based DGM for incremental learning

10.2.3 Remark: Estimating the Models Online

In offline learning, learning can be done in one single global step, considering all data together. In data stream mining, this is not possible, and the learning follows the pipeline presented in Section 10.1.3. We remind that this pipeline is made up of three steps: prediction, diagnosis, update.

When all possible data is available, it is possible to minimize the objective function 10.9 as a whole over $M_{:T} = (M_1, \ldots, M_T)$.

In practice, this minimization is complex for various reasons. One of them is the inter-dependency of the M_t which appear in several terms of the sum (because of the Δ_t functions). In particular, all models have to be learned again at each time step. In the perspective of data stream mining, this global update is obviously impossible and algorithmic hypotheses have to be chosen in order to overcome this difficulty.

We choose to adopt a *greedy* approach in order to solve the problem online. At time T, we consider that only the models for time steps in $\Delta_T^{-1}(\{1\}) \cup \Delta_{T-1}^{-1}(\{1\})$ can be modified. Older models are supposed to be fixed once and for all. This assumption is coherent with the incremental paradigm in which data are not stored and computations must be fast: There is no need to minimize the whole complexity, but only the terms corresponding to the most recent observations. Re-optimizing the whole modeling process would require to store all data $X_{:T}$ and would be highly time-consuming.

Hence, the optimization problem to solve at each time step is the following:

$$\begin{aligned} \underset{M_{T-1}, M_T}{\text{minimize}} \quad & K(M_{T-1}|M_{\Delta_{T-1}^{-1}(\{1\})}) + K(X_{T-1}|M_{T-1}) + K(Y_{T-1}|M_{T-1}, X_{T-1}) \\ & + K(M_T|M_{\Delta_T^{-1}(\{1\})}) + K(X_T|M_T) \end{aligned} \tag{10.5}$$

This assumption is not the only possible hypothesis one could rely on in order to solve the problem of online minimization of objective 10.9. However, it seems to us that it is the simplest and the most coherent in the perspective of data stream mining. It respects the temporal aspect of the stream by discarding older models from having a role in the most recent terms, and is more simple to assess than the whole minimization process that is not possible as far as the data stream becomes too long.

10.2.4 Classes of Models

In the part dedicated to transfer learning, we presented several classes of models that could apply for the transfer. We propose here the same kind of presentation, but we will present the classes of models following the passive and active classification of data stream mining algorithms. The purpose is to present how the proposed framework offers a common description to a large variety of existing methods.

10.2.4.1 Active Methods

Active approaches of incremental learning aim to detect change time steps t explicitly. Previous knowledge is used while no change has been detected.

In order to understand the way active methods work, we consider that, at time t, the model M_t is made up of the decision function β_t at time t and of a window of previous observations of size δ_t:

$$M_t = \langle \beta_t, X_t, Y_t, \ldots, X_{t-\delta_t}, Y_{t-\delta_t} \rangle$$

Using this model and considering a first-order logic for the model transfer, we can assess the complexity of each of the terms in Equation 10.9. In particular, the model transfer term $K(M_t|M_{t-1})$ can have several expressions:

- **Change of δ_t:** Points that were not present in the previous window have to be encoded explicitly. Only the positions X are strictly required, since the labels Y can be reconstructed from X and β. In case $\beta(X) \neq Y$, one additional bit is required.
- **Change of β:** Not only the new decision function β needs to be given, but the terms Y in the sliding window must be re-estimated.

If a drift is detected at time step n, the last $T-n$ terms in the sum are changed in order to take into account the introduction of a new decision function β' and of the new data description using β_1. As the complexity term $K(Y_t|X_t,\beta)$ corresponds to a correction term, a link can be established with empirical risk. Denoting $R_{n+1:T}(\beta)$ the empirical risk of binary classifier β on the last $T-n$ data, the detection of a break point obeys the criterion:

$$R_{n+1:T}(\beta) - R_{n+1:T}(\beta_1) > \frac{K(\beta')}{T-N} \tag{10.6}$$

This criterion is rather general and does not depend on the algorithm. In particular, we would like to discuss the case of ADWIN (Bifet and Gavalda, 2007). ADWIN stores all data one by one in a sliding window. At a time t, the algorithm decides whether it splits the window into two sub-windows, the first one being associated to the previous decision function, the second one being associated to a new decision function. This choice typically corresponds to the choice of the parameter n in previous equation. The major difference between ADWIN and the proposed solution is that the new decision function β' is not known by ADWIN. However, noticing that $R_{n+1:T}(\beta) - R_{n+1:T}(\beta') = R_{n+1:T}(\beta) - R_{1:n}(\beta) + R_{1:n}(\beta) - R_{n+1:T}(\beta')$, we can re-write the condition:

$$\begin{aligned} R_{n+1:T}(\beta) - R_{1:n}(\beta) &> \frac{K(\beta')}{T-N} + R_{n+1:T}(\beta') - R_{1:n}(\beta) \\ &\geq \frac{K(\beta')}{T-N} + R_{n+1:T}(\beta') - 1 \\ &\geq \inf_{\beta'} \left\{ \frac{K(\beta')}{T-N} + R_{n+1:T}(\beta') \right\} - 1 \end{aligned}$$

which corresponds to the form of the condition in ADWIN.

10.2.4.2 Passive Methods

Passive approaches of incremental learning do not consider abrupt changes but continuously adapt the learned decision functions to new incoming data.

Most passive methods rely on ensemble learning, and in particular on *bagging*. The key point of bagging is that at any time t the system relies on a pool of base learners $\{h_t^i\}_{1\leq i\leq N}$. In addition to the pool, a parameter w_t is needed to describe how these base learners are combined together. The model M_t can then be modeled as:

$$M_t = \langle h_t^1, \ldots, h_t^{N_t}, w_t \rangle$$

where N_t is the number of base learners at time t. This number can be fixed or evolve along the learning procedure. In order to express the value of Y_t using M_t and β_t, a method based on a majority vote of all expert learners is usually employed, thus the correction term $K(Y_t|M_t, X_t)$ can depend on all experts.

In order to deal efficiently with concept drift, ensemble methods have to remove outdated experts from the pool and replace them by more recent base learners. Several strategies are used to select learners to eliminate, including systematic removal of experts with performance lower than a threshold, or elimination of the worst base learner. These strategies are related to minimum description length, since a base learner is removed when the complexity of Y_t, depending on the learner to remove, is higher than the complexity of elimination.

The elimination of a base learner corresponds to a change in the model, which has a cost in terms of complexity. The higher value of $K(M_t|M_{\Delta_t^{-1}(\{1\})})$ has to be compensated by a lower value of $K(Y_t|M_t, X_t)$, hence a better performance. The same reasoning applies to the addition of a base learner, which can be done when the complexity of adding a new expert to the pool does not increase the overall complexity.

10.3 Algorithms

In this section, we will develop methods to solve optimization problem 10.5 in a context of incremental learning (i.e. with low memory and high speed).

10.3.1 Dealing with Previous Models

We propose to classify algorithms depending on the way they deal with previously acquired models. As mentioned earlier, the dependency on the past is given by the complexity term $K(M_t|M_{:t-1})$ which encodes the description length of model M_t at step t and the previous models $M_{:t-1}$ up to step t. Using the previously defined association functions Δ, the expression has already been simplified into $K(M_t|M_{\Delta_t^{-1}(\{1\})})$.

In order to describe model M_t with the help of the set $M_{\Delta_t^{-1}(\{1\})}$, two strategies may be chosen: either use many models in the set or select one single model.

The first strategy is employed in all ensemble learning methods (for instance in (Street and Kim, 2001), (Elwell and Polikar, 2009) or (Brzezinski and Stefanowski, 2014)). Such as in classical machine learning, ensemble learning methods construct the solution to a new problem by considering a weighted sum of the predictions of previous models.

In the second strategy, the key idea is to select the optimal model inside the set of predecessors $M_{\Delta_t^{-1}(\{1\})}$. The selected predecessor is the best model in the sense of MDL principle.

In the perspective of selecting one single predecessor for each model, the total objective of Equation 10.5 can be divided in two parts. The first part corresponds to the description of completed data at time step t once the solution has been given to the system:

$$\phi_1(M_{t-1}, M) = K(M_{t-1}|M) + K(X_{t-1}|M_{t-1}) + K(Y_{t-1}|M_{t-1}, X_{t-1}) \tag{10.7}$$

The second part corresponds to the description of incomplete data at time step t:

$$\phi_2(M_t, M) = K(M_t|M) + K(X_t|M_t) \tag{10.8}$$

Equation 10.5 can be reformulated as:

$$\underset{M_t, M_{t-1}, \tilde{M}_1, \tilde{M}_1}{\text{minimize}} \quad \phi_1(M_{t-1}, \tilde{M}_1) + \phi_2(M_t, \tilde{M}_2) \tag{10.9}$$

In the following, we propose a basic algorithm to solve problem 10.9 with a general class of models. This algorithm calculates model transformations at each step.

10.3.2 An Algorithm for Continuous Adaptation

In Continuous Adaptation Incremental Learning, the system infers a new model at each time step t for both model M_{t-1} and M_t. The system has access to all previously learned models $M_{:t}$ and chooses a predecessor among the models $M_{\Delta_{t-1}^{-1}(\{1\})}$ and $M_{\Delta_t^{-1}(\{1\})}$.

In practice, we separate the choice of the predecessor for M_{t-1} and for M_t. Such a separation is motivated by the fact that the description of data at time t depend on model M_{t-1} only by the transfer term in the case where the predecessor of M_t is M_{t-1}.

Consequently, and using the notations introduced previously, we can describe the learning algorithm in three steps:

1. Minimize the objective $\phi_2(M_t, M)$ over predecessor $M \in M_{\Delta_t^{-1}(\{1\})} \backslash \{M_{t-1}\}$ and M_t
2. Minimize the objective $\phi_1(M_{t-1}, M)$ over predecessor $M \in M_{\Delta_{t-1}^{-1}(\{1\})}$ and M_{t-1}
3. Minimize the objective $\phi_1(M_{t-1}, M) + \phi_2(M_t, M_{t-1})$ over predecessor $M \in M_{\Delta_{t-1}^{-1}(\{1\})}$, models M_{t-1} and M_t

In practice, this algorithm can have interesting properties in terms of comprehension of the underlying process. We propose to represent the dependency between two models by a vertex in a graph of models. Such a graphical representation makes the dependencies obvious and would enable a user interpret the decision, for example by detecting easily periodic behaviors.

10.3.3 Experimental Results

In order to illustrate the pertinence of our framework in practice, we have tested its performances on classical data sets with the naive prototype-based model. We considered three datasets: SEA Street and Kim, 2001 (50000 instances, 3 attributes, artificial), Weather Elwell and Polikar, 2011 (18159 instances, 8 attributes, real) and Electricity Market Harries, tr, and Wales, 1999 (45312 instances, 3 attributes, real).

For all datasets, we tested the instance-incremental version of our algorithm (by streaming directly over the data) and the batch-incremental version (by grouping successive data into a same batch). The experiments were all done with a fix sliding window size: $|\Delta_t^{-1}(\{1\})| = 3$. We tested the two proposed algorithms.

Table 10.1 presents the performances for different size S of batches. When $S = 1$, the situation corresponds to a problem of instance-incremental learning. Otherwise, the situation corresponds to batch-incremental learning.

As the purpose of this chapter is not to establish a competitive performance for the suggested algorithms with the prototype-based model, we do not propose any comparison to state of the art algorithms. The key idea is that the obtained results

are not necessarily better but similar to existing methods. Developing more accurate algorithms will be an improvement perspective to the proposed framework.

In practice, the calculation time with the passive approach makes impossible to use this algorithm directly for a real-time process: the optimization algorithms take too much time even in the case of instance-incremental and makes the system unable to deal with a real data stream.

TABLE 10.1: Error rate for several batch sizes

	$S = 1$	$S = 5$	$S = 10$	$S = 20$
SEA	0.32	0.37	0.37	0.34
Weather	0.32	0.28	0.36	0.36
Electricity	0.29	0.31	0.31	0.28

The obtained results are not competitive with state of the art algorithms, which was expected: Our method is not specifically designed to perform well and the class of models is very basic. We tested a very direct application of the equations presented above regardless of time complexity nor performance of the method. An effort has to be made in this direction in future works. The results are good enough to validate our framework yet: They are not bad for a highly general method and a simple class of models. However, we would like to point out that the poor results are also due to a property of our passive algorithm: at each time step, it tries to fit to one data point exactly. The only intuitive guarantee against overfitting is the model changing penalty $K(M_t|M_{\Delta_t^{-1}(\{1\})})$. A solution to overcome this weakness would be to consider that X_t is not only one instance but a sliding window of past instances. This idea was not tested in the scope of this thesis.

10.4 Conclusion

In this chapter, we have proposed a generic way to describe the problem of online learning, for data stream mining. The solution, which is inspired by our approach of transfer learning as well as Hidden Markov Models, is generic and describes various state of the art methods. One of the interests of our description is that it is very generic and offers a clean approach to data stream mining from a theoretical point of view. In particular, it could be used to propose a theoretical approach of the domain, which is, at this point, almost non-existent.

The following chapters present two applications of our methodology. The first application is based on ADWIN algorithm and concerns online topic modeling and online hybrid recommendation. The second application is a cognitive modeling of the phenomenon of U-shaped learning: It is a perfect example of the generic aspect of our framework.

Chapter 11

Incremental Topic Modeling and Hybrid Recommendation

In the previous chapter, we presented a generic model for incremental learning based on MDL principle. We have suggested that our framework is generic and can model a large variety of state of the art algorithms. Among these algorithms, we have shown that ADWIN (Bifet and Gavalda, 2007) actually corresponds to data compression and can be interpreted in the terms of our framework.

In this chapter, we propose an application of ADWIN in two different domains: online topic modeling and hybrid recommender systems. Topic modeling is an unsupervised learning task which consists in extracting prominent themes in text data. It is widely used in natural language processing, for instance for text classification. Recommendation is another example of an unsupervised task, which consists in associating some items to user based on their inferred tastes. Topic modeling and recommendation are two classical machine learning problems that have been little considered in online settings.

The remainder of this chapter is organized as follows. In a first section, we investigate the problem of online topic modeling. The algorithm we propose will be tested on artificial and real datasets. In Section 11.2, we propose an application of this algorithm for hybrid recommender systems based on textual data.

This chapter presents and develops the ideas contained in three papers (Murena, Al Ghossein, Abdessalem, and Cornuéjols, 2018; "Online Learning with Reoccurring Drifts: The Perspective of Case-Based Reasoning"; Al-Ghossein, Murena, Abdessalem, Barré, and Cornuéjols, 2018). They are the result of a collaboration with Marie Al-Ghossein and Talel Abdessalem (LTCI, Télécom ParisTech). Our main contribution is the general methodology, while the implementations and the use of stochastic matrix factorization have been proposed by co-authors.

11.1 Online Topic Modeling

In this section, we present an adaptation of ADWIN algorithm for online topic modeling in the presence of concept drift.

11.1.1 Topic Modeling

The abundance of text sources provided by online platforms and social networks offers new opportunities and introduces new challenges in the domain of text modeling. Two classes of methods have emerged, based either on n-gram language models (Chen and Goodman, 1996) or probabilistic topic modeling (Hofmann, 1999).

While the first class focuses on semantic modeling of languages based on the order of words, probabilistic topic modeling describes documents as an unordered bag of words drawn from mixtures of word distributions called topics. Even if the human interpretation of topics remains hard to achieve (Chang, Gerrish, Wang, Boyd-Graber, and Blei, 2009), these frameworks are used for a large variety of tasks ranging from text analysis (Phan, Nguyen, and Horiguchi, 2008; Rosen-Zvi, Griffiths, Steyvers, and Smyth, 2004), recommendation (Wang and Blei, 2011; Hu and Ester, 2013), sentiment analysis (Rao, Li, Mao, and Wenyin, 2014) to image annotation (Feng and Lapata, 2010). Among topic models, Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan, 2003) has gained more and more attention for its simplicity and its modularity. Several variants of the original model have been developed to achieve new tasks that cannot be performed with the original model (see for instance (Rosen-Zvi, Griffiths, Steyvers, and Smyth, 2004; Hu and Ester, 2013)).

Latent Dirichlet Allocation (Blei, Ng, and Jordan, 2003) is a probabilistic graphical model designed to provide a definition of documents based on latent features called *topics*. A topic corresponds to a word distribution and a document is modeled as a weighted mixture of topics. The generative process can be described as follows:

1. Choose $\theta \sim Dirichlet(\alpha)$
2. For each word W_n in document:
   (a) Choose a topic $z_n \sim Mult(\theta)$.
   (b) Choose a word w_n for the multinomial $p(w_n|z_n, \beta)$.

The corresponding generative model is given in figure 11.1.

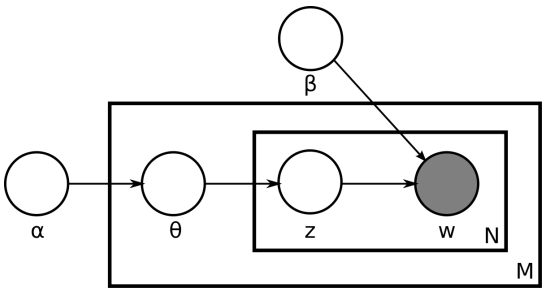


FIGURE 11.1: Generative model of Latent Dirichlet Allocation.

In this model, several parameters have a direct interpretation in terms of document analysis. First, the α parameter (hence the parameter of the Dirichlet distribution) influences the parameter of the multinomial topic distribution and corresponds to the mean value of a topic distribution θ inside a document. For instance, when $\alpha = (1, \ldots, 1)$, the topics are uniformly represented in documents. This parameter is important in document stream analysis since it depicts the topic trends. The parameter β is a word-topic distribution: the t-th column of matrix β is the vector of probabilities for a word to be drawn inside t-th topic.

LDA is trained either offline (Blei, Ng, and Jordan, 2003) or online (Hoffman, Bach, and Blei, 2010), following Maximum Likelihood Principle. The algorithms used for the optimization are usually based on variational inference or Gibbs sampling.

The base model of LDA infers topic distributions from a given batch of documents. This setting is not adapted to evolving environments, including text mining

on documents generated continuously at high rates or streams of documents. Nevertheless, solutions have been proposed to adapt LDA to temporal frameworks where the data distribution varies over time.

Dynamic Topic Models (DTM) (Blei and Lafferty, 2006) are an attempt to include a dynamic behavior into LDA. DTM models the *word-topic distribution*, i.e., the distribution of words inside a topic, as an evolving parameter. The distribution of this parameter at time t is defined with respect to its distribution at time $t-1$. A closely related idea is developed by SeqLDA (Du, Buntine, Jin, and Chen, 2012), but it is applied at the level of a book where the time parameter is associated to the index of the paragraph.

An alternative is offered by continuous-time models (Wang and McCallum, 2006) which assume that the distribution over topics is influenced by word co-occurrences (such as in standard LDA) and by the document date. The major disadvantage of this method is its offline nature: the model can only be learned once we have the whole corpus. It is thus inefficient in the context of stream mining. A frequent strategy for stream mining with LDA consists in grouping documents by time slices (see for instance (Blei and Lafferty, 2006; Griffiths and Steyvers, 2004)). On the other hand, online incremental LDA offers an interesting alternative since it does not require storing previous data and relies only on the new received documents (AlSumait, Barbará, and Domeniconi, 2008). The major problem of this method is the difficulty of defining time slices. In particular, modifications in topics might occur on a time period significantly smaller than the chosen time slice. This scale-dependency is taken into account by some continuous-time methods (Wang and McCallum, 2006; Iwata, Yamada, Sakurai, and Ueda, 2010).

Our approach takes a completely different direction. We propose to use change detection methods to estimate change of topics in document streams.

11.1.2 Adaptive windowing for Topic Drift Detection

11.1.2.1 Principle

The proposed method for topic change detection is based on the use of ADWIN combined with a training of LDA. We propose a framework in which documents arrive one by one in the form of a data stream. A document received at time step t is denoted by $\mathbf{w}_t$. Given (α, β), the vector parameters of the two Dirichlet distributions, the likelihood of the model, as shown by (Blei, Ng, and Jordan, 2003), is given by:

$$\mathcal{L}(\mathbf{w}) = \frac{\Gamma\left(\sum_i \alpha_i\right)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{i=1}^{k} \theta_i^{\alpha_i - 1}\right) \left(\prod_{n=1}^{N} \sum_{i=1}^{k} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j}\right) d\theta \tag{11.1}$$

where Γ is the gamma function and w_n^j measures the quantity of word j in document n. In this expression, k represents the number of topics, N the number of words in the document and V the size of the vocabulary.

A change in the stream of likelihood corresponds to a change in the data distribution and can be detected by ADWIN algorithm. The selected indexes by ADWIN correspond to the documents received after the drift.

The principle of our method relies on a couple of intuitive guarantees:

- The likelihood measures the generative quality of the model with regards to observed data. When a model is not adapted, the likelihood decreases.

- ADWIN is sensitive to changes in the mean value of a time series. Thus, it will detect a change in the likelihood caused by a change of the model.
- ADWIN will select large sub-windows to train a new LDA model. The drift will be predicted with a better accuracy for large window sizes, which is also optimal to train a LDA model.

Following this idea, our method can be described as follows. At time step t, the system has access to a LDA model M_t which describes the data. When the system gets a new document, we compute the likelihood of observing the document, adds it to the current window, and inspects it with ADWIN to check if a drift occurred. When a drift is detected, the current LDA model is trained on the documents selected by the kept sub-window.

11.1.2.2 Algorithm

The algorithm we present is a direct implementation of these ideas. It is based on the idea of separating the tasks of document modeling and topic drift detection by associating a different model for each task. The LDA model used for document modeling is denoted by LDA_m and the LDA model used for drift detection is denoted by LDA_d .

The LDA model used for document modeling, LDA_m , is updated with each received document and retrained when a drift is detected.

For each received document, our approach, called *Adaptive Window based Incremental LDA* (AWILDA), computes the associated likelihood of the model LDA_d and adds it to ADWIN. If a drift is detected, the model LDA_m is retrained on the sub-window selected by ADWIN. Besides, LDA_m is updated with each received document based on Online LDA algorithm. We note that to initialize the model, we train it on a relatively small chunk of documents before starting the detection.

Whereas the LDA model used for document modeling, LDA_m, is updated with each received document and retrained when a drift is detected, the LDA model used for topic drift detection, LDA_d, is retrained on the sub-window selected by ADWIN for each detected drift. It is not updated as more documents are received.

11.1.2.3 Theoretical guarantees

Since it is based on theoretically trusted algorithms, AWILDA presents interesting theoretical properties which guarantee the quality of its results regarding drift detection.

We introduce the same notations as presented in (Bifet and Gavalda, 2007). We consider a window W of length n which is divided into two sub-windows W_0 and W_1 of respective sizes n_0 and n_1. Let m be the harmonic mean of n_0 and n_1 (hence $\frac{1}{m} = \frac{1}{n_0} + \frac{1}{n_1}$). We suppose that, in ADWIN, the drift is detected for $|\hat{\mu}_{W_1} - \hat{\mu}_{W_0}| \geq \epsilon_{cut}$ (where $\hat{\mu}_{W_0}$ designates the mean value over sub-window W_0). Let δ be such that:

$$\epsilon_{cut} = \sqrt{\frac{1}{2m} \ln \frac{4n}{\delta}} \tag{11.2}$$

With these parameters, Theorem 3.1 in (Bifet and Gavalda, 2007) ensures both false positive rate bound and false negative rate bound. These results can be adapted to our setting.

Theorem 8. *At every time step, if documents are generated by a single LDA model in time period covered by W, the probability that AWILDA detects a drift at this step is at most δ.*

Proof. On the covered window, ADWIN gets a time series $X_t = \mathcal{L}(D_t)$ where D_t are equally distributed (for a single LDA model) and $\mathcal{L}$ represents the likelihood of LDA_d which is constant on W for AWILDA. Thus the mean of the variables remains constant on W. The conclusion follows from the properties of ADWIN. □

Following the same direction, the following theorem can be proven for false negative rate bound.

Theorem 9. *Suppose that, at a time step t, window W can be split in two parts W_0 and W_1 and documents are independent and identically distributed by a LDA distribution LDA_0 (resp. LDA_1) on sub-window W_0 (resp. W_1). If $|\mathbb{E}_{D\sim LDA_d}[p_{LDA_1}(D) - p_{LDA_0}(D)]| \geq 2\epsilon_{cut}$, then with probability $1-\delta$ AWILDA detects a drift inside sub-window W_1.*

Proof. The idea of the proof is the same. The mean value of $X_t = \mathcal{L}(D_t)$ on sub-window W_0 is:

$$\mu_t = \mathbb{E}_{D\sim LDA_0}[p_{LDA_d}(D)] = \mathbb{E}_{D\sim LDA_d}[p_{LDA_0}(D_t)]$$

An equivalent result can be found for W_1, and the theorem comes directly. □

Unlike for Theorem 8, a simple interpretation of Theorem 9 is not direct. For instance, two LDA models can be distinct and not share the targeted property. Finding conditions on the parameters of the three distributions is an interesting task that we will not address here. However, it has to be noticed here that the guarantee on the false negative rate depends on the choice of LDA_d .

11.1.2.4 Nature of the drift

In practice, concept drift can happen in different ways. A drift is called *abrupt* when it happens at a given time step at any amplitude. On the other hand, a drift is called *gradual* when small distribution variations are happening at each time step on a certain period of time.

The case of abrupt drift has been explicitly studied with the setting of Theorem 9. It corresponds to the case where the document distribution changes from one given state to another between sub-windows W_0 and W_1. Results given in (Bifet and Gavalda, 2007) show that the detection delay can be estimated by $O(\mu \ln(1/\delta)/\epsilon^2)$ where μ is the mean of the distribution before drift. In our case, this delay is of critical importance since it defines the size of the chunk for retraining the model. AWILDA faces a trade-off between predicting a drift as early as possible (in order to maximize the likelihood) and collecting as many data as possible to get a good estimator of the underlying LDA model.

The case of gradual drift is less adapted to the developed framework. Properties of ADWIN have been shown in the case of a linear gradual drift, but these results are difficult to translate directly into our setting where the time series tracked by ADWIN has a complex mathematical definition. Understanding the behavior of AWILDA in the case of gradual drift is a task that would come together with a proper study of Theorem 9.

In our experiments, we will consider abrupt drifts only. A related discussion will be proposed in the conclusion.

11.1.3 Experimental Results

In this Section, we present the experiments we conducted in order to prove the effectiveness of our approach. We show in particular how it performs when addressing the problems of topic drift detection and document modeling, using a set of synthetic and real datasets.

11.1.3.1 Datasets

Synthetic data. To demonstrate the ability of detecting drifts, we generate synthetic datasets where we artificially insert drifts at random moments throughout the sequence of documents. Synthetic datasets are denoted by Sd_r, where r is the number of simulated drifts. Documents observed between two consecutive drifts are generated by one LDA model following its generative process. At each occurring drift, we draw uniformly the hyperparameters α and β. For the generation of one dataset, the number of topics is fixed for all the models.

We present experiments performed on the following two synthetic datasets: Sd_4 and Sd_9, containing 4 and 9 drifts respectively. Handling document streams is a very common task in environments where short texts are generated and shared, e.g., newswires, tweets. Thus, we choose to generate documents containing 100 words, and we fix the vocabulary size to 10,000 words and the number of topics, k, to 15. Following the setting in (Blei, Ng, and Jordan, 2003), α and β are first set to $50/k$ and 0.1 respectively, and are then changed at each drift. In Sd_4, we generate exactly 2,000 documents from each distribution, separating two consecutive drifts by the same number of documents. In Sd_9, we vary the number of documents generated by each model between 500 and 1,000 documents.

Real data. We also conduct experiments on real-world data. We use three real data sources: *Reuters-21758*, consisting of newswire articles classified by categories and ordered by their date of issue, *ml-100k*, consisting of abstract of movies, and *plista*, consisting of a collection of news articles published in German on several news portal. In the procedure of data preprocessing, we removed stop words, words occurring once, down-cased and stemmed all remaining words.

The ApteMod version of dataset *Reuters-21758* [1] contains 12,902 documents classified in multiple categories (for a total of 90 categories). Since our approach is designed to detect topic drifts in document streams and to adapt the model accordingly, we reorder the newswire articles based on their categories. We artificially ensure an emergence of topics at specific points of the document stream and we try to provoke a drift in the topic distributions. We derive from the initial ordered dataset two sets of articles that we use in our experiments. In the first set, denoted by $Reuters_1$, we select the articles belonging to the category “acq" followed by the articles belonging to the category “earn". We expect the algorithm to detect the sudden change in topics mentioned in the documents. In the second set, denoted by $Reuters_4$, we select articles classified in a specific category and add them consecutively to the dataset. This is done for the five following categories: “interest", “trade", “crude", “grain", and “money-fx".

The *ml-100k* dataset corresponds to the MovieLens 100k dataset [2] and gathers 100,000 ratings from 1,000 users on 1,700 movies, spanning over 18 months. Movies

[1] http://archive.ics.uci.edu/ml/
[2] http://www.movielens.org

become available according to their reported release date, and we use DBpedia [3] to collect abstracts written in English and describing each one of them. In this section, we do not use the user ratings, but only the abstracts. The ratings will be used in Section 11.2.

The *plista* dataset is described in (Kille, Hopfgartner, Brodt, and Heintz, 2013) and captures interactions collected during the month of February 2016 on several German news portals. We remove from the dataset interactions corresponding to unknown users, users with less than three interactions, and items with no available textual description. Finally, the dataset gathers 32,706,307 interactions from 1,362,097 users on 8,318 news articles. The date of an article corresponds to its publication date. Such as for *ml-100k*, we ignore the user interactions for the moment.

11.1.3.2 Setting of AWILDA

As defined in equation 11.1, the likelihood of a LDA model is not computable. Thus, we relied on an upper-bound $\mathcal{L}'$ proposed in variational inference (see equation 1 in (Hoffman, Bach, and Blei, 2010)). In practice, the results observed with this upper-bound are not satisfying due to a lack of precision: the probabilities to observe data are very low and the method fails at discriminating them with enough accuracy. In order to overcome this difficulty, we considered the logarithm of $\mathcal{L}'$ (hence an upper-bound of log-likelihood). This quantity is theoretically unbounded, which is a problem for ADWIN, but in practice it is observed that the values vary only in a small interval (the width of which depends on the dataset). In our experiments, we prevented the quantity to decrease too much by fixing a minimal bound so that the quantity of interest becomes bounded. A reasonably low value for this threshold was never reached in the scope of the presented experiments. We do not have any way to evaluate an optimal value for this bound in a general case though.

11.1.3.3 Evaluation

Our evaluation concerns the tasks of topic drift detection and document modeling.
Topic drift detection. We evaluate the ability of detecting drifts by checking the latency between the moment when the real drift happens and the moment it is detected.
Document modeling. Given a LDA model trained on a set of documents, the goal in document modeling is to maximize the likelihood on unseen documents. For the evaluation, we use the measure of perplexity (Jelinek, Mercer, Bahl, and Baker, 1977), which is defined by:

$$perplexity(D_{test}) = 2^{-\frac{\sum_{d=1}^{M} \log_2 p(w_d)}{\sum_{d=1}^{M} N_d}} \tag{11.3}$$

Perplexity is the tool used by default in language modeling to measure the generalization capacity of a model on new data. Since we are considering document streams, the perplexity is computed for each received document using the current model.

The performance of AWILDA is compared to the online version of LDA (Hoffman, Bach, and Blei, 2010).

[3] http://www.dbpedia.org

We also compare AWILDA to three other variants. In these variants, the model LDA_m is updated in a similar way as for AWILDA, but the methods differ in the way the detection model LDA_d is updated:

- **AWILDA-2.** LDA_d is trained on a first small chunk of documents that is used to initialize all the models. It is not updated as more documents are received.
- **AWILDA-3.** LDA_d is updated for each received document and is equivalent to a classic online LDA model.
- **AWILDA-4.** LDA_d is updated for each received document using online LDA algorithm. When a drift is detected, the model is retrained on the sub-window selected by ADWIN.

Regarding the theoretical study, it can be easily verified that Theorem 8 and Theorem 9 hold true for AWILDA-2, but not for AWILDA-3 and AWILDA-4. In particular, it is noticeable that we do not have guarantees for the performances of AWILDA-3 and AWILDA-4 since the model LDA_d is updated at each step. *A priori*, there is no chance that the means remain constant when the likelihood function $\mathcal{L}$ varies.

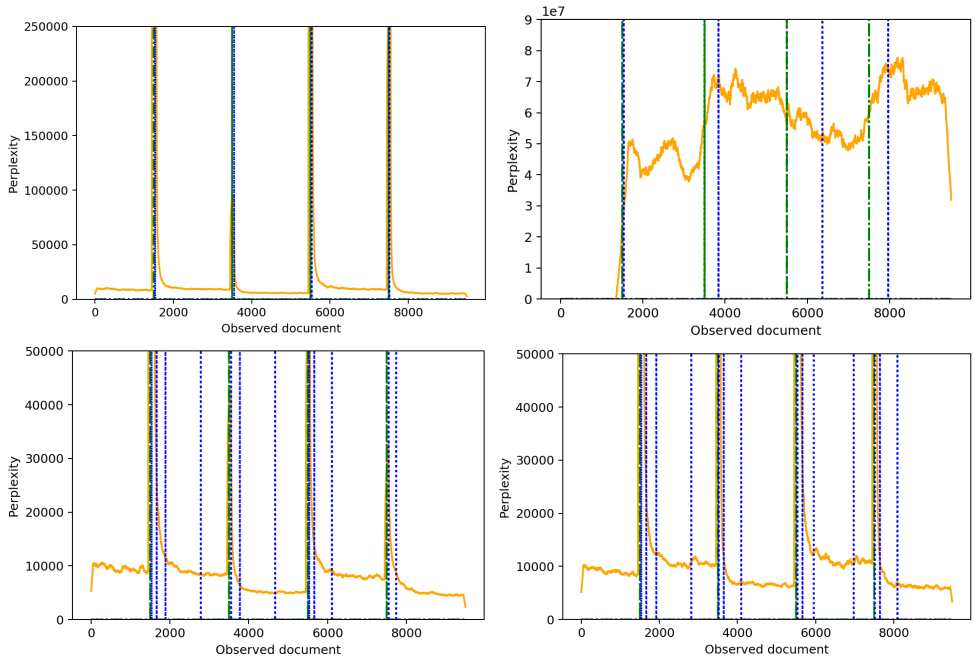


FIGURE 11.2: Topic drift detection on the Sd_4 dataset using AWILDA (first figure), AWILDA-2 (second figure), AWILDA-3 (third figure), and AWILDA-4 (fourth figure).

11.1.3.4 Comparison of AWILDA and its variants on Sd_4

In the first set of experiments, we compare the performance of AWILDA and its variants when performing the task of topic drift detection on the synthetic dataset Sd_4. The results are presented in Figure 11.2. The LDA model used to compute perplexity is learned and updated differently depending on the method considered. We represent the perplexity as a moving average with a sliding window of 100 observations. The exact occurrence of drifts is marked by a green dashed vertical line and the detection of drifts is marked by a blue dotted vertical line.

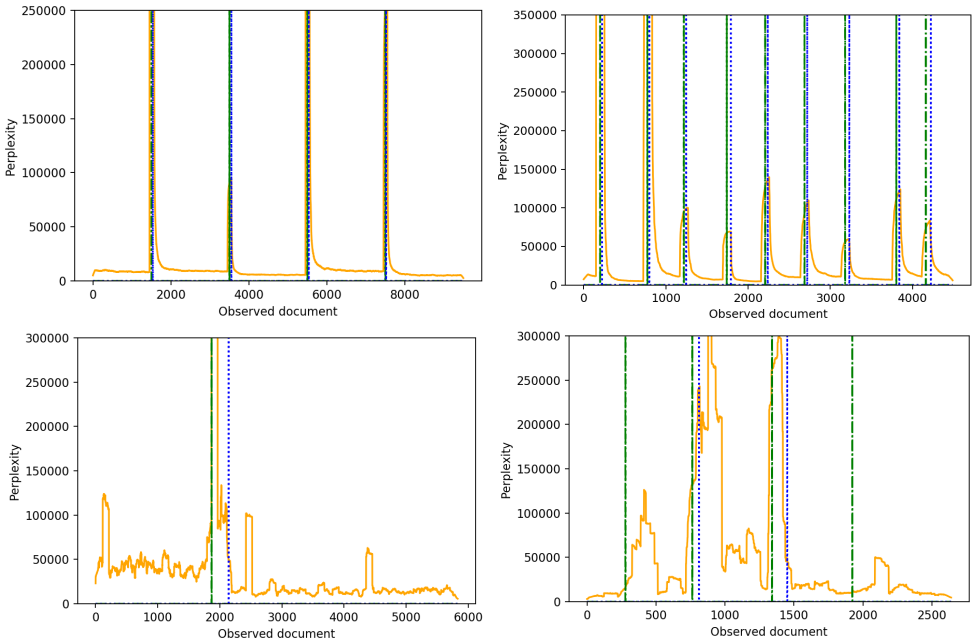


FIGURE 11.3: Topic drift detection using AWILDA and applied on the Sd_4 dataset (first figure), Sd_9 dataset (second figure), $Reuters_1$ dataset (third figure), and $Reuters_4$ dataset (fourth figure).

AWILDA and AWILDA-2 detect only true positive drifts, while AWILDA-3 and AWILDA-4 detect false and true positive drifts. AWILDA is also more reactive than AWILDA-2 and spots drifts faster. Updating the LDA_d model with each received document in AWILDA-3 and AWILDA-4 modifies the underlying distribution of topics, leading ADWIN to detect false positive drifts.

AWILDA performs best for all the studied datasets, and we present in the following the results for the other datasets.

11.1.3.5 Performance of AWILDA on controlled datasets

As shown in Figure 11.3, AWILDA is able to detect all the drifts occurring in the datasets Sd_4, Sd_9, and $Reuters_1$ after receiving only a few observations from the new distribution. Concerning the $Reuters_4$ dataset, our approach spots two drifts and misses the two others. We note that in this particular dataset, we switch from a topic to another relatively fast, i.e., around 500 documents per category. Topics in articles can also be interconnected which makes the task even more complicated.

11.1.3.6 Comparing AWILDA with online LDA

In the last set of experiments, we compare our approach with online LDA (Hoffman, Bach, and Blei, 2010). For this analysis, we only use real datasets: $Reuters_1$, *ml-100k* and *plista*.

For $Reuters_1$, we show in Figure 11.4 the evolution of perplexity throughout the set of documents before and after the drift occurs. The perplexity is computed using LDA_m of AWILDA.

Figure 11.5 shows the perplexity measured on the document streams of *ml-100k* and *plista* for AWILDA and online LDA (the results displayed here are obtained with a number of topics fixed to 10, knowing that similar patterns appear for other values of this parameter). The perplexity is represented as a moving average with a

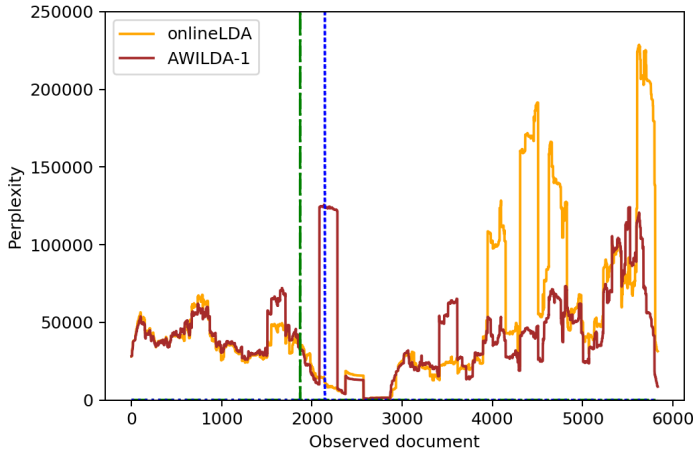


FIGURE 11.4: Comparison of online LDA and AWILDA for the task of document modeling with $Reuters_1$.

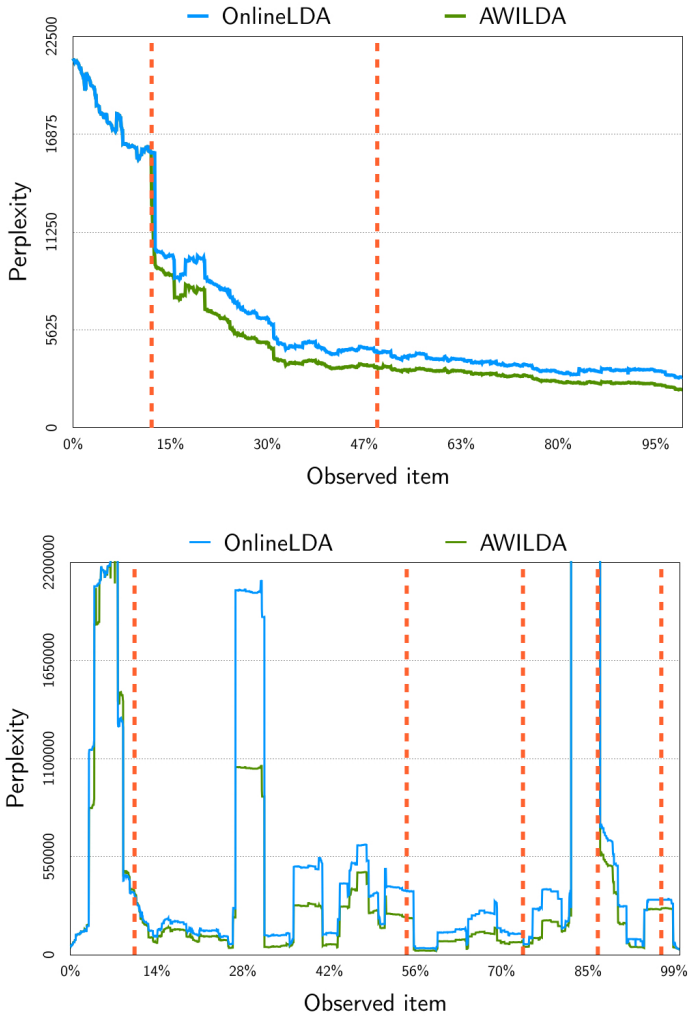


FIGURE 11.5: Comparison of online LDA and AWILDA for the task of document modeling on *ml-100k* (first subfigure) and *plista*, (second subfigure) using the measure of perplexity.

sliding window of 200 observations. The red dotted vertical line marks the detection of a drift by AWILDA. AWILDA detects two drifts for the held-out documents of *ml-100k* and five drifts for *plista*. This difference in behavior is expected knowing the volume and nature of both datasets (movies vs. news). A further analysis of the datasets with experts from both domains will help to establish the link between the detected drifts and real-life events occurring in the same time period, for better understanding and explainability.

Before detecting any drift, online LDA and AWILDA are trained in the same way and on the same data, which explains the close values of perplexity. After detecting the first drift, AWILDA outperforms online LDA for the task of document modeling. As documents continue to arrive, AWILDA is more adapted to the new data. Its drift detection component allows it to adjust to changes after each drift, resulting in a better performance. For the $Reuters_1$ dataset, where perplexity is not averaged, we observe a temporary increase in perplexity for AWILDA just after a drift occurs. This is due to the fact that AWILDA retrains its model on the relatively small sub-window selected by ADWIN and is not optimal then.

11.1.4 Discussion

We notice that the observed properties of the four variants of the algorithm are close to the predictions which were given by Theorems 8 and 9. In particular, it has been shown that AWILDA and AWILDA-2 would perform better than AWILDA-3 and AWILDA-4 with regards to false positives.

The superiority of AWILDA over the other variants raises interesting questions. It is noticeable that the best algorithm in terms of drift detection is also the only one which detection model is actively updated at each drift and not passively, at each step or for each observation. This property is particularly interesting: it means that the best algorithm in terms of drift detection is also the most efficient one in terms of computation time. However, in some examples, it might not be the optimal algorithm for the accuracy of the predicted model: there is no theoretical guarantee that the documents selected by ADWIN are a good representative set for the new distribution.

Moreover, the non-updating property of AWILDA is of particular interest: it illustrates the idea that good drift prediction does not require to have good modeling properties, which may be counter-intuitive in a way. The extreme case, AWILDA-2, shows rather good performance as well whereas the detection model is never updated, which means that it does not encode any information relative to the underlying distribution. A random LDA model could also work for this task. Having a completely unrelated detection model might produce false-negative errors though: if the detection model is too different from the actual model, there is a chance that the likelihood change, when the underlying model varies, is not important enough to be detected.

11.2 Incremental Hybrid Recommendation

In this section, we present how AWILDA can be used in incremental recommender systems. The addressed task consists in recommending items to users in a setting where items, users and ratings can appear in a stream fashion, and where the global item generation varies over time.

11.2.1 Online Hybrid Recommendation

Due to the abundance of available choices in online platforms and services, recommender systems (RSs) have been playing an essential role to help users and empower companies. Approaches to recommendation can mainly be categorized into three classes. First, *content-based* (CB) approaches rely on information extracted from user profiles and item descriptions. Second, *collaborative filtering* (CF) approaches make use of user activities and past interactions (e.g. ratings and clicks) to learn preferences and generate recommendations. Lastly, *hybrid approaches* aim to combine both techniques in order to overcome their weaknesses: While CB methods tend to be overspecialized and lack a sense of novelty, the performance of CF methods drops with an increase of rating sparsity and in the cold-start setting.

To get the best of both worlds, *hybrid* approaches allow CF approaches to exploit auxiliary information like text (as in (Wang and Blei, 2011; Wang, Wang, and Yeung, 2015)) and images (He and McAuley, 2016). Collaborative Topic Regression (CTR) (Wang and Blei, 2011) is a popular hybrid approach combining probabilistic topic modeling for content analysis and latent factor models for CF (Pan et al., 2008).

Hybrid RSs are particularly useful to cope with the problem of *cold-starts* (Schein, Popescul, Ungar, and Pennock, 2002) which occurs when new users or new items are introduced into the system. While recommending new and fresh items is essential, CF methods have difficulties in doing so, as no or few feedback related to these items is observed. Hybrid RS are able to recommend new items by leveraging auxiliary information. They also help to alleviate the sparseness of rating or feedback data, thus improving the quality of recommendation. To this end, previous work has utilized text data such as abstracts (Wang and Blei, 2011), synopses (Wang, Wang, and Yeung, 2015), or reviews (Bao, Fang, and Zhang, 2014). Several techniques have been used to model documents like LDA (Wang and Blei, 2011), stacked denoising autoencoders (Wang, Wang, and Yeung, 2015) or convolutional neural networks (Kim, Park, Oh, Lee, and Yu, 2016). Images have also been leveraged in this context and visual appearances of items can be added to the preference model (He and McAuley, 2016).

Most hybrid RS proposed in the literature are meant to work in batch, where an initial model is first built from a static dataset and then rebuilt periodically as new chunks of data arrive. In real-world applications, the recommendation problem can be formulated as a data stream problem where RS are designed to learn from continuous data streams and adapt to changes in real-time. Recently, (Frigó, Pálovics, Kelen, Kocsis, and Benczúr, 2017) has shown that simple online algorithms can generate better recommendations than more complex ones that are only updated periodically. Online RS are mainly based on incremental learning to continuously update models when receiving new observations. Incremental CF approaches have been proposed in this direction, like incremental neighborhood-based methods (Miranda and Jorge, 2009) and incremental matrix factorization (using stochastic gradient descent (Vinagre, Jorge, and Gama, 2014b) or alternating least squares (He, Zhang, Kan, and Chua, 2016)).

Learning from data streams should also account for concept drifts which occur when the definition of modeled concepts changes over time. User preferences and item descriptions are expected to change in different ways, at different moments and at different rates (Ding and Li, 2005; Koren, 2010). Incremental learning is a

way of passively adapting to current changes in the data distribution, by continuously learning from new data. Actively accounting for changes of user preferences has been based on the intuition that users' recent observations are more relevant than older ones. Sliding window techniques have been explored in this direction (Nasraoui, Cerwinske, Rojas, and Gonzalez, 2007; Siddiqui, Tiakas, Symeonidis, Spiliopoulou, and Manolopoulos, 2014; Matuszyk, Vinagre, Spiliopoulou, Jorge, and Gama, 2015). We note that these techniques make assumptions concerning the relevance of old observations and the rate at which all preferences drift, which are not always accurate.

11.2.2 From Incremental Matrix Factorization to Adaptive Collaborative Topic Modeling

Matrix factorization (MF) is a popular collaborative filtering technique that is used to model users' interactions by representing users and items in a space of latent factors learned from the data. If R designates the matrix of interactions (where $R_{ui} = 1$ if the user u interacted with the item i, and 0 otherwise), then MF aims to approximate R as a product of two matrices P and Q by minimizing over P and Q:

$$\sum_{(u,i)\in D} \left(R_{ui} - P_u Q_i^T\right)^2 + \lambda_u \|P_u\|^2 + \lambda_i \|Q_i\|^2 \tag{11.4}$$

where D is the set of observed interactions, and λ_u and λ_i are regularization parameters. The score of an item i for a user u, denoted by $\hat{R}_{ui}$, is computed using the scalar product between P_u and Q_i^T. Items are ordered by descending proximity of $\hat{R}_{ui}$ to value 1, and top-N items are recommended for u.

Classic algorithms for MF are not suitable for a data stream setting. A variant of MF adapted to the incremental nature of data streams (Vinagre, Jorge, and Gama, 2014b) suggests the following procedure. Observations $\langle u, i\rangle$ are received one after the other and handled by the algorithm. For each received observation, P and Q are updated using the gradient of the objective for this observation only (which corresponds to an estimator of the gradient on the whole data set). When either a user or an item are observed for the first time, they are added to the matrices with a random initialization, and the values of P and Q are then updated using the observation.

In our setting, observations are supposed to arrive in real-time and are mainly of two types. First, *interactions*, denoted by $\langle u, i\rangle$, designate positive actions (clicks, ratings) performed by users and concerning a certain item. Second, *additions of items*, denoted by $\langle i, doc_i\rangle$, usually occur when a new item becomes available at a certain

time step, and we consider cases where a textual description of the new item is provided.

Algorithm 4: Overview of CoAWILDA

Data: Set of observations O, Number of latent factors K, Learning rate η, Regularization parameters λ_u and λ_i
Result: P, Q
for o *in* O **do**
  if $o = \langle i, doc_i \rangle$ *(new item added)* **then**
    $\theta_i \leftarrow AWILDA(doc_i)$;
    $\epsilon_i \sim \mathcal{N}(0, \lambda_i^{-1} I_K)$;
    $Q_i \leftarrow \theta_i + \epsilon_i$;
  if $o = \langle u, i \rangle$ *(interaction received)* **then**
    if $u \notin Rows(P)$ *(new user observed)* **then**
      $P_u \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$;
    $e_{ui} \leftarrow 1 - P_u.Q_i^T$;
    $P_u \leftarrow P_u + \eta(e_{ui} Q_i - \lambda_u P_u)$;
    $\epsilon_i \leftarrow \epsilon_i + \eta(e_{ui} P_u - \lambda_i \epsilon_i)$;
    $Q_i \leftarrow \theta_i + \epsilon_i$;

CoAWILDA is presented in Algorithm 4. When a new item is received, we use AWILDA to model the descriptive document and extract topic proportions θ_i. The item latent vector Q_i representing an item i results of the addition of the topic proportions θ_i and an item latent offset ϵ_i. When a new interaction $\langle u, i \rangle$ is observed, we update the user latent factor P_u and the item latent offset ϵ_i, following the procedure of incremental MF. Recommendation is performed as described previously, where $\hat{R}_{ui} = P_u.Q_i^T = P_u.(\theta_i + \epsilon_i)^T$.

11.2.3 Experimental Results

In this Section, we discuss how our approach performs when addressing the problem of online recommendation, using real-world datasets.

11.2.3.1 Datasets

For the experimental evaluation of CoAWILDA, we used two of the datasets presented in Section 11.1.3.1: *ml-100k* and *plista*.

We remind that *ml-100k* gathers 100,000 ratings from 1,000 users on 1,700 movies, spanning over 18 months. Since we are addressing the problem of recommendation with implicit feedback (positive-only data), our goal is to recommend the movies the user is going to rate. We note that the dates reported in *ml-100k* correspond to the rating date of the movies and not to the actual watching date. Knowing that we are not concerned with the problem of evolution of user preferences in this work, we use *ml-100k* to evaluate our approach. As a textual description of movies, we use the abstract automatically extracted from DBpedia.

The *plista* dataset contains a collection of news articles published in German on several news portals, as weel as interactions collected during the month of February 2016. The reduced dataset, after pre-processing, gathers 32,706,307 interactions from 1,362,097 users on 8,318 news articles.

11.2.3.2 Evaluation protocol

RSs are traditionally evaluated using holdout methods. These methods are not adapted to the online setting (Vinagre, Jorge, and Gama, 2014a) mainly because when we randomly sample data for training and testing, we loose the temporal dimension and do not respect the original order of observations.

Since the topic model requires an initial phase of training, we adopt the evaluation process introduced in (Matuszyk and Spiliopoulou, 2014). We sort the dataset chronologically and then split it in the following three subsets:

- **Batch Train** subset. The first 20% of the dataset are used for the initial training of the models.
- **Batch Test - Stream Train**. The next 30% of the dataset are used for the validation of the initialized models, and for incremental online learning to ensure the transition between the first and the last phase.
- **Stream Test and Train**. The last 50% of the dataset are used for prequential evaluation, which is a test-then-learn procedure performed while iterating over the observations (Gama, Sebastião, and Rodrigues, 2009). Each observation $\langle u, i \rangle$ is used to evaluate the model by generating recommendations for user u, and then to update the model using $\langle u, i \rangle$.

We use recall@N and DCG@N to measure the quality of recommendation. These metrics are described in (Frigó, Pálovics, Kelen, Kocsis, and Benczúr, 2017) for the online setting. We report the results for the *Stream Test and Train* subset.

11.2.3.3 Compared Methods

Since previous work has demonstrated the advantages of using online recommendation compared to batch recommendation (Vinagre, Jorge, and Gama, 2014b; Frigó, Pálovics, Kelen, Kocsis, and Benczúr, 2017), we focus on incremental methods. We also only consider one approach for incremental MF, knowing that our method CoAWILDA can integrate any other algorithm for incremental MF or any model-based method. We compare the performances of several incremental methods adapted to the online setting, including variants of the one we propose.

- **CoAWILDA** is the method we propose, combining ADaptive Window based Incremental LDA (AWILDA) for topic modeling and incremental MF for CF. For *ml-100k*, we set the number of topics $K = 20$, $\eta = 0.04$, $\lambda_u = 0.01$, and $\lambda_i = 0.1$. For *plista*, we set $K = 10$, $\eta = 0.042$, $\lambda_u = 0.01$, and $\lambda_i = 0.1$.
- **CoLDA** relies on classical online LDA (Hoffman, Bach, and Blei, 2010) for topic modeling and incremental MF for CF. It replaces AWILDA from CoAWILDA with classical online LDA. For *ml-100k*, we set $K = 20$, $\eta = 0.05$, $\lambda_u = 0.01$, and $\lambda_i = 0.1$. For *plista*, we set $K = 10$, $\eta = 0.045$, $\lambda_u = 0.01$, and $\lambda_i = 0.1$.
- **AWILDA** denotes the method we propose for adaptive topic modeling. We try to use it for recommendation without the collaborative component, by representing users in the space of topics and updating their profiles as we get more observations. For *ml-100k*, we set $K = 20$, $\eta = 0.04$, and $\lambda_u = 0.01$. For *plista*, we set $K = 10$, $\eta = 0.042$, and $\lambda_u = 0.01$.

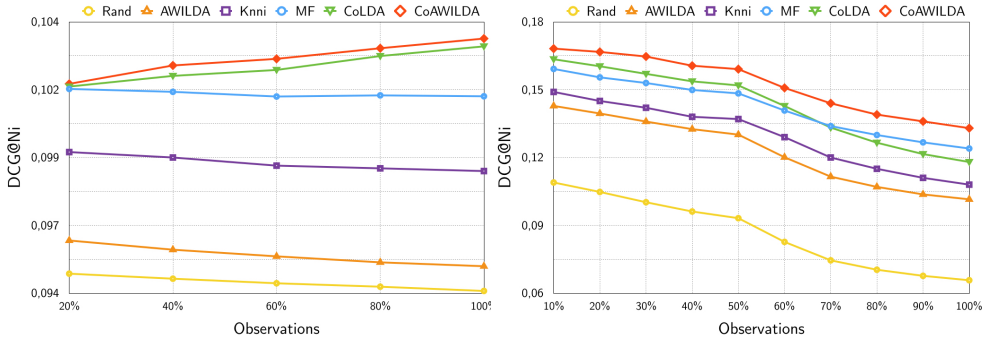


FIGURE 11.6: DCG@Ni of our approach CoAWILDA and other variants and incremental methods for *ml-100k* (first subfigure) and *plista* (second subfigure), where Ni is the number of available items. The evolution of DCG@Ni with the number of evaluated observations is reported.

- **MF** is the incremental MF (Vinagre, Jorge, and Gama, 2014b). Compared to CoAWILDA and CoLDA, MF does not leverage content information about items. For *ml-100k*, we set $K = 50$, $\eta = 0.01$, $\lambda_u = 0.02$, and $\lambda_i = 0.02$. For *plista*, we set $K = 50$, $\eta = 0.008$, $\lambda_u = 0.01$, and $\lambda_i = 0.01$.
- **Knni** is the incremental item-based approach proposed in (Miranda and Jorge, 2009). We set the number of neighbors to 300.
- **Rand** randomly selects items for recommendation.

For each of the methods, we performed a grid search over the parameter space of the methods in order to find the parameters that give the best performance (parameters reported above). We report the performance corresponding to the parameters leading to the best results.

11.2.3.4 Results and Discussion

Figure 11.6 shows the DCG@Ni of the methods we compare for *ml-100k* and *plista*, where Ni is the total number of items included in each dataset. The idea is to evaluate how each approach performs when ranking the items for each user. We report the metric value with respect to the number of observations processed, in order to analyze its evolution over the time spanned by the *Stream Test and Train* set.

CoAWILDA outperforms all the other methods evaluated for both datasets. The comparison between CoAWILDA and CoLDA demonstrates the effectiveness of the AWILDA algorithm for modeling document streams describing new items, and for improving the quality of item modeling and thus recommendation. CoLDA is not able to adjust to drifts occurring in topic modeling which deteriorates the recommendation quality over time.

The performance of CoLDA for *plista* can be divided in two phases. In the first one, the topic model is still able to carry out good document modeling and is beneficial for the recommendation: CoLDA performs better in terms of item ranking than MF which does not account for content analysis. In the second phase, and with the incapacity of online LDA to adjust to drifts, MF outperforms CoLDA. This means that not only the topic model is not adapted to newly received data, but it is also badly affecting the recommendation quality and there is no interest in using it anymore. We also note the importance of evaluating the evolution of the models over

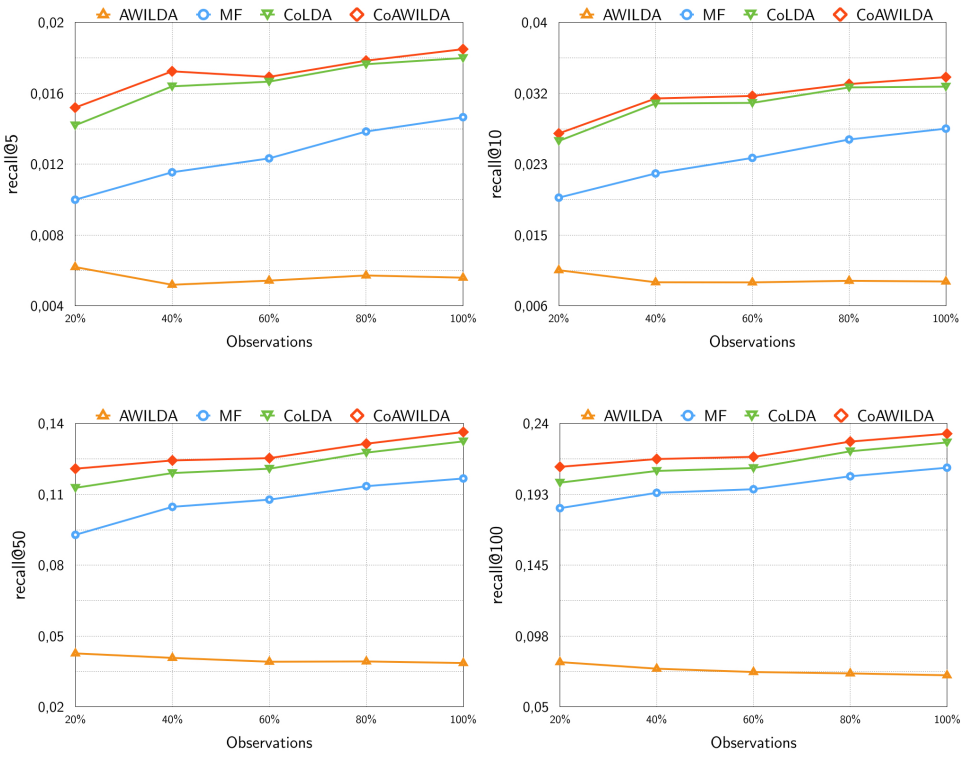


FIGURE 11.7: Recall@5, recall@10, recall@50, and recall@100 of our approach CoAWILDA and its variants on *ml-100k*.

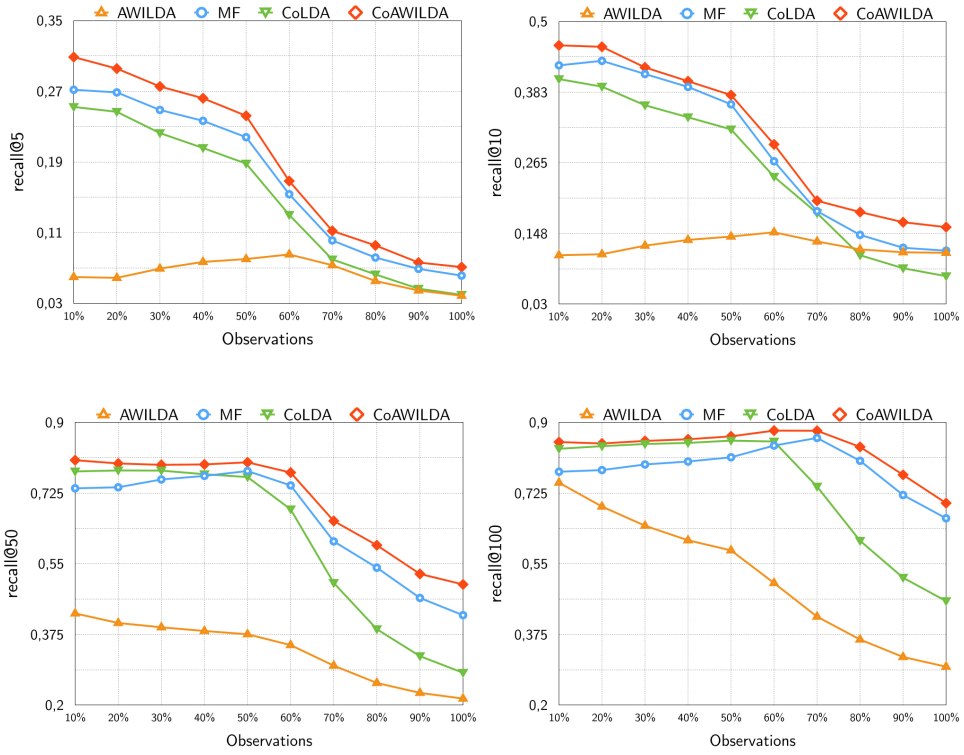


FIGURE 11.8: Recall@5, recall@10, recall@50, and recall@100 of our approach CoAWILDA and its variants on *plista*.

time to show how they are affected by eventual changes occurring in the data. This phenomenon appears for *plista* where more frequent drifts occur over time, mainly due to the nature of news data. Concerning *ml-100k*, CoLDA performs better than MF but still worse than CoAWILDA.

AWILDA is a content-based method and only relies on topics extracted from items to model user preferences. It performs poorly compared to the other methods, and proves the importance of having a CF component. Knni performs better than AWILDA, but is not as robust as MF and the hybrid approaches evaluated.

The number of available items grows significantly over time in *plista*. This results in the dropping of performance (in terms of ranking) of all methods over time. This is not the case in *ml-100k*, since only few movies are added in the corresponding time period. More data is received and more learning is done over time, which can explain the improvements in the performance of CoAWILDA and CoLDA.

Figures 11.7 and 11.8 show the recall@5, recall@10, recall@50, and recall@100 of our approach CoAWILDA and its variants on *ml-100k* and *plista* respectively.

The experiments for *ml-100k* confirm the ideas we mentioned before. CoAWILDA outperforms the other variants and performs better than CoLDA which relies on online LDA and does not adapt to changes in the data. CoLDA performs better than MF demonstrating the benefits of using content information. AWILDA relies only on content information which is a weak approach to model user preferences when used alone.

The experiments for *plista* highlight an interesting behavior. For recall@5 and recall@10, MF performs better than CoLDA for all considered observations. For recall@50 and recall@100, we observe two different behaviors where, first, CoLDA performs better than MF, and then MF outperforms CoLDA. We recall that the reported results are measured on the second half of the dataset (*Stream Test and Train* subset). Drifts may have occurred during the training phase, which is typically the case for *plista*. When measuring the recall@N, CoLDA is already weakened by the drifts that have happened and that were not taken into account. This leads to a point where the information learned by the topic model hurts the quality of recommendation, and MF starts performing better than CoLDA. This change of behavior occurs at different moments, depending on the recall we are measuring. For a higher N (e.g., recall@50, recall@100), the performance of CoLDA remains above the performance of MF for a longer time than for a lower N (e.g., recall@5, recall@10). Top list recommendation is thus more affected by the deterioration of the topic model.

All experiments demonstrate the effectiveness of using CoAWILDA, the strength of which relies in adapting to changes occurring in the data. Methods that do not detect and adapt to these changes, i.e., CoLDA, perform worse than CoAWILDA.

11.3 Perspective: Coping with Reoccurring Drifts

In this section, we discuss the possibility to handle reoccurring drifts in AWILDA, ie. concept drifts that come back to previously encountered models. We propose an interpretation of reoccurring drifts in the terminology of Case-Based Reasoning (CBR).

11.3.1 Reoccurring Drifts

Reoccurring drifts are particularly frequent and model the reoccurrence of previously encountered states (either cyclic or episodic). The question of reoccurring

drifts is essential in applications where seasonal effects can be observed or where the environment can oscillate between several states. In streams of documents, which is the focus of this chapter, many factors can generate reoccurring drifts. In news, the articles can be affected by the recurrence of some global contexts (for instance electoral context) that might affect the whole dataset. In reviews, seasonality effect is particularly important too.

Dealing with reoccurring drifts requires a bit more adaptation and especially the use of a memory to evaluate the relatedness of the current observation with the past. Various algorithms have been designed to tackle this issue.

The first method that was explicitly designed for reoccurring drifts (working with categorical attributes) is FLORA3 (Widmer and Kubat, 1996), an evolution of the original window-based FLORA method. When a drift is detected, FLORA3 inspects a pool of saved models instead of relearning a brand new model from scratch. The reuse procedure can be decomposed into three steps: Finding the optimal model (i.e., the model which makes the best predictions on the current data), update the chosen model (in order to make it consistent with current state), and comparing the updated version of the model to its memorized version. As an alternative to FLORA3, SPLICE-2 (Harries, Sammut, and Horn, 1998) offers another adaptation to recurring concept drifts on categorical features. The algorithm considers batches of data on which the concept is supposed to be stable. These batches are then clustered together, based on a notion of context similarity. In (Yang, Wu, and Zhu, 2006), the past history is modeled by a Markov chain and the future state is predicted according to the computed transition matrix.

Ensemble approaches are ideal for recurring drifts. For instance, Ensemble Building (EB) (Ramamurthy and Bhatnagar, 2007) aims to combine multiple classifiers with weights depending on their scores. If none of the known classifiers have good prediction rate on the currently observed chunk, a new classifier is trained and added to the pool. In a slightly different way, (Gama and Kosina, 2009) chooses current models from a pool of previously learned model. The models are stored in memory, as well as their associated referee. In (Jaber, Cornuéjols, and Tarroux, 2013), the traces of past relevant concepts are stored in the pool of base-learners. These base-learners are learned each time that no existing classifier is a good predictor on the current window of examples. A diversity criterion on the pool of base-learners guarantees that the pool is both diverse and not cluttered.

The approach of (Katakis, Tsouma-kas, and Vlahavas, 2010) is very similar but exploits an idea that is close to CBR: Batch examples are selected by the algorithm and transformed to *conceptual vectors*. These vectors are then clustered together and a new classifier is learned for each cluster. Finally, the more generic algorithm Learn^{++}.NSE (Elwell and Polikar, 2011) is also perfectly tuned for recurring drifts: The algorithm is based on a passive incremental approach and proposes a weighted majority vote on a pool of classifiers.

The use of memory, which is at play with reoccurring drifts, is highly similar to the problems encountered in the domain of Case-based Reasoning (CBR). In particular, the four steps of CBR are observed in memory management for stream mining. Retrieval is implied in the process of detecting similar states in the past (*did the drift lead to a previously encountered distribution?*); Reuse brings a solution to the current case based on the retrieved cases; Revision exploits information of the new case to adapt current cases; Retention evaluates if the new case has to be kept in memory (De Mantaras et al., 2005).

Interestingly enough, the similarities between the main questions of CBR and online learning have not been exploited much. Apart from the ensemble techniques mentioned above (in particular (Katakis, Tsouma-kas, and Vlahavas, 2010)) which are implicitly related to CBR, some methods use CBR in an explicit way. In (Salganicoff, 1997), all new observations are directly stored in memory but, depending on their relevance to the context, they can be deactivated or reactivated. It is shown that this strategy improves the robustness of lazy learning algorithms to concept drift. CBR is used in the context of spam classification with concept drift (Delany, Cunningham, Tsymbal, and Coyle, 2005): The case base is filled with a vector representation of emails and managed using a Case Base Editing strategy (Delany and Cunningham, 2004) which removes both noisy and redundant cases. This case base editing strategy is also used by (Lu, Lu, Zhang, and De Mantaras, 2016). The problem of instance-based learning has also been expressed in the context of data streams (Beringer and Hüllermeier, 2007): The proposed method updates the case base at each detection of a drift, implying the removal of a large number of cases.

11.3.2 Drift Adaptation seen as a CBR Problem

In this section, we present an interpretation of online learning in terms of case-based reasoning. The presented notions are given at an abstract level. An application to AWILDA will be proposed next.

11.3.2.1 General Process

In a context of stream mining, it is not possible to have a full CBR process at each step. The methodology we propose allies the performance qualities of active methods for stream mining and the use of memory, which is typical of CBR.

The data stream is analyzed by a drift detection algorithm (for instance ADWIN (Bifet and Gavalda, 2007)) on the base of a *score*. The purpose of this algorithm is to detect when the data distribution changed and when an adaptation is needed. Since a drift is necessarily detected with some delay, a drift detection comes with a batch of instances $\mathcal{D}$ generated by the new distribution. The score is computed based on a representation model of the data. It can correspond to the error rate of the model or to its likelihood for instance. In the following, we will denote by $score(\mathcal{D}, \mathcal{M})$ the score of data $\mathcal{D}$ relative to the model $\mathcal{M}$.

Instead of relearning the model from the batch selected by ADWIN, we propose to select the model from a case base and to adapt it in order to fit the new data. This use of case base is ideal for dealing with recurring concept drifts, as suggested by the state of the art.

11.3.2.2 Case Representation

One of the central questions of CBR concerns the management of the case base and the representation of cases. In the context of online learning, we propose the following storage process. A case corresponds to a data point, after or before any transformation process. As suggested by (Katakis, Tsouma-kas, and Vlahavas, 2010), the points are then grouped into clusters corresponding to concepts. Each of the clusters is associated to a unique decision model which can be either discriminative (e.g., a classifier in supervised setting) or generative (e.g., a probability distribution in unsupervised setting).

In a perspective of reusing previously solved cases to address new questions, this representation consists of a factorized representation of problems: the solution (here the decision model) is shared by several cases.

11.3.2.3 Case Retrieval

When a drift is detected, the first question is how to associate the batch of points to a corresponding group of cases. Using the representation we proposed, the relatedness of a batch to any case inside a cluster can be measured by its relatedness to its associated model. As a good candidate for this measure, we propose to use the score function.

The optimal cluster of cases is chosen to be the cluster such that the associated model maximizes $score(\mathcal{D}, \mathcal{M})$. Note that, especially for the first drifts, none of the learned models might describe well the observed data. In order to discard incorrect models, a threshold can be given for the score, under which no cases are selected. Such a threshold might be given by a complexity measure. In the scope of this thesis, we will ignore this problem.

11.3.2.4 Case Reuse

The retrieved cases do not necessarily correspond exactly to the current distribution of data. In order to cope with this problem, the decision model in use is retrained on a specific batch of data. This batch contains the points in the case cluster and the points in batch $\mathcal{D}$. This reused model thus incorporates both knowledge from the past and from current data. The model is taken as the reference model for the next observations, until a new drift is detected.

11.3.2.5 Case Revision

In the time interval between two drifts, we propose a case revision based on two aspects. On the one hand, the description model is updated online for each new observation, using a stochastic optimization scheme (Bottou, 2010). On the other hand, the most relevant data instances are kept in a short-term memory, in order to feed the case in the retainment phase. The relevance of an instance is evaluated with the score function, for the current model. These two actions are complementary: the model update is important in order to keep the decision model up-to-date, while the data selection contributes to an optimal case design.

11.3.2.6 Case Retainment

When a drift is detected, the model has to be saved in the case base. Two possibilities appear: either to re-write the selected case or to create a new case. This decision is motivated by the impact of creating a new model onto the global case base. If $(\mathcal{M}^{old}, \mathcal{D}^{old})$ designates the previous model and the cases associated to it, and $(\mathcal{M}^{new}, \mathcal{D}^{new})$ designates the current model and the data stored in short-term memory, one possibility to discriminate the two options is to compare $score(\mathcal{D}^{old}, \mathcal{M}^{new})$ and $score(\mathcal{D}^{old}, \mathcal{M}^{old})$. If the first score is higher, the new model is better at describing data from previous case model and thus the model has to be overridden. Otherwise, the previous model was satisfactory and the new model is relevant only for the new cases. Thus a new model has to be created and is associated to the instances in short-term memory.

In the case where the previous model is overridden, the cases stored in short-term memory are added to the case cluster of the model. In simple applications, where the number of cases per cluster is limited, only the cases with higher score are kept.

11.3.3 Application to AWILDA

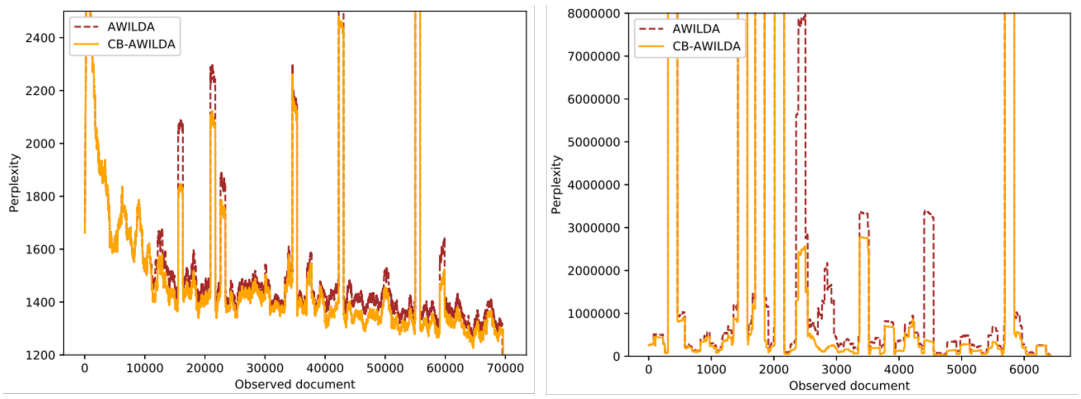
Textual content written by individuals and shared online on several platforms (e.g., tweets, news, reviews) is usually affected by their specific context that is in turn influenced by real-life events. It is essential to account for changes happening in the distribution of topics and words in order to improve document modeling. While AWILDA retrains a model at each detected drift, it cannot leverage previous learned information about a concept when it reappears due to its possible recurrence. We propose to store learned models that are no longer adapted to the current context and reuse them later when they are valid again.

In terms of the methodology described above, this problem can be described as follows. Each point corresponds to a document (described as a bag-of-words) and documents arrive sequentially as a stream. The task we address here is a modeling task: The purpose is to identify a good model that fits the data in real time. As a consequence, the model used to select the cases to cluster corresponds to the LDA model itself. The score function that we use is the log-likelihood, which measures the probability of observed documents to be generated by the model.

In order to demonstrate our approach, we present the experiments we conducted on two datasets from different domains. The first dataset gathers hotel reviews posted on TripAdvisor (Ganesan and Zhai, 2012) and is denoted by *trip*. The dataset comprises approximatively 200k reviews published from October 2001 to November 2009 and related to hotels located in ten different cities. We expect to observe a recurrence of concepts in this type of dataset due to the seasonality effect that influences the behavior of tourists and the hotel aspects they attach importance to. The second dataset is the *plista dataset* (also denoted by *news*).

We compare our approach, denoted by CB-AWILDA, to AWILDA. AWILDA is better suited to handle abrupt drifts: The model is retrained for each detected drift using the documents corresponding to the new distribution. AWILDA and CB-AWILDA are considered to be receiving a stream of document in real-time and to process documents sequentially. We use the first 20% of the document stream to initialize the models and we measure perplexity for all the documents received afterwards. We report the results obtained by fixing the number of topics to 5, and the minimum number of cases to 2.

Figure 11.9 shows the perplexity measured on the document streams of *trip* and *news* for AWILDA and CB-AWILDA. The performance of both methods at the beginning of the process is relatively similar. This is expected since the learning process is the same before any drift is detected. As more documents are received, CB-AWILDA outperforms AWILDA for the task of document modeling. For each detected drift, AWILDA is retrained using the documents related to the new distribution. This is thus pushing the model to forget previously learned information that may be valid in the future. On the other hand, CB-AWILDA leverages previously seen documents that correspond to the current distribution and uses them in the learning process. CB-AWILDA is therefore more adapted to the documents that are currently being received, which results in a better performance in terms of perplexity.



trip dataset *news* dataset

FIGURE 11.9: Evaluation of AWILDA and CB-AWILDA for the task of document stream modeling on the *trip* (first figure) and the *news* (second figure) datasets.

11.4 Conclusion

In this chapter, we proposed an analysis of the problem of online topic modeling. The proposed approach is very simple, since it only combines the modeling strength of a generative model (LDA) to an efficient drift detection method (ADWIN), but it proves its efficiency on both artificial and real datasets. We also put our algorithm in the perspective of a different domain of application: recommender systems. We have illustrated the performances of an online hybrid recommender system that updates its item representations based on drift detection and we have shown that drift detection helps keeping an up-to-date model of the observations. Lastly, we discussed a simple way to reuse previous knowledge, inspired by the framework of case-based reasoning.

Chapter 12

U-shaped phenomenon in Incremental Learning

In Chapter 10, we introduced a framework for incremental learning based on minimum description length principle, and we claimed that this framework is generic. However, we only illustrated it with examples taken from machine learning, with vector representation of data. The purpose of this short chapter is to present an application taken from a completely different domain and to show the validity of our framework.

The domain we consider here is the phenomenon of U-shaped learning in language acquisition. This phenomenon is well-known in the domain of cognitive sciences and corresponds to a non-uniform learning curve, divided in three steps: learning, un-learning and re-learning. This phenomenon is observed, in particular, in the acquisition of language by children ("he went" $\rightarrow$ "he goed" $\rightarrow$ "he went"). We will show that this phenomenon is well-illustrated by a complexity-based framework.

The remainder of this chapter is organized as follows: After a brief introduction on language acquisition and existing models of U-shaped phenomena, we propose a detailed description of our framework in Section 12.2. This section will convey notions that were described in the very beginning of this thesis, in Chapters 4 and 5 Finally, in Section 12.3, we present the results of simulations based on our model.

12.1 Context: Language Acquisition

The phenomenon of U-shaped learning describes a three-step procedure of learning, unlearning and relearning. A well-known occurrence of a U-shaped curve is the acquisition of aspects of language, in particular the past-tense learning in English: at first, children learn correct syntactic forms (eg. "play / played", "sing/sang"), then proceed to an overgeneralization (eg "play/played" but "sing/singed") and eventually acquire the correct grammatical rule and the exceptions. This phenomenon is counter-intuitive as it stands in opposition to the idea of a monotonic cumulative learning (ie. the idea of a *continuous model of cognitive development*). However, cognitive sciences and developmental psychology have shown empirical evidence for this phenomenon in various domains, among which language learning (Bowerman, 1982; Marcus et al., 1992), understanding of physical notions (Stavy, 2012; Bowerman, 1982) or face recognition (Carey, 1982). In this chapter, we will focus on language acquisition, but we will discuss how our model could be applied to other domains and be extended to explain the phenomenon as a whole.

Besides this phenomenon, an intriguing problem concerns the acquisition process itself. Many authors estimate that feedback is not necessary to acquire a language correctly (Bowerman, 1988; Marcus, 1993); others consider that only noisy or indirect signal is sent to children, such as clarification questions, physical manifestations of incomprehension (Demetras, Post, and Snow, 1986). This ability is directly related to U-shaped learning, negative feedback being the most direct way to overcome over-generalization.

Attempts have been proposed to model and explain the U-shape phenomenon. In particular, a common line of research follows Gold's model of language learning (Gold, 1967; Jain, Osherson, Royer, and Sharma, 1999) and studies the logical necessity of U-shape performance when learning is based on *positive evidence* (ie. when learners get information about the language in a naturalistic environment, for instance by hearing sequences as they occur). A particularly interesting results states that a U-shape behavior is not necessary in case of explanatory learning (ie. when the learner converges to a single hypothesis) with full memory (Case and Kötzing, 2010; Baliga, Case, Merkle, Stephan, and Wiehagen, 2008). Several studies have investigated the impact of memory limitation onto U-shaped learning (Carlucci, Case, Jain, and Stephan, 2007). In particular, the framework of *iterative learning* assumes that the system has only access to the last conjecture and the current data only (Lange and Zeugmann, 1996). A complete review of the main results regarding U-shape with positive evidence can be found in (Carlucci and Case, 2013).

A different point of view is based on simplicity arguments and state that the learner actually infers a probabilistic grammar that provides the simplest encoding of the inputs (Hsu, Chater, and Vitányi, 2013). This idea of using complexity to describe the U-shaped phenomenon is close to our idea. However, the approach proposed by these authors is based on pure complexity which is non-computable. For this reason, their approach is not sufficient to provide a cognitively plausible and easy-to-use model of language acquisition. The model is based on Solomonoff's general theory of induction (Solomonoff, 1964). It relies on Bayesian theory where the prior is approximated by a universal probability distribution: Despite the real interest of asymptotic results and of probabilistic modeling, such results do not encompass the (at least partial) potential acquisition of a language from very few examples (one-shot learning) (Tenenbaum, Kemp, Griffiths, and Goodman, 2011). These models are based on the assumption that the grammar is probabilistic: In particular, the complexity of a grammar is supposed to be related to the probability of its use. We think that this idea is restrictive and that a more direct computation of complexity, not related to probabilities, can be proposed. Because of its apparent similarity with our propose model, we will discuss this framework in more details in the following.

Our point of view lies at the intersection of three approaches: Gold's computational learning theory (Gold, 1967), simplicity theory (Dessalles, 2013; Hsu, Chater, and Vitányi, 2013) and grammar learning by analogy (Lepage, 1998). We consider that agents learn morphological transformations by memorizing a compressed table of association which is continuously updated when new words occur. From this point of view, the inferred grammatical rules correspond to the compressor itself.

12.2 A modeling Framework

In this section, we present the general framework used to interpret the phenomenon of U-shaped learning.

12.2.1 Assumptions

We follow a Gold-like approach to language acquisition. We consider that the learner faces a stream of language cases. No indication is given to the learner regarding the correctness of the case nor its origin (cases produced by the learner himself or by external agents). This hypothesis is meant to avoid any bias brought by negative evidence. In particular, it attempts to model the uncorrected mistakes made by the learner which might be taken for granted.

For simplicity purpose, we restrict our study to a problem of conjugation or declension in which the learner has access to the base form of a word and to its modified form (ie. declined or conjugated form). We impose a couple of straightforward restrictions. First, the learner is supposed to **learn rules for one and only one task**: For instance, the learner acquires knowledge on past formation in English, or accusative case in Latin... This assumption is equivalent to having a learner able to make a distinction between all the possible learning tasks it could face. Secondly, the learner is supposed to have a **prior access to the whole vocabulary**. This means that the learner cannot encounter a new word during learning. Finally we suppose that, when facing an inflected word, the learner has access to its base form: For instance, when facing the word “was”, the learner knows that it is a transformation of “be”.

We will discuss the validity of these *a priori* hypotheses in the concluding remarks. We will also consider their impact on a more realistic modeling of actual language acquisition.

The described process is the following: at a time-step t, the learner infers a grammar G_t based on its previous grammatical knowledge G_{t-1} and on memorized forms. It has access to two distinct memories: the perceptive memory, memorizing received data only, and the generative memory in charge of storing the current state of grammar. In practice and in the context of an easy-to-use model, we will not consider general grammatical systems, but only simple cases.

No restriction is imposed on the origin of the observed data. In particular, the data can be either generated by a rigorous speaker (hence be exact) or be generated by the learner itself. Allowing the learner to interfere with noiseless data produced by the environment is intended to present a more realistic modeling: indeed, it has been shown that children often lack negative feedback when they make mistakes, hence have confirmation of incorrect inferred grammars (Bohannon and Stanowicz, 1988).

The idea of this modeling is to offer a continuous description of language acquisition. A newly inferred grammar has to be inspired both by the previous state (which enforces continuity) and by a description of data (which enforces correctness). The general grammatical inference task can be seen as a trade-off between these two general tendencies: fitting data and continuity in time.

12.2.2 A Complexity-Based Framework

In this framework we proposed, we can see a pattern similar to the general principle of online learning as described in Chapter 10. We propose to describe here in more details how the two frameworks can be related.

At each time step t, the learner faces a word (in non-inflected form) X_t and aims to produce its inflected word in the given context Y_t. For instance, it is possible to have X_t =“play" and Y_t =“played".

The model M_t corresponds to a restriction of the general grammar to the task of interest. This grammar is supposed to explain the transformation $X_t \rightarrow Y_t$.

Based on these notations, it is still possible to use the objective function given in Equation 10.4. For simplicity purpose, we choose the association function Δ_t to be $\Delta_t(t-1) = 1$ and $\Delta_t(u) = 0$ for all $u < t-1$. Such an association function corresponds to a first-order process where the model at time t is determined by its predecessor only, hence the model at time $t-1$. With this convention, we obtain an objective function equal to:

$$\sum_t K(M_t|M_{t-1}) + K(X_t|M_t) + K(Y_t|M_t, X_t) \tag{12.1}$$

Such as for data stream mining, this objective cannot be optimized once and for all when all data have been observed. In this application, the stream is potentially unlimited and decisions have to be taken on the fly, for the learner to have an updated grammar at any time. The greedy approach that is inherent to stream mining is a necessity here: The learner improves its grammatical knowledge in order to have locally the best of its knowledge.

12.2.3 Computing Complexities

In order to choose the corresponding models, we rely on the description language proposed in Chapter 4.

Using this language, the minimal program to generate a declension (or conjugation) implements the following steps: 1) Store the general transformation rule into memory; 2) Apply transformation to the first radical; 3) Apply transformation to the second radical. The following program applies this process to the "rosa : rosam :: vita : vitam" example:

```
let(?, next, ?, 'm'),      // Step 1
  mem, 0 , 'rosa',         // Step 2
  mem, 0, mem, 'vita';     // Step 3
```

This program can be transformed into a binary code and thus used for complexity evaluation of the whole transformation. Even if this coding was explicitly designed for analogies, it is interesting to notice that it can be used in a more general setting without significant change. In particular, step 1 can be interpreted as the rule which can be applied to more than two examples.

12.2.3.1 Encoding the Grammar

We choose to represent the grammar (or, equivalently, the model) as an unordered list of rules and exceptions. A list is an instruction of the form "if [condition on radical], then [transformation]". For instance, the regular plural form in English can be described by the rule "If true, then add `s` at the end of the radical".

We choose to encode the rules as an instruction in our program which generates the ordered pair `radical : inflected`. Generating this ordered pair requires external information, which will be considered later, in the description of observations. An encoded rule describing the formation of plural form in French, as presented above, could be `?, next, ?, 's'`.

To these rules are added some exceptions, ie. cases that do not respect the rules. For example, the rule cited above for the plural in French does not apply to word

"cheval", the plural of which is "chevaux" and not "chevals". Exceptions are encoded as the full ordered pair: `'cheval', next, 'chevaux'`.

In the language, we choose the empty set of operators, which means that only operations of concatenation are permitted. This restriction is well-adapted to the domain of natural language, in which the formation of words does not rely on prior knowledge on the structure of the alphabet. Only the operator `repeat` could be of any interest, since some languages use the repetition in their grammar[1], but this operator can be easily replaced by its description. For example, the Indonesian plural rule can be expressed easily as `?, next, ?, ?`.

12.2.3.2 Grammar Transfer

For the evaluation of the transfer term $K(M_t|M)$, we propose to evaluate the way a grammar is transformed. Several transformations are possible:

- **A rule (resp. exception) is added:** The rule (resp. exception) must be given explicitly. The complexity is the complexity of the added rule (resp. exception).
- **A rule (resp. exception) is removed:** The id of the rule (resp. exception) must be given. The complexity is the complexity of the number of rules (resp. exceptions).
- **A rule is modified:** The condition of the rule remains unchanged, but the inflection is changed. This case is rare but can happen when an exception is incorrectly classified as a rule.

As a consequence, adding a rule is more costly in terms of complexity than removing one, unless the number of rules is already high. This observation is consistent with the intuition that a "small" grammar can be easily improved by adding new rules, while a very large grammar will benefit from being simplified.

Given a grammar M_{t-1}, a grammar M_t is given as a list of such procedures and the complexity $K(M_t|M_{t-1})$ is defined as the sum of the complexities of these procedures.

12.2.3.3 Encoding the Observations

The grammar model M_t is used to describe the observations (X_t, Y_t). Three cases are observed:

- **The observation is well described by the grammar:** Either the observation is well described by a rule or is already an exception. In this case, nothing is done.
- **The observation is not described by the grammar:** The observation is not an exception and no rule condition applies on it. In this case, the system has to determine a new rule out of one example (which is in general less complex than storing the observation as an exception).
- **The observation is incorrectly described by the grammar:** A rule applies on it, but the result does not correspond to the correct result. In this case, the system has to define if the observation is encoded as an exception or if a rule can be modified.

[1] We can cite Latin, in which makes use of syllable doubling in some forms of preterit (do → dedi), or Indonesian, in which reduplication is used for plural (orang → orang-orang).

Regarding the creation of rules, the minimum complexity strategy favors rules that propose the maximal factorization of X_t and Y_t. Consider for instance the observation (`work`, `worked`). Several rules can generate this observation, including `work` $\rightarrow$ `worked`, `k` $\rightarrow$ `ked` or simply $\emptyset \rightarrow$ `ed`. Obviously, the better compression is given by the last rule, which also corresponds to the maximal generalization.

12.3 Experimental Results

12.3.1 Causes of U-shaped Phenomenon

For the reasons exposed earlier, the model we propose allows us to test a couple of properties that might impact U-shaped learning in a very direct way:

1. **Finiteness of memory**: The limitation of memory has already been studied from a theoretical point of view (Carlucci, Case, Jain, and Stephan, 2007). The proposed framework offers a straightforward way to explore the influence of the size of the memory onto the learning process.

2. **Feedback on mistakes**: As discussed previously, children may lack negative feedback during language acquisition. We propose to explore the learning process when the learner faces correct samples only and in the presence of mistakes.

The purpose of this experimental part is twofold. First, we want to show that the proposed framework satisfies the desired properties in terms of language acquisition: the method is able to estimate rules describing observations and the global learning evolution follows a U-shaped evolution. Secondly, we propose to investigate the impact of both memory finiteness and feedback on mistakes onto U-shaped learning.

Another aspect that might play a role in the emergence of U-shaped curves is the impact of word frequency. Obviously, words do not have the same use frequency and irregular words are more frequently used (Plunkett and Marchman, 1991). It would be interesting to experimentally test this hypothesis, but this work has not been done here.

Following these simulation steps, a first conclusion is the convergence of the learning procedure toward a set of meaningful rules. These rules can be divided into two categories. The first category is made up of "regular" cases: they define what is intuitively admitted to be the standard past formation. The second category describes sub-classes of "irregular" cases. For instance, the algorithm detects that verbs ending in '-and' have their past in '-ood'. The automatic finding of such rules is a demonstration of how the compression favors general rules when possible.

Since the language generator is chosen to be pseudo-random, we notice that the induced grammar can change from one simulation to the other. Two different grammars can describe with equal quality the same set of words, but might differ in their rules and exceptions. This phenomenon is inherent to the incremental nature of acquisition: the order of apparition has a strong impact on the inference.

From these first conclusions, two conclusions emerge. First, our model is good enough to describe a rule acquisition process, even if we have not yet shown that this process follows a U-shaped behavior yet. The second conclusion is relative to inference itself: It can be shown that two systems facing the same situations but in a different order will infer different rules. This aspect illustrates the intrinsically subjective nature of cognitive language representation in the absence of feedback.

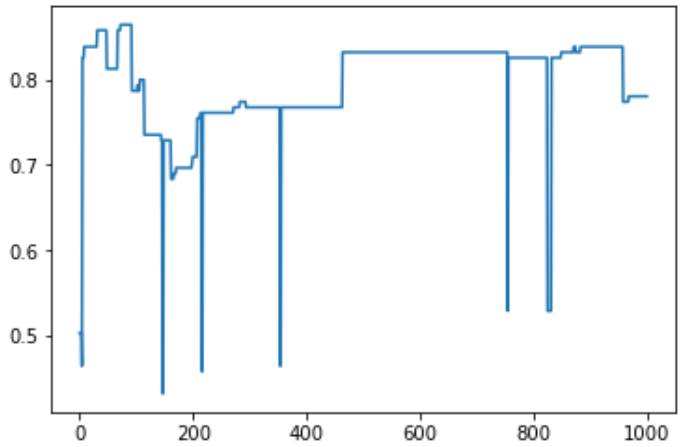


FIGURE 12.1: Generalization rate evolution during training for memory size of 5.

12.3.2 Finiteness of Memory

As an illustration of the impact of memory limitations on U-shaped learning, we consider language acquisition simulation with a limited history window and we show the influence of this parameter on the learning process.

When memory is very limited in size, we generally observe multiple short-term U-shaped phenomenons during acquisition (Figure 12.1). Even if the generalization rate curve has a global increasing tendency, large-amplitude drops are observed at times, corresponding to incorrect inferences by some rule.

When the size of the memory increases, the curve tends to become smoother and local drops disappear. Global U-shape becomes observable in most simulations, as depicted in Figures 12.2 and 12.3 for a window width of 20 and 100.

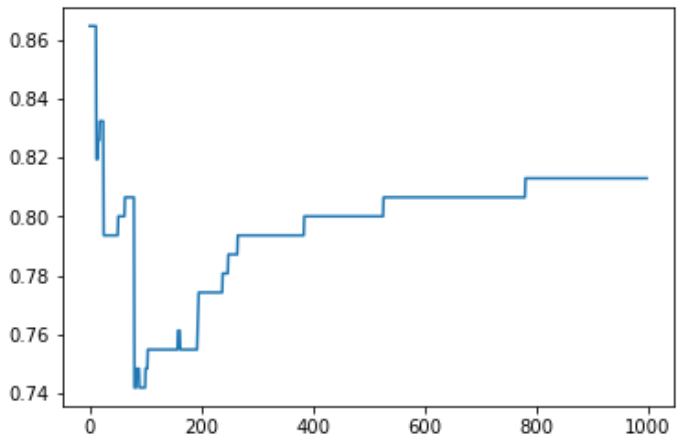


FIGURE 12.2: Generalization rate evolution during training for memory size of 20.

When the memory is not limited, U-shape phenomenon tends to vanish in general but might persist locally (Figure 12.4). It is interesting however to notice that, depending on the word order, learning may or may not converge to a perfect inference. In some situations, even with finite memory, the incremental nature of learning stops the system at a non-perfect inference stage.

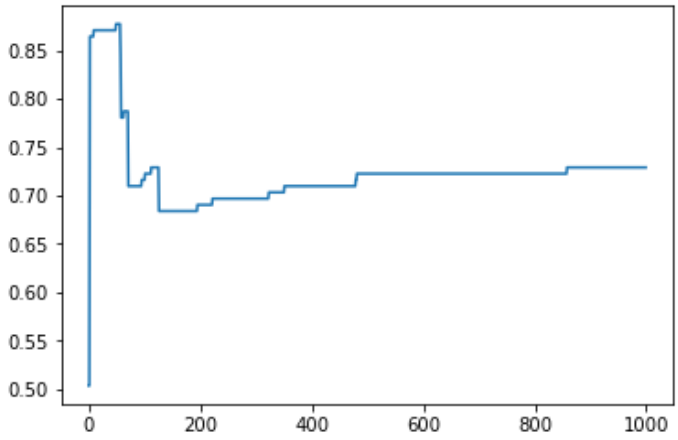


FIGURE 12.3: Generalization rate evolution during training for memory size of 100.

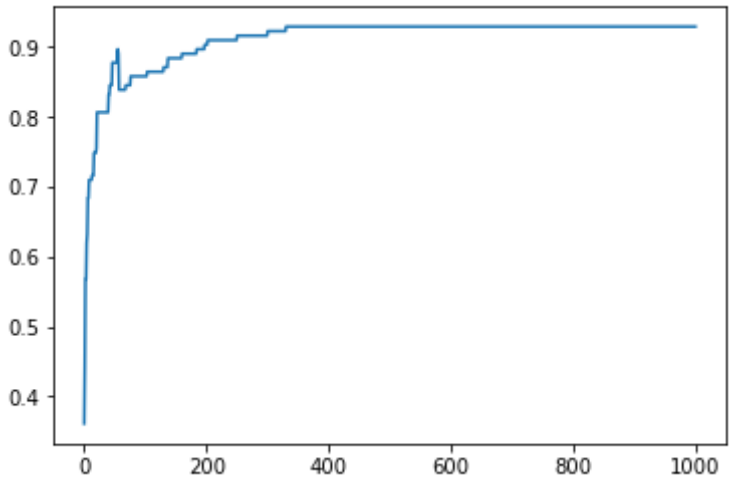


FIGURE 12.4: Generalization rate evolution during training for unlimited memory.

We can verify that memory size has a crucial impact on U-shaped learning. The proposed framework leads to results with similar properties to those observed in human language acquisition: despite a globally increasing behavior, drops in the learning rate can be observed when an overly general rule is inferred. In some situations, the system can be blocked in this poor representation which is locally optimal for it. This observation points to the necessity of an external learning process to correct potential mistakes of an acquisition from positive evidence.

12.3.3 Uncorrected Mistakes

We consider now that the learner faces uncorrected mistakes during learning. Such mistakes can be produced by the learner himself in the absence of correction. For instance, a child may overgeneralize past tense formation and say "'goed' instead of "went" without being corrected by his parents.

We incorporate a mistake probability in the generation of data and measure its impact on learning performance with a memory window of length 30.

As a first conclusion, we notice that adding mistakes does not change the U-shaped phenomenon, at least when the probability is low (Figure 12.5). When the probability tends to 1, the phenomenon is attenuated but remains visible in most cases.

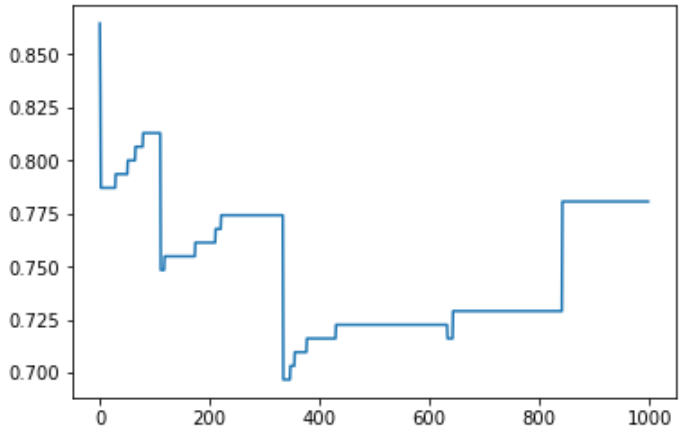


FIGURE 12.5: Generalization rate evolution during training with window size of 30 and mistake probability of 0.1.

Uncorrected mistakes influence the U-shaped phenomenon mainly on the overall performance. We notice that the best performance obtained with uncorrected mistakes is globally higher than the performance obtained without them. Besides the number of generated rules is much lower and no more "generalization over exceptions" is produced (rules such as "-and" transformed to "-ood").

Even if this result might seem counter-intuitive at first sight, it can be explained easily in the context of our study. Produced mistakes are often in the direction of over-generalization of regular rules . Thus, adding such mistakes to the learning procedure encourages the system to find regularity more than exceptions. In real language, irregular verbs are known to be more frequently produced, and they are still perceived as exceptions when they occur in a correct form.

12.3.4 Discussion

The results presented here are to be taken with care, since our model is based on strong hypotheses. The focus on the acquisition of one single aspect of language was necessary to validate our model, but at the expense of realism. In real situations, children have to acquire several aspects of language at once (semantic, various morphological features of conjugation, syntax...) and they do not focus on one single task. Our simplification consists in considering all these tasks as independent. It might be acceptable on a first level of analysis. The influence of other aspects does not necessarily impact the learning procedure directly in our case.

It may indirectly justify a second hypothesis we made: we assumed that the learner has access to the base form of words. In practice, this form is not always directly reachable, but it might be easier to recover from the whole context (in particular the semantic context).

Finally, the child does not have a complete knowledge of the target vocabulary and may hear inflected words, without having ever encountered the corresponding base form. A variant of our model could make the system able to infer the base form from the acquired rules. Our hypothesis still holds: in languages such as English,

irregular verbs are often the most frequent ones (Plunkett and Marchman, 1991) and rare words tend to show strong regularities. Based on this remark, we can reformulate our hypothesis as follows: learners are able to estimate the base form of irregular words because it is highly likely they have encountered them in the past, and the base form of regular words by taking advantage of their regularity.

12.4 Conclusion

In this chapter, we presented an application of our incremental framework in the domain of cognitive modeling. We presented it in the context of language acquisition. Based on very simple restrictions, we explained that the model M_t could be the encoding of a grammar and that the solution Y_t could be the estimated inflection of a base form X_t. Despite the limitations of our approach, it turned out to be successful to reproduce the phenomenon of U-shaped learning, which means a succession of three phases: learning, unlearning and relearning.

This result is interesting on several perspectives. From the perspective of our model, it tends to justify that the complexity-based framework we proposed models actual phenomena that can be observed in human cognition. From the cognitive point of view, it can be another argument in favor of the complexity-awareness of human beings, already pointed out by (Chater, 1999). Finally, from the perspective of learning theory, it raises a fundamental question: Is U-shaped learning restricted to grammar acquisition? In particular, can it be observed in non-symbolic domains, for instance in data stream mining? An extensive study of this question should be done in future works.

Part IV

Information Transfer in Unsupervised Learning

Chapter 13

Introduction to Multi-Source Clustering

In previous parts, we have explored several transfer problems in supervised setting, ie. where the solution is explicitly given, at least in one domain. In this final part, we explore a completely different task, the task of multi-source clustering. In this task, several clustering algorithms process the same data in order to produce a clustering. However, the algorithms do not have access to the same information (*views*) in datasets and process with their own biases. The purpose of multi-source clustering is to exchange information among the clustering agents in order to refine their decision or to find a consensus.

The work presented in this part has been done in collaboration with Jérémie Sublime (ISEP) and Basarab Matei (Université Paris 13).

In this first chapter, we propose a brief overview of the domain, reminding a couple of notions inherent to clustering and discussing algorithms and applications for multi-source clustering. We will introduce the two main tasks of the domain: *multiview clustering* and *collaborative clustering*.

The remainder of this chapter is organized as follows: In a first section, we present a general introduction to clustering. In Section 13.2, we propose a general introduction to multi-source clustering and its potential applications. We present a couple of state-of-the-art methods in cooperative clustering (when the multiple sources aim to find a consensus) in Section 13.3, and in collaborative clustering (no consensus required) in Section 13.4.

13.1 Reminder on Clustering

In this section, we present the general problem of clustering, in a single source setting.

13.1.1 Definition and Issues

The clustering task, unlike classification or regression, belongs to the family of unsupervised learning problems. Such problems are characterized by the absence of any label at the training step. Given a dataset $X = \{X_1, \dots, X_n\}$, a clustering method aims to find a solution vector $(y_1, \dots, y_n) \in \mathbb{N}^n$. The sets of the form $\{X_i : y_i = k\}$ for $k \in \mathbb{N}$ are called clusters. As exposed in Chapter 2, the clustering task corresponds to a labeling of the dataset, but this definition is not enough. A notion of "good labeling" is required and, informally speaking, relies on the idea that similar points have to be grouped together. However, this definition is extremely imprecise and the difficulties of clustering originate in the absence of a clear definition.

A first problem, that is inherent to the notion of similarity is transitivity. If a and b are similar and b and c are similar, then a and c are not necessarily similar. However, the concept of "belonging to a same cluster" is transitive ($sameCluster(a,b) \wedge sameCluster(b,c) \Rightarrow sameCluster(a,c)$). This simple observation points out a fundamental problem in the task of clustering. Consequently, it might happen, depending on the clustering algorithm, that two points are grouped in a same cluster but are less similar than two points that are not in the cluster. As a consequence, (at least) two interpretations of the pseudo-definition of clustering can be given. On the one hand, a clustering algorithm may avoid separating similar points; on the other hand, it may avoid grouping together points that are too dissimilar.

Another problem is the potentially large number of partitions that can be produced by clustering algorithms and that should be explored in order to find the optimal one. Consider a set of n objects that have to be partitioned in K groups. (Cornuejols, Wemmert, Gançarski, and Bennani, 2018) shows that the total number of partitions is equivalent to $K^n/K!$ as $n \to \infty$ and that, for $n = 25$, exploring the entire space of partitions would require 147,000 years (given one million partitions per second).

Considering these two observations, we understand that the search for a good clustering algorithm necessarily implies strong biases, both on the choice of a similarity criterion and on the exploration of the space of partitions.

Three families of clustering methods can be found:

- **Hard clustering:** Each object is associated to one and only one cluster.
- **Soft clustering:** Each object is associated to at least one cluster. An object can be present in several clusters, with the same degree.
- **Fuzzy clustering:** Each object is associated to all clusters with various degrees.

In this thesis, we will consider hard clusterings only.

13.1.2 Families of Algorithms

Clustering is a rather old problem, and a large diversity of methods have been developed. We propose a brief overview of the existing families of methods and of the most important algorithms. The purpose of this overview is not to be exhaustive but to provide general ideas on the algorithms. These ideas will be useful in the next chapter, when discussing how to adapt a large variety of methods into the framework we propose. For a more complete overview, we refer the interested readers to the dedicated surveys (Xu and Wunsch, 2005; Berkhin, 2006).

Prototype-based algorithms rely on the representation of data by a set of representative points, called *prototypes*. This idea, which is based on the notion of *vector quantization*, has already been introduced and discussed in this thesis (in particular in Section 8.2.2).

The most famous prototype-based algorithm is undoubtedly the *k-means* algorithm. This simple algorithm aims to minimize the following objective function over P:

$$f(X,P) = \min_{p_1,\dots,p_k} \sum_{i=1}^{k} \sum_{x \in \mathcal{C}_i} d(x - p_i)^2 \tag{13.1}$$

where $\mathcal{C}_i$ designates the i-th cluster, ie. the set of points in the dataset X that are attached to the i-th prototype. This objective function is minimized by alternating

two phases until convergence (MacQueen, 1967) First, the points in the dataset are associated to their closest prototype, then the position of each prototype is modified to correspond to the average position of its associated points. This procedure guarantees that the objective function decreases at each step and, thus, that the algorithm converges in a finite number of steps. This algorithm is very popular for its simplicity, however it has major drawbacks. It can be observed that the algorithm is extremely sensitive to the initialization of the prototypes. In fact, determining the optimal partition in k-means is NP-hard. Moreover, the formed clusters are, by construction, hyper-spherical and often fail to detect structures of different scales. Two variants of k-means, used when the distance between points are known, but not their positions in the space, are *k-medoids* and *k-medians*. Both select the prototypes inside the dataset, but k-medoid aims to optimize the same objective function as k-means, while k-median focuses on the distance to the prototype, rather than the square of the distance.

Another popular algorithm based on prototype-based representation is the *Affinity Propagation* algorithm (Frey and Dueck, 2007). This algorithm relies on the technique of message-passing in order to update two matrices: a responsibility matrix, which measures to what extent a point can be a good representative for another point, and an availability matrix, which measures to what extent it would be appropriate to select a point when taking the other points' choices. This two matrices are updated sequentially until convergence. The main advantage of Affinity Propagation is that it automatically chooses the optimal number of clusters. However, it is computationally more expensive than k-means.

Generative clustering methods rely on a probabilistic representation of data and their purpose is to infer the data distribution. This distribution is chosen to be a mixture of local distributions, for instance of multinormal distributions for the famous Gaussian Mixture Model (GMM). A mixture model combines multiple distributions by picking them randomly based on a multinomial distribution, the parameter of which corresponds to the probability of a point to be drawn by the corresponding local distribution. In the applicative case of clustering, each local distribution corresponds to a cluster, and the parameter of the multinomial distribution measures the probability of a point to belong to the corresponding cluster. If we denote by $\pi = (\pi_1, \dots, \pi_k)$ the parameter of the multinomial distribution and consider that the local distributions belong to a parametric class of distribution $\{p_\theta\}_{\theta \in \Theta}$, the corresponding likelihood is given by:

$$\mathcal{L}(X; \pi, \theta_1, \dots, \theta_k) = \prod_{x \in X} \left(\sum_{i=1}^{k} \pi_i p_{\theta_i}(x) \right) \tag{13.2}$$

This objective function is maximized using the Expectation-Maximization (EM) algorithm (Dempster, Laird, and Rubin, 1977), a procedure that alternates an update of the posterior probabilities of the clusters for points in the dataset and an update of the parameters of the distribution for fixed posterior probabilities.

Density-based methods aim to find regions of high-density of data and that are well-separated from other regions. The density is measured by the number of neighbors to a point, belonging to a neighborhood centered on the point of interest. The more data points are present in this area, the more the point belongs to a dense region and will be connected to its neighbors.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester,

Kriegel, Sander, and Xu, 1996) implements this idea in the most direct way. The algorithm inspects all the points sequentially and groups them in a same cluster if the number of points in a sphere of radius ϵ contains more than a fixed minimal number of points. A variant of DBSCAN, called Ordering Points To Identify the Clustering Structure (OPTICS) (Ankerst, Breunig, Kriegel, and Sander, 1999), considers that the parameter ϵ is optional. For each point, the algorithm determines a core distance, which corresponds to the maximal distance in a neighborhood of m points, where m is the threshold for dense regions. This distance is used in the definition of a reachability distance that measures to what extent two points can be linked together.

Finally, **spectral clustering** focuses on the idea that two points belonging to different clusters must be non similar (Shi and Malik, 2000). As a way to do this, spectral clustering algorithms define a weighted graph $\mathcal{G}$, the nodes of which are the data points. The vertex between X_i and X_j is associated to a weight W_{ij} (defined according to several criteria which will not be described here). Spectral clustering then uses the K orthogonal eigenvectors of the Laplacian matrix of $\mathcal{G}$ associated to the smallest eigenvalues.

13.1.3 Performance Measures

The unsupervised nature of clustering, as opposed, for instance, to classification, makes the evaluation and validation task more difficult. In classification, the quality of a classifier is measured simply by the number of produced errors. In clustering, the lack of ground truth does not allow to define such a direct measure. Moreover, the discussion on the difficulty of clustering made it clear that there is no absolute quality criterion. This observation is validated by the impossibility theorem of (Kleinberg, 2003), which states that clustering cannot satisfy more than two of the following properties: scale-invariance, richness and consistency.

Several validation criteria have been developed (Halkidi, Batistakis, and Vazirgiannis, 2002), each one measuring a different characteristic of the clustering. They are frequently divided in two classes: unsupervised indexes, which do not exploit any external information, and supervised indexes, which measure a similarity with a known partition of data.

In our experiments, we will focus on three indexes: Davies-Bouldin index, Silhouette index and Rand-index.

Davies-Bouldin index is an unsupervised index that measures the compactness and the separability of clusters (Davies and Bouldin, 1979). It is defined as follows:

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\frac{\Delta_i + \Delta_j}{D(C_i, C_j)} \tag{13.3}$$

where $\Delta_i = \min_{x,y\in\mathcal{C}_i} d(x,y)$ represents the minimal distance between two points in a same cluster and $D(.,.)$ is a measure of separation of two clusters. The index is not normalized but lower values indicate a better quality clustering.

Silhouette index is an unsupervised index that measures the compactness and separability of clusters (Rousseeuw and Leroy, 2005). It is defined as follows:

$$SC = \frac{1}{K}\sum_{i=1}^{K}\frac{1}{|\mathcal{C}_i|}\sum_{x\in\mathcal{C}_i}\frac{b_x - a_x}{\max(a_x, b_x)} \tag{13.4}$$

where a_x designates the mean distance between instance x and other objects in the same cluster, and b_x the mean distance between instance x and all objects that do

not belong to the same cluster as x. Positive values of silhouette index correspond to good clusterings.

The adjusted Rand index (Hubert and Arabie, 1985) is a supervised index that exploits an external partition of data, considered as ground truth. This index is equal to 1 if the two partitions are equal.

13.2 Multi-Source Clustering: An Overview

The clustering framework presented until now is classical and does not involve any transfer in the sense of what has been investigated in supervised learning. We now introduce the problem of multi-source clustering, which will be studied in the next chapters. The domain of multi-source clustering originates from various practical situations that have emerged recently. The purpose of this section is to illustrate these problems as well as the need of a form of collaboration in clustering.

13.2.1 Overcoming the Individual Biases

As exposed before, clustering is an ill-defined problem, the intuitive definition of which contains inherent ambiguities. As a consequence, any clustering algorithm is intrinsically biased toward some tasks. Among the methods that have been described earlier, we have for instance a clear bias of k-means for spherical clusters (which will prevent it from getting satisfying results on datasets such as the half-moons introduced before in this thesis), and it is well-known that DBSCAN is particularly inefficient for clusters of different sizes or densities.

The idea of refining the predictions by collaboration is well-known in supervised clustering, where ensemble learning (in particular bagging or boosting methods) is used to overcome the local failures of classifiers. Applying the same idea in clustering is thus tempting and would consist in a collaboration of the various clustering algorithms. A discussion of this idea is proposed in Chapter 15.

13.2.2 Clustering in Distributed Environments

As seen in the case of data streams, modern technologies change the way data are generated. Mobile sources of data are now frequent and produce large amounts of data in heterogeneous distributed sources. These new distributed environments have been studied in the perspective of supervised learning (Vanhaesebrouck, Bellet, and Tommasi, 2017) and unsupervised learning (Depaire, Falcón, Vanhoof, and Wets, 2011). In these environments, centralizing the algorithms is not possible due to the volume of data that would have to be stored.

Besides, as mentioned by (Pedrycz, 2002), such distributed environments can also be limited by privacy issues: The different sources might not have the rights to access data of other sources. Exchanging solutions and parameters of clustering can be a solution to provide other sources with local information without providing data directly.

13.2.3 Multi-view Data

In traditional datasets, a data point is represented by a fixed number of features. The learning is done from all the features at once. In some datasets, the features can be of different types or origins. In such situations, (Zimek and Vreeken, 2015) points out

that the obtained clusters might vary from one origin to the other. This difference is a consequence of the different distributions of data.

Consider for instance the dataset used in (Houthuys, Karevan, and Suykens, 2017). This dataset consists of daily meteorological data from multiple European cities. Each day corresponds to an instance and each city can be interpreted as a view, hence as a different aspect of the data. In the context of a clustering task on this dataset, a question arises: Are all the views to be merged together, or do they bring a variety of interpretation that must be held? On the one hand, classical clustering techniques cannot run on multi-view data. On the other hand, not only the interpretation of clustering meteorological conditions from multiple days is unclear, but also a merged solution would ignore the local specificity of the weather in the different considered cities.

13.2.4 The Solution of Multi-Source Clustering

All these problems make it clear that traditional clustering algorithms may have weaknesses when addressing more sophisticated datasets, or data produced in mobile and distributed environments. To cope with this problem, an idea emerged of cooperation between several clustering algorithms. This idea is inspired by the ensemble methods in supervised learning, but these techniques cannot be employed directly because of the major differences between classification and clustering. In classification, labels have a semantic interpretation, and it is sufficient to compare the labels together in order to determine if two classifiers are coherent together or not. However, in the unsupervised setting, this operation is not possible anymore: The labels of the clusters vary from one method to the other and are completely arbitrary. This simple but major difference requires an adaptation in the techniques, that will be described later.

In practice, two problems are studied in multi-source clustering:

- **Cooperative clustering:** This task consists in extracting a consensus out of multiple local clustering algorithms.
- **Collaborative clustering:** Unlike cooperative clustering, collaborative clustering does not aim to find a consensus but, on contrary, to refine the solutions found by local algorithms by using external information brought by other algorithms. Collaborative clustering can be used as a first step for cooperative clustering.

In the following sections, we review briefly the main issues and methods existing in the domain of cooperative clustering and collaborative clustering. This review is not exhaustive and we refer the reader to specialized surveys such as (Cornuejols, Wemmert, Gançarski, and Bennani, 2018) or (Vega-Pons and Ruiz-Shulcloper, 2011) for more information.

13.3 Cooperative Clustering

The family of cooperative clustering algorithms regroups all methods that aim to extract a consensus from various local clustering solutions. Cooperative learning generally proceeds in two steps: First, the local solutions are estimated without any communication between the algorithms; then the algorithms exchange their information in order to determine a consensus. (Vega-Pons and Ruiz-Shulcloper, 2011)

classifies cooperative clustering in two categories: consensus based on objects co-occurrence and consensus based on median partition.

13.3.1 Consensus Based on Objects Co-Occurrence

The consensus functions based on objects co-occurrence consider how many times two objects belong to the same cluster. When most local clusterings group two objects together, there is a higher chance that these objects are grouped together in the consensus.

A first direction followed by these methods consists in solving the **labeling correspondence** problem, hence to establish a cross-method mapping between the labels of clusters.

For instance, SAMARAH (Wemmert and Gançarski, 2002) offers a first elegant solution that works with any kind of local hard clustering algorithms. The method exploits the existence of good correspondences between the local solutions. The method can be divided in three steps: local clustering, collaboration and consensus. The collaborative steps is based on the results of local clusterings and proposes to build correspondences between clusters with help of a probabilistic confusion matrix. The mapping is then refined by solving conflicts locally. The idea of this refinement is very close to the method we will propose in Section 14.3. After the refinement, the authors propose an aggregation algorithm based on a majority vote.

Among other methods based on pairwise relabeling, we could mention for instance (Ayad and Kamel, 2010) which addresses this questions in terms of multi-response regression.

A solution to avoid the labeling correspondence problem consists in using a **co-association matrix**, hence merging the local partitions into a matrix, the value of which represents the proportion of times data x_i and x_j belong to a same partition. A simple approach consists then in thresholding the matrix, such as done by (Fred, 2001) which keeps all associations greater than .5.

13.3.2 Consensus Based on Median Partition

A completely different direction considers that the consensus solution corresponds to a median partition, hence a “barycenter" of the local solutions. Such approaches aim to minimize an objective of the form:

$$S^* = \arg\max_S \sum_{j=1}^{J} \Delta(S, S^j) \tag{13.5}$$

where J is the number of local algorithms and Δ a similarity measure between clustering solutions.

The main difficulty of such methods is then the construction of the similarity measure Δ. Various strategies can be found in the literature. The counting pairs similarity measures are based on the pairwise correspondences. These measures are often inspired by supervised indexes as exposed in Section 13.1.3. On the other hand, some similarity measures use a comparison of clusters considered as pairs. They can be based on measures such as Jaccard distance (Ben-Hur, Elisseeff, and Guyon, 2001).

A class of measures are defined with the tools of information theory. However, to our knowledge, they all rely on classical information theory and not on complexity. Since Kolmogorov complexity is more or less related to mutual information, one

may cite (Strehl and Ghosh, 2002) which defines function Δ based on normalized mutual information.

13.3.3 Discussion

As observed in Table 1 of (Vega-Pons and Ruiz-Shulcloper, 2011), very few algorithms seeking consensus actually rely on the objects and on the characteristics of the local clustering algorithms. Among them, the locally adaptive clustering algorithms explicitly use the local object distribution, but mostly rely on prototype-based methods. For instance, a method like Weighted Similarity Partition Algorithm (Domeniconi and Al-Razgan, 2009) considers weighted distances of points to the center of clusters.

This approach of not considering the local parameters nor the objects is arguable. The main advantage of this choice is the global independence of the proposed methods to the nature of the algorithms. As a consequence, most cooperative clustering algorithms can find consensus for any kind of local algorithms, and allow cooperation between local algorithms of different nature. On the other hand, considering local objects and parameters would be a strength, since they can bring valuable information to the consensus and refinement procedures.

13.4 Collaborative Clustering

Unlike cooperative clustering, the goal of which was to extract a consensus from a list of partitions, the purpose of collaborative clustering is to refine the solutions found by local clustering algorithms. Collaboration can be a step in a cooperative process, but here we consider it as the main objective.

One of the first appearances of this idea can be found in the works of (Pedrycz, 2002) which investigates the problem of collaboration in distributed environments. The methodology of the paper introduces a classical process in two steps: a local step where local algorithms are trained on their dataset independently of each other, followed by a collaborative step where the computed solutions and parameters are estimated in order to refine the solutions. This technique has strong limitations though: in particular, it can work only with a fuzzy k-means algorithm and a fixed number of clusters.

Following the same direction, some work has been done to adapt the procedure to other families of clustering algorithms: Self-Organizing Maps (Grozavu and Bennani, 2010), Generative Topographic Maps (Ghassany, Grozavu, and Bennani, 2012b) or Gaussian Mixture Models (Bickel and Scheffer, 2005; Cleuziou, Exbrayat, Martin, and Sublemontier, 2009).

A strong limitations of the methods presented above is their dependency on one single type of clustering algorithm. Some efforts have been done recently (Sublime, Matei, Cabanes, Grozavu, Bennani, and Cornuéjols, 2017) to propose a general collaborative algorithm which would apply to various types of clustering algorithms, but the proposed method applies only to probabilistic clustering algorithms.

13.5 Conclusion

In this chapter, we introduced the problem of collaboration and cooperation in unsupervised learning. Due to its differences with supervised learning, it is not possible to apply directly the existing techniques such as Bagging. We have shown that

two families of methods have emerged, that pursue different objectives: On the one hand, *cooperative clustering* aims to find a consensus from different data clustering; on the other hand, *collaborative clustering* aims to refine the solutions found at a local level by exploiting global information brought by other clustering algorithms.

A brief study of the existing techniques enlightened an interesting phenomenon. There does not exist a general method that exploits the data X and the parameters of the local algorithms while enabling collaboration between algorithms of various nature. Cooperative algorithms mostly rely on the produced solutions only in order to estimate an average common solution, and collaborative algorithms use the data points and local parameters but specialize on the collaboration between algorithms of the same nature. In the next chapter, we will show how Minimum Description Length principle can be used in order to define a generic method that exploits local information.

Chapter 14

Complexity-based Multisource Clustering

In the previous chapter, we introduced the general problem of multi-source clustering, as well as a couple of methods that have been developed to solve this problem. It appeared that these methods either lack generality or are not entirely satisfying in the way they describe data. The purpose of this chapter is to present a framework, based on a descriptive model, that helps solving this task in a general setting.

The main problem observed in the domain of unsupervised ensemble learning is the difference of descriptions of the different algorithms. Indeed, there is no direct way to compare the parameters of two algorithms such as K-Means and DBSCAN. We show here that Kolmogorov complexity is a simple and intuitive language to describe two different domains or views in a common unit of measure.

The remainder of this chapter is organized as follows: We first present our model of collaboration. In Section 14.2, we describe how local clusterings can be measured by complexity. In Section 14.3, we present a simple algorithm based on a direct simplification of the model for collaborative clustering. Finally, in Section 14.4, we provide experimental results obtained with our algorithm.

This chapter is an extended version of the article (Murena, Sublime, Matei, and Cornuéjols, 2018).

14.1 Graphical Model for Unsupervised Collaboration

In this section, we propose a DGM for the task of unsupervised ensemble learning.

14.1.1 Notations

We consider a dataset X that can be divided into J views, denoted by $X^1, \dots, X^J$. A view corresponds to a restricted representation of the dataset. Each data point has a representation on each of the views. We call N the number of points.

We consider J clustering algorithms, denoted by $\mathcal{A}^1, \dots, \mathcal{A}^J$. A clustering algorithm is defined as a mapping from the data points to integers. The clustering algorithm $\mathcal{A}^j$ processes view X^j and outputs a solution vector $S^j \in \mathbb{N}^N$. In practice, we consider that the number of clusters is finite, and equal to K^j for algorithm $\mathcal{A}^j$. This number can differ from one algorithm to the other.

A clustering algorithm $\mathcal{A}^j$ can be associated to a parameter $\theta^j \in \Theta^j$. The parameter set may differ from one clustering algorithm to another.

14.1.2 A Model for Collaboration

In Section 9.3.1, we presented a graphical model corresponding to transfer learning for multiple target tasks (Figure 9.1). The framework we suggest for multi-source clustering is very close to the introduced DGM.

The purpose of multi-source clustering is to provide a lossless description of *all* views. We propose to follow the same direction as done in the previous chapters and to consider that the transfer of information is not managed at the level of data points but at the level of the underlying models. In our context, the model will be the characterization of the clustering algorithms.

The resulting DGM is described in Figure 14.1 and corresponds to the following objective function to minimize:

$$K(S) + \sum_{j=1}^{J} K(\theta^j) + K(S^j|S) + K(X^j|\theta^j, S^j) \tag{14.1}$$

We introduce the term S, that we call a *meta-solution*. This term can be interpreted as a consensus of all local clustering algorithms on the dataset.

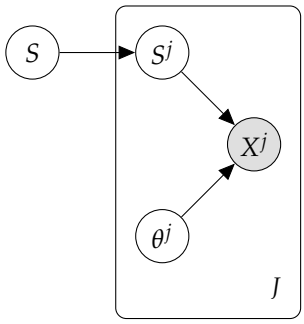


FIGURE 14.1: Model-based DGM for multisource clustering.

The proposed method is generic since it does not make any assumption on the nature of the algorithms nor on the representation of the views. Besides, it offers a solution to all problems that can be addressed by multi-source clustering:

- The parameters θ^j offer a description of the clustering on j-th view. In some cases, these parameters can be used for generalization (when new points arrive and have to be associated to one of the pre-trained clusters.
- The solutions S^j correspond to a partition of data on j-th view. These solutions are refinements of the solutions without collaboration, ie. with algorithms $\mathcal{A}^j$ only. Consequently, these partitions are the solution of the problem of collaborative clustering.
- The meta-solution S is independent of the views and can be interpreted as a global consensus. The local solutions S^j are described based on this meta-solution which can be then interpreted as a "*barycenter*" of the local solutions. The meta-solution S corresponds to the solution of multi-view clustering.

As a final remark, we can observe that the local solutions S^j are determined according to a trade-off between good local description (which implies a low value of $K(X^j|S^j, \theta^j)$) and good collaboration with other local solutions (which implies a low value of $K(S^j|S)$). We also notice that we do not consider any meta-parameter θ.

This is due to the fact that, unlike the local solutions, the local parameters are all defined on different sets. A consequence of this choice is that there is no possible consensus on generalization: If a new point is observed (with various representations on all views), it is not possible to associate it directly to a cluster at the global level.

14.2 Complexity of Local Clustering

In this section, we propose to express the term $K(X^j|\theta^j, S^j)$ for various classes of clustering algorithms.

14.2.1 Complexity of Prototype-Based Models

Prototype-based models regroup a large number of clustering techniques including K-Means, SOM, GTM or Affinity Propagation. As exposed in the previous chapter, these methods rely on a lossy representation of data based on points (belonging or not to the dataset X) called prototypes. The purpose of prototype-based methods is to determine the optimal position of the prototype in the input space in order to minimize a general objective function. It has been discussed in Section 8.2.2 why the prototype-based methods allow data compression.

Obviously, the prototype position determined by the cited algorithms does not necessarily correspond to the minimum complexity position. However, we assume that they provide a good approximation of optimal position in the sense of Kolmogorov complexity. This property corresponds to a research bias: Instead of exploring the whole space in order to get the optimal solution, the choice of a specific algorithm constrains the search of the optimum.

Prototype-based models are associated to a parameter θ describing the positions of prototypes. The solution vector S corresponds to the solution, hence to the point-prototype association. Given solution vector S and parameter θ, the construction of the position of a point is simple: It is based on the relative position of the point to its attached prototype.

All the corresponding computations are identical to the computations given in Section 8.2.2. We refer the reader to this section for the corresponding values of complexity.

14.2.2 Complexity of Probabilistic Models

Probabilistic models propose to model clusters by their density. They are often based on mixture of models, in particular mixture of Gaussian distributions (GMM algorithm).

The parameter θ associated to a probabilistic model corresponds to the parameter of the distribution (we consider the case of parametric distributions). The solution vector designates the distribution in the mixture to which each point is associated. In order to actually compute the complexity, we use the property that the complexity of a point x given a distribution p is upper-bounded by $K(x|p) \leq -\log p(x) + \mathcal{O}(1)$.

In particular, for a mixture of k distributions $(p_1, \ldots, p_k)$, a point in cluster i will be described with a complexity:

$$K(x|S, \theta) = -\log p_i(x) \tag{14.2}$$

14.2.3 Complexity of Density-Based Models

The family of density-based models corresponds to algorithms that use proximity of points in the dataset to form the clusters. These methods have been presented in previous chapters.

Even if these algorithms do not rely on an explicit parameter θ, it is possible to propose a description of points based on a reordering of the dataset, which would then be seen as the parameter of the algorithm. The density-based models aim to find the better attachment of points inside the dataset. This is in particular the idea proposed by OPTICS (Ankerst, Breunig, Kriegel, and Sander, 1999).

Based on this idea, the computation of the complexity can be done as follows. We denote by π_i the index of the parent of point i in the ordering proposed by the method. Exactly as suggested for the prototype-based method, the idea will be to describe the position of a point by its relative position with respect to a reference point, which is not a prototype in this case but the parent in the ordering. Points that have no parents (hence first point of a class in the ordering) are described by their absolute position. The total complexity is then given by:

$$K(X^j|S^j,\theta^j) = \sum_{i=1}^{n} K(X_i^j|X_{\pi_i}^j) + \mathcal{O}(1) \tag{14.3}$$

In this expression, the value of π depends on the solution S^j: two points can be linked only if they belong to the same class. Thus, changing S will change the order of the points and thus affect the complexity.

14.2.4 Complexity of Other Models

We consider now the case of models that cannot be related to any of the three previous models. We have not used such models in our study, but the proposed framework can be adapted to take such models into account. Three solutions can be found in order to assess the complexity of data given such models.

- **Ignoring the model:** The simplest, but least satisfying, solution consists in ignoring the model (S^j,θ^j) and estimating the complexities of data as:

  $$K(X^j|\theta^j,S^j) = K(X^j) + \mathcal{O}(1)$$

  This approach consists in ignoring the local description of models and focusing on the collaborative part.

- **Adding prototypes:** Prototypes are very useful and can be added to the problem even if they are absent at first sight. The prototype of a class can be defined as the mean position when data are elements of a vector space, or by equivalent ways in other cases. This method is simple but has major drawbacks. The position of prototypes found in this way can be extremely arbitrary, such as in the example of concentric cases, where the prototype for the inner cluster and the outer cluster are identical.

- **Exploiting density:** A convenient way to measure complexity is to adopt a similar strategy as proposed for density-based models. The idea is to describe a point in a cluster based on its relative position toward another well-chosen point in the same cluster.

14.3 Algorithm for Collaborative Clustering

In this section, we explain how we optimize the objective function in Equation 14.1. In the scope of this work, we consider only the case where the solutions $S^1, \cdots, S^J$ produced by the algorithms are hard partitions, and therefore can be described as vectors.

14.3.1 Forgetting Consensus

Even if the framework offers the opportunity to find a consensus, we focus on the problem of collaborative clustering, hence on refining local solutions. Since S is used only as an intermediate parameter, we can eliminate it from the algorithm.

To do so, we first isolate the minimization over S in the objective of Equation 14.1:

$$\begin{aligned}&\min_{S,S^1,\dots S^J} K(S) + \sum_{j=1}^{J} K(\theta^j) + K(S^j|S) + K(X^j|\theta^j, S^j)\\ &= \min_{S^1,\dots,S^J} \sum_{j=1}^{J} \left[K(\theta^j) + K(X^j|\theta^j, S^j) + \min_{S} \left(\frac{1}{J} K(S) + K(S^j|S) \right) \right]\end{aligned}$$

We consider that data are independent in the description of S, which implies that the complexity term $K(S)$ is constant and can be ignored in the minimization. In order to forget the consensus term S, we use the following proposition:

Proposition 10. *If $S^1, \dots, S^J$ are J solution vectors such that $K(S^i)$ does not depend on i. If $\mathbb{S}$ corresponds to the space of solution vectors, then the following inequality holds true:*

$$\min_{S \in \mathbb{S}} \sum_{j=1}^{J} K(S^j|S) \leq \frac{1}{J-1} \sum_{j=1}^{J} \sum_{i \neq j} K(S^j|S^i) + \mathcal{O}(J) \tag{14.4}$$

Proof. To prove this proposition, we use two properties from Table 5.1: $K(x|y) \leq K(x) + \mathcal{O}(1)$ and the chain rule. Based on these two inequalities, we have:

$$\sum_{j=1}^{J} K(S^j|S) \leq \sum_{j=1}^{J} K(S^j) + \mathcal{O}(J) \leq \sum_{j=1}^{J} \min_{i \neq j} \left\{ K(S^i) + K(S^j|S^i) \right\} + \mathcal{O}(J)$$

We observe that the minimal value over a finite set is necessarily lower than the mean value in the set, which means that

$$\min_{i \neq j} \left\{ K(S^i) + K(S^j|S^i) \right\} \leq \frac{1}{J-1} \sum_{i \neq j} \left\{ K(S^i) + K(S^j|S^i) \right\}$$

This observation, as well as the hypothesis that $K(S^i)$ is constant, leads to the result. □

In the minimization process It is important to note at this point that this change is a purely mathematical trick and has no real foundation in terms of Turing machine description: in this setting, a local solution would be constructed from another local solution, but loops are not prohibited (for instance S^1 constructed from S^2 and S^2 constructed from S^1), which is not possible from a physical point of view.

The simplified objective for complexity minimization is then:

$$\mathbf{S}^* = \arg\min_{\mathbf{S}} \sum_{j=1}^{J} K(X^j|\theta^j, S^j) + K(\theta^j) + \frac{1}{J-1} \sum_{i \neq j} K(S^j|S^i) \tag{14.5}$$

14.3.2 Global Approach

Following the model of other collaborative and multi-view algorithms, the optimization is done in 2 steps (Grozavu and Bennani, 2010; Sublime, Matei, Cabanes, Grozavu, Bennani, and Cornuéjols, 2017):

- A **local step** during which each algorithm $\mathcal{A}^j$ processes its local view X^j and produces a first model $M^j = \langle \theta^j, S^j \rangle$ based only on the local information. These local models are used as initial values.
- A **global step** during which Equation (14.5) is optimized.

The key difficulty of the algorithm lies therefore in the global step, and in particular in the estimation of the complexity $K(S^i|S^j)$. This term is evaluated by defining a generic Turing machine which transforms a solution vector into another solution vector. The most direct idea for such a machine is to build a naive mapping from the clusters of $\mathcal{A}^i$ to the clusters of $\mathcal{A}^j$. In supervised learning, such a mapping is direct and obtained through the supervision: The groups of same label are mapped together. In unsupervised setting, the semantics of the labels is arbitrary and does not reflect any real information: It comes that the labels of the clusters can be permuted without changing the result, and, consequently, that the operation of building a mapping is necessary and not trivial. In general, such a mapping does not have any noticeable property: in particular, it is neither injective nor surjective (see the example in Figure 14.2).

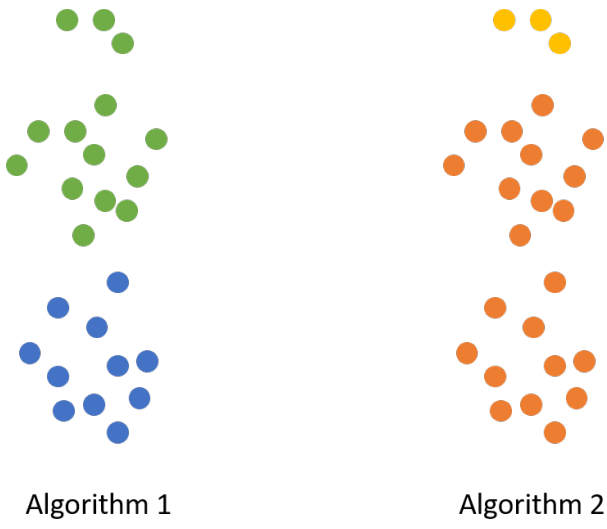


FIGURE 14.2: Illustration of a non-injective and non-surjective mapping. The colors represent the clusters. The majority mapping from $\mathcal{A}^1$ to $\mathcal{A}^2$ is not surjective since the yellow class is not the image of a class of $\mathcal{A}^1$. It is not injective either since the orange class is the image of two classes of $\mathcal{A}^1$.

The mapping we propose to use is based on a symbolic system of rules and exceptions. We define general transformation rules, which affect the whole solution

vector (for instance "Cluster 1 in $\mathcal{A}^i$ is transformed into cluster 7 in $\mathcal{A}^j$"). The general transformation is refined with exceptions, that are meant to override the rules (for instance "Instead of applying the rules, point number 42 is associated to cluster 3 in $\mathcal{A}^j$"). The principle of such a mapping is illustrated in Figure 14.3

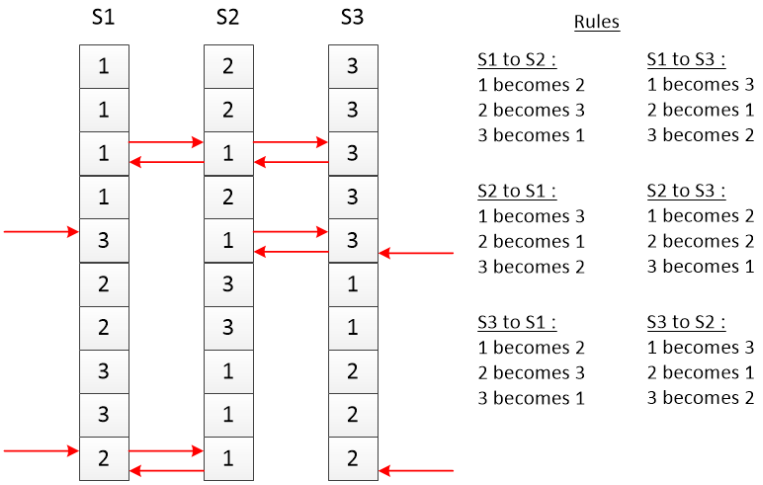


FIGURE 14.3: Illustration of a mapping between three solution vectors. The rules are given at the right of the figure. Exceptions are displayed with red arrows, the direction of which corresponds to the direction of the mapping.

More formally, we propose to encode the mapping as a key-value set

$$\langle (1, \mathcal{R}_{j,i}(1)), \ldots, (K_j, \mathcal{R}_{j,i}(K^j)) \rangle$$

(where K^j denotes the number of clusters for algorithm $\mathcal{A}^j$). The function $\mathcal{R}_{j,i}$ is called a *rule* and associates each cluster index of $\mathcal{A}^j$ into a cluster index of $\mathcal{A}^i$. For instance, $\mathcal{R}_{j,i}(3) = 1$ means that cluster 3 in $\mathcal{A}^j$ is transformed into cluster 1 in $\mathcal{A}^i$. A rule can be interpreted as the following program: `if cluster == kj: return ki` for all values of `kj`. Thus, the complexity of a rule is:

$$K\left(\mathcal{R}_{j,i}\right) = \sum_{j=1}^{K^j} \Big(K(k^j) + K(\mathcal{R}_{j,i}(k^j)) \Big) = K^j \times (\log K^j + \log K^i) \tag{14.6}$$

Such a mapping is often not sufficient to offer a full description of a transformation from one solution into another: Some exceptions have to be added to describe the exact transformation. An exception is encoded as a tuple $(n, k^i) \in \{1, \ldots, N\} \times K^i$ where n is the data index, k^i the cluster index, and N is the size of the dataset. An exception overwrites the transformation rule by stating that the n-th point is in the cluster k^i, or, in pseudo-code `if point == n: return ki`. If $\mathcal{E}_{j,i}$ designates the set of all exceptions in the mapping $S^j \to S^i$, the complexity of $\mathcal{E}_{j,i}$ is then:

$$K(\mathcal{E}_{j,i}) = \sum_{e \in \mathcal{E}_{j,i}} \Big(K(n_e) + K(S^i_{n_e}) \Big) = |\mathcal{E}_{j,i}| \times (\log N + \log K^i) \tag{14.7}$$

Consequently, merging the two complexities expressed in equations 14.6 and 14.7, we choose a machine defined in such a way that the description length $K(S^i|S^j)$

is equal to:

$$K(S^i|S^j) = K^j \times \left(\log K^j + \log K^i\right) + |\mathcal{E}_{j,i}| \times \left(\log N + \log K^i\right) \tag{14.8}$$

Based on this mapping, we propose to split the global step in two alternating operations:

1. **Solution mapping**: Mappings are found for any pair of solutions $(S^i|S^j)$
2. **Mapping optimization**: The obtained mappings are slightly corrected in order to decrease complexity.

14.3.3 Solution Mapping

In order to define the mapping in practice, we consider a majority rule: The best rules in terms of minimization of objective complexity 14.8 are the ones which minimize the number of exceptions.

To this end, we consider the confusion matrix $\Omega^{i,j}$ that maps the clusters of S^i to the clusters of S^j:

$$\Omega^{i,j} = \begin{pmatrix} \omega^{i,j}_{1,1} & \cdots & \omega^{i,j}_{1,K_j} \\ \vdots & \ddots & \vdots \\ \omega^{i,j}_{K_i,1} & \cdots & \omega^{i,j}_{K_i,K_j} \end{pmatrix} \text{ where } \omega^{i,j}_{a,b} = |S^i_a \cap S^j_b| \tag{14.9}$$

From there an *argmax* on each line of $\Omega^{i,j}$ in Equation 14.9 gives us the majority mapping rule for each cluster of $\mathcal{A}^i$ into a cluster of $\mathcal{A}^j$. Using this method, a compression is obtained by defining a general mapping transforming all labels of S^i into labels of S^j and correcting the errors afterwards. The time complexity to compute all the rules between all solutions vectors using this method is in $\mathcal{O}(N)$ for solutions vectors of length N.

This operation has to be repeated for each pair of solutions $(S^i|S^j)$, hence the time complexity of the solution mapping step is $\mathcal{O}(N \times J^2)$. Afterwards, exceptions can be obtained easily (in linear time complexity). The complete algorithm is detailed in Algorithm 5.

Algorithm 5: SOLUTIONMAPPING

Input: A set of J clustering solutions S
Output: A set of rules $\{\mathcal{R}_{j,i}\}_{1\leq i,j\leq J}$ and exceptions $\{\mathcal{E}\}_{1\leq i,j\leq J}$
for $i = 1 \dots J$ **do**
  for $j = 1 \dots J$ **do**
    Compute $\Omega^{i,j}$
    for $k = 1 \dots K^i$ **do**
      $\mathcal{R}_{j,i}[k] \leftarrow \arg\max_l \Omega^{i,j}_{k,l}$
    for $n = 1 \dots N$ **do**
      if $\mathcal{R}_{j,i}\left[S^j[n]\right] \neq S^i[n]$ **then** $\mathcal{E}_{j,i}[n] \leftarrow S^i[n]$
return $\{\mathcal{R}_{j,i}\}_{1\leq i,j\leq J}, \{\mathcal{E}_{j,i}\}_{1\leq i,j\leq J}$

14.3.4 Mapping Optimization

The mapping optimization step is based on a very simple idea: Optimizing Equation 14.1 consists in searching for errors the correction of which would have the most positive impact on the collaborative term $\sum_{j \neq i} K(S^i|S^j)$ with a minimal impact on the local term $K(X^i|M^i)$. Corrections that do not improve the collaborative term or have a negative impact are ignored.

The mapping optimization is the most complex step of the method. It consists in removing exceptions one by one in the obtained set $\{\mathcal{E}_{j,i}\}_{1 \leq i,j \leq J}$. Removing an exception results in a single change inside a clustering solution. The system decides to remove an exception if this deletion leads to a reduction in complexity. Because a deletion modifies the solutions, the deletion order has importance in this algorithm. This issue is also encountered in SAMARAH method (Wemmert and Gançarski, 2002).

The key idea we rely on in order to solve the problem is the independence hypothesis on data points. Considering that all data points are described independently, the mapping optimization step can be done on all data points in parallel. It consists in removing exceptions one by one until no exception removal makes the complexity decrease. A recursive approach has been chosen to determine a solution for one data with fixed rules. The proposed algorithm, exposed in Algorithm 6, tries to remove exceptions one by one in a backtracking process. The advantage of backtracking is that it gives an exact solution.

Algorithm 6: MAPPINGOPTIMIZATION

Input: Multi-view solution vector for one point: $s = (s^1, \ldots, s^J)$, Rules $(\mathcal{R}_{i,j})_{i,j}$
Output: Refined solution vector s, Associated complexity

```
E ← {}
for j = 1 ... J do
    for i = 1 ... J do
        if s[i] ≠ R_{j,i}(j) then
            E ← E ∪ {(j, i)}
K ← ComputeComplexity(s)
for (j, i) ∈ E do
    s' ← s
    s'[i] ← R_{j,i}(j)
    s', K' ← MappingOptimization(s, (R_{i,j})_{i,j})
    if K < K' then
        s ← s'
        K ← K'
return s, K
```

Such as presented here, the algorithm might fail in solving some problems in an expected way. Consider for instance a problem with $J = 4$ views on the data and where the clustering algorithms are designed with the same number of clusters and such that all the rules are identity mapping: $\mathcal{R}_{i,j}(k) = k$ for all i, j and k. We also consider that the local complexity term is constant for all algorithms. This situation correspond to a simple consensus. In this case, if one data point is associated with the four clustering solutions $s = (0, 0, 0, 1)$, the expected refined solution is $s' = (0, 0, 0, 0)$. However, with Algorithm 6, the obtained result would

be $s' = (1,1,1,1)$ which has the same complexity with the imposed assumptions. This example raises the question of discriminating between several different solutions of same complexity. In practice, we added a constraint on the depth of the solution in the backtracking tree. When two solutions are equal in complexity, the modified algorithm selects the solution of minimal depth in the backtracking tree (or, equivalently, the solution with the minimal number of corrections).

14.3.5 Dealing with Sparsity

One advantage of the rule-based representation of the mapping is that it does not require a full knowledge of the data. Since the construction of the rules, as exposed in Algorithm 5, relies on a majority vote only, it is not required that all data points are associated to a cluster. Missing values can be inferred afterwards using the estimated rules.

We consider one missing view j for one data point and we denote by x the multi-view representation of the considered point (hence the value of x^j is missing). The problem consists in associating the value of the solution $s^j \in \{1,\dots,K^j\}$. Since the view x^j is empty, the local complexity term for algorithm $\mathcal{A}^j$ is constant and the problem consists in minimizing over s^j the collaborative term:

$$s^j = \arg\min_{s^j} \sum_{i \neq j} \Big(K(s^i|s^j) + K(s^j|s^i) \Big) \tag{14.10}$$

where the conditional complexity $K(s^i|s^j)$ is equal to $\mathbb{I}(s^i \neq \mathcal{R}_{j,i}(s^j))$. Finding the optimal value can be done with a naive algorithm testing all the possible solutions. The complexity of such an algorithm is linear in K^j, the number of clusters of algorithm $\mathcal{A}^j$, and in J, the total number of algorithms: $\mathcal{O}(K^j \times J)$. We notice the similarity between the problem described in 14.10 and the median-partition problem described in Equation 13.5.

When more than one view is missing, the optimization problem becomes more complex, since inter-dependencies between the estimated values appear. The brute force approach consists then in testing all possible combinations of solutions, which is computationally too expensive. Two simplified strategies are then possible:

1. **Total independence:** Filling in parallel all missing values, considering only observed values. The sum in Equation 14.10 is over observed data only.

2. **Ordered filling:** Filling the missing values in a pre-determined order, considering then all previously filled values for the minimization of Equation 14.10.

None of these two strategies is guaranteed to converge to the global optimum.

The question of the order is crucial for the second strategy: If it is not well-chosen, the algorithm might converge to sub-optimal solutions. The question is difficult though: Which missing data should be filled first? Apart from random order, which is clearly sub-optimal, we can imagine two main strategies: either filling the least ambiguous missing values first (ie. values which bring the highest consensus) or filling the most ambiguous ones first. Further analysis is required to answer this question, that has not been done in the scope of this thesis. In the experimental section, we worked with the total independence hypothesis.

14.4 Experimental Validation

In this section, we present experimental results obtained with the algorithm presented above.

14.4.1 Datasets

In this section, we propose an applicative setting in which we used our proposed method on various multi-view data sets, real and artificial.

We considered the following data sets:

- The Wisconsin Data Breast Cancer (UCI): this data set contains 569 instances with 30 parameters and 2 classes. These 30 parameters contain 10 descriptors for 3 different cells (10 each) of the same patient. This data set can easily be split into 3 views: one for each cell.
- The Spam Base data set (UCI): The Spam Base data set contains 4601 observations described by 57 attributes and a label column: Spam or not Spam (1 or 0). The different attributes can be split into views containing word frequencies, letter frequencies and capital run sequences.
- The VHR Strasbourg data set (Rougier and Puissant, 2014): it contains the description of 187058 segments extracted from a very high resolution satellite image of the French city of Strasbourg. Each segment is described by 27 attributes that can be split between radiometical attributes, shape attributes, and texture attributes. Furthermore, the color attributes can also be split between red, blue and near-infrared attributes. The data set is provided with a partial hybrid ground-truth containing 15 expert classes.
- The Battalia3 data set (artificial): Battalia3 is an artificial dataset created using the exoplanet random generator from the online game Battalia.fr; This data set describes 2000 randomly generated exoplanets with 27 numerical attributes and their associated class (6 classes). The attributes can be split between system and orbital parameters (7 attributes), planet characteristics (10 attributes) and atmospheric characteristics (10 attributes).
- The “MV2" data set (artificial): a data set created specifically to test this kind of algorithm. It features 2000 randomly generated data, split into 4 views of 6 attributes each, and a total of 4 classes. All attributes were generated either from Gaussian distributions with parameters linked to the matching class, or are random noise, or are linear combinations of other attributes.

Dataset	Size	Attributes	Views
WDBC	569	30	3
SpamBase	4601	57	3
VHR Strasbourg	187058	27	3
Battalia3	2000	27	3
MV2	2000	24	4

TABLE 14.1: Dataset characteristics.

14.4.2 Experimental Results

To assess the effectiveness of our proposed method, in this section we propose an experiment in which we compare it with four other collaborative and multi-view methods from the literature: the entropy based collaborative clustering (EBCC) (Sublime, Matei, Cabanes, Grozavu, Bennani, and Cornuéjols, 2017), a re-implementation of the multi-view EM algorithm (Bickel and Scheffer, 2005), the collaborative GTM algorithm (Ghassany, Grozavu, and Bennani, 2012a) and the collaborative SOM algorithm (Nistor Grozavu, 2009). For fairness purposes, with collaborative GTM, collaborative SOM and MV-EM all being based on Gaussian Mixture models, we used both our proposed method and the EBCC algorithm with GMM clustering algorithms as well.

The 3 methods are compared using two unsupervised indexes: the Davies-Bouldin index (DBI) and the Silhouette index (Sil.), both of which assess in different ways the quality of the cluster in terms of compacity and whether or not they are well separated. The Davies-Bouldin index is a positive not normalized index the value of which is better when it is lower. The Silhouette index is a normalized index which takes values between -1 and 1, 1 being the best possible value.

Furthermore, since all data sets were acquired from originally supervised problems, they were all provided with available labels. Consequently, in our experiments, we also used the Rand Index based on the original classes as an external index.

Dataset	**Our Model**		MV-EM		EBCC		GTM^{col}		SOM^{col}	
	DBI	Sil.	DBI	Sil.	DBI	Sil.	DBI	Sil.	DBI	Sil.
WDBC	**0.98**	**0.55**	1.63	0.42	1.63	0.42	1.8	0.37	1.68	0.41
SpamBase	**3.08**	**0.19**	4.77	0.086	4.73	0.085	4.60	0.093	4.35	0.113
VHR Strasbourg	3.46	0.14	3.21	0.12	**2.89**	**0.175**	-	-	-	-
Battalia3	**2.29**	0.34	2.43	0.16	2.83	0.14	2.68	**0.35**	2.51	0.25
MV2	1.61	0.37	**1.34**	0.35	**1.34**	0.35	1.61	0.38	1.44	**0.39**

TABLE 14.2: Experimental results: raw average results on unsupervised indexes.

Dataset / Rand	**Our Model**	MV-EM	EBCC	GTM^{col}	SOM^{col}
WDBC	0.95	0.79	0.87	0.96	**0.97**
SpamBase	0.76	0.74	**0.86**	0.83	0.84
VHR Strasbourg	**0.78**	0.73	0.75	-	-
Battalia3	**0.86**	0.78	0.80	0.78	0.79
MV2	**0.93**	**0.93**	**0.93**	0.90	0.90

TABLE 14.3: Experimental results: raw average results on the Rand Index.

For VHR Strasbourg dataset, the runtime (without the initial local clusterings) was less than one hour with parallel computing, a couple hours otherwise. For other data the runtime ranged from less than one second to 2-3 minutes for larger data sets.

In Table 14.2, we show the average results achieved on the unsupervised indexes at the end for the multi-view or collaborative process. The results for the supervised indexes (Rand index) are shown in Table 14.3. Both the Davies-Bouldin index and the Silhouette index where computed using the partitions found on the local views

and the complete data as reference. The absence of results for both collaborative GTM and SOM algorithms for the VHR Strasbourg dataset is due to the fact that neither of these algorithms was able to provide a result in a reasonable amount of time.

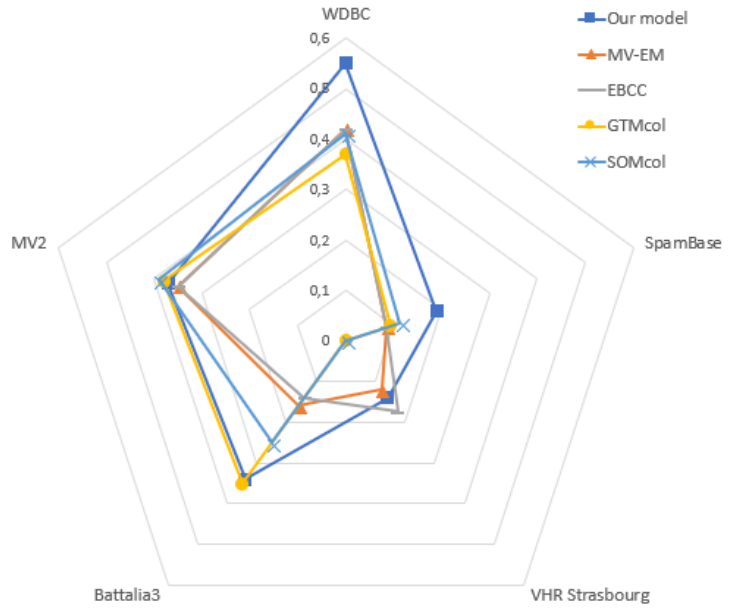
In Figure 14.4, we show a radar map made from the Silhouette and Rand Index tables. As one can see from the figure, our method overall outperforms the other algorithms with a much larger area coverage and we still achieve close to state of the art results with datasets for which our method is not the best one. Without surprises, the older MV-EM algorithm has the overall worst performances, followed by Kohonen maps based collaborative algorithms and then the more recent Entropy based collaborative Framework (EBCC) which sometimes has better results than our proposed method albeit with a smaller coverage area in both supervised and unsupervised indexes. Furthermore, unlike the collaborative SOM and GTM algorithms, our method does scale to relatively large dataset like VHR Strasbourg. We would like to point out that scaling is not an issue here, neither in terms of number of data nor in terms of number of features. As explained in the paper, each data can be treated separately, so a parallel run can be done. Moreover, time complexity depends on the local complexities (which are, in general, linear in the number of features). These results highlight the strength of our method, and come to back up its strong theoretical background, compared with the other competitors, with good experimental performance.

14.5 Conclusion

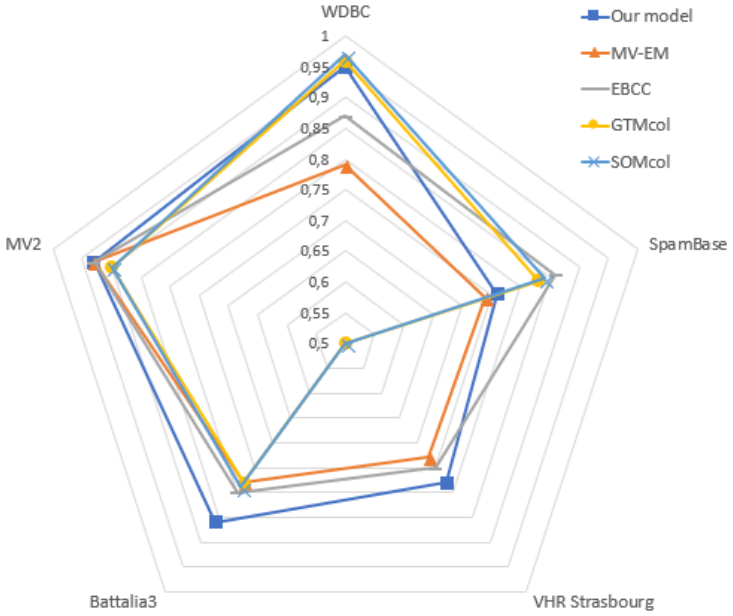
In this chapter, we have proposed a new perspective on the problem of multi-source clustering by showing that it can be reduced to a model selection with the MDL principle. Compared to state of the art methods, our methodology is based on a theoretical background and does not rely on heuristics, but leads to a very similar minimization problem. Besides, its strength is highlighted by excellent experimental results both for artificial and real data, with a naive and parameter-free algorithm.

The study proposed here is just one of the various approaches to the problem. First, the properties of the designed algorithms have to be investigated from a theoretical point of view, in particular in the direction of stability. In addition, our focus was on collaborative clustering but an adaptation of our method to unsupervised ensemble learning (finding consensus) comes directly. Finally, a study of the applications of sparse multi-source clustering can be an interesting perspective to this work.

In the next chapter, we present a general discussion on the possibility of multi-source clustering. In particular, we will discuss in more details the question of stability and will provide a couple of initial properties.



(A) Silhouette index



(B) Rand Index

FIGURE 14.4: Radar maps for Silhouette and Rand Index on the datasets of interest.

Chapter 15

Can clustering algorithms collaborate?

In Chapter 13, we introduced the problem of multi-source clustering from the applicative point of view and exposed it as a problem encountered in real-life situation. Although the method presented in Chapter 14 shows very good results on a couple of multi-source datasets, a fundamental perspective on multi-source clustering is still needed.

The purpose of this chapter is to introduce some ideas on the possibility of a collaboration in clustering. We will discuss a couple of questions relative to collaboration, with a main focus on the fundamental question of multi-source clustering: Can clustering algorithms collaborate? Before answering this question, we will have to discuss its actual meaning and the various ways it can be answered.

The remainder of this chapter is organized as follows: In a first section, we propose a general discussion on the idea of collaboration in unsupervised learning. Based on this discussion, we propose a first attempt of an answer in Section 15.2, by studying the way to define the "best" collaborators. Finally, in Section 15.3, we interpret collaboration from the point of view of stability, both from a theoretical and an applied point of view.

This chapter is an opening chapter which aims to discuss the very nature of collaboration in unsupervised learning. Consequently, it is different from the previous chapters and, in particular, will not refer to complexity. It has to be seen as an opening to other research perspectives relative to the transfer of knowledge.

15.1 Collaboration: A Difficult Concept in the Absence of Supervision

The general idea of collaboration (or, in general, of *ensemble learning*) is inspired by the supervised setting where ensemble methods, such as *Bootstrap aggregating* (or *bagging*), are commonly used to enhance the quality of prediction for the learned classifiers. Behind the ensemble techniques lies the idea that a group of predictors must be better than individual ones. This assumption has been observed in psychology since the seminal research of (Galton, 1907) which shows that, when predicting the weight of an ox, the average precision obtained by a crowd is better than the individual precision of all individuals in the crowd. This phenomenon of collective wisdom is presented in more details in (Surowiecki, 2004). However, is there any fundamental argument that justifies this phenomenon? Is it legit to extend it to machine learning?

Three main arguments are given in (Dietterich, 2000) to justify the success of ensemble methods:

1. **Statistical argument:** When there is not enough points in the training set, the set of hypotheses giving good accuracy on the data is large. Ensemble learning is a way to select a good classifier in this large hypothesis set, averaging incorrect decisions.

2. **Computational argument:** Many objective functions in machine learning are actually not convex and the optimization scheme tends to be stuck in local minima. Ensemble avoids this by combining the decisions taken by initiating the optimization from various starting points.

3. **Representational argument:** It happens that the true function is not an element of the hypothesis space. In such cases, ensemble is a way to explore functions that do not necessary belong to the hypothesis space of the algorithm.

These informal arguments apply to the case of unsupervised ensemble learning as well. As presented in previous chapters, collaboration is supposed to compensate for the "errors" done by local clustering methods. The real problem in the adaptation of the ideas of collaboration to unsupervised learning is the absence of notion of "good prediction" or "bad prediction".

It is well-known that the quality of the collaborator is important in supervised ensemble learning. If the collaborators are worse than random classifiers, the collaboration does not lead to a global improvement of the collaboration.

In unsupervised ensemble learning, there is no way to evaluate the quality of a clustering. Since the clusters are not labeled, the absence of semantic prevents from defining errors and, as a consequence, misclassification rate or risk. From there, there is *a priori* no guarantee at all that collaborators engaged in collaboration will provide a better clustering together than independently.

The notion of "good" clustering is also complex and requires to come back to the fundamental description of clustering. In this thesis, we have claimed that clustering is a descriptive task and that a good description is a short description. An attempt has been proposed by (Kleinberg, 2003) to define good clustering: The authors propose three theoretical criteria that a good clustering should satisfy (scale-invariance, richness and consistency). However, it is shown that no clustering method can satisfy these three criteria together.

Another attempt to define the quality of clustering is the approach of stability developed by (Ben-David, Von Luxburg, and Pál, 2006). This theory, which proposes an adaptation of PAC learning to the unsupervised case, defines the stability of a clustering algorithm. Stability is a notion that is shared by supervised and unsupervised worlds and thus can be an interesting candidate to assess the quality of a collaboration.

15.2 Selecting the Best Collaborators

In this section, we propose to find a way to discover the best collaborators in collaborative clustering in a context where the objective problem is of the form:

$$\mathbf{S}^* = \arg\min_{S} \sum_{j=1}^{J} \mathcal{L}(X^j, S^j) - \sum_{i \neq j} \tau_{i,j} \Delta(S^i, S^j) \tag{15.1}$$

hence the sum of local terms $\mathcal{L}$ and of a collaborative penalty Δ. The results presented in this section are adapted from the results published in (Sublime, Matei, and Murena, 2017) (mainly Sections 3 and 4).

15.2.1 Introducing the problem

As presented in Chapter 13, many collaborative clustering problems share a same methodology. They aim to find the solutions $\mathbf{S} = (S^1, \dots, S^J)$ that maximize an objective function of the form:

$$\sum_{j=1}^{J} \mathcal{L}(X^j, S^j) - \sum_{i \neq j} \tau_{i,j} \Delta(S^i, S^j) \tag{15.2}$$

The term $L(X^j, S^j)$ measures the local score of a solution S^j for data S^j, hence a quality of a solution in the description of data. The term $\Delta(S^i, S^j)$ corresponds to a dissimilarity between two solutions and measures the quality of the collaboration. The coefficients $\tau_{i,j}$ measure the weight of a collaboration between algorithm $\mathcal{A}^i$ and algorithm $\mathcal{A}^j$.

In most techniques (including in the complexity-based objective given in Equation 14.5), the weights are chosen to be uniform, which implies that all collaborators have the same impact. However, we can wonder how optimally these coefficients can be chosen in order to maximize the score of the algorithm.

Including this idea leads to the maximization of Equation 15.2 over both the solutions $\mathbf{S}$ and the coefficients $(\tau_{i,j})$. This problem can be solved by alternating maximization over the solutions and the coefficients. General algorithms focus only on the first step (maximizing with respect to the solutions), but we propose to consider here the second problem. Since the local term does not involve the coefficients, the problem can be reduced to:

$$\underset{\tau}{\text{minimize}} \quad \sum_{j=1}^{J} \sum_{i \neq j} \tau_{i,j} \cdot \Delta(S^i, S^j) \tag{15.3}$$

If the only constraint on the coefficients $\tau_{i,j}$ is their non-negativity, and assuming that $\Delta \geq 0$, the solution is necessarily $\tau_{i,j} = 0$. This case is trivial and corresponds to the complete absence of collaboration. To overcome this problem, we propose a normalization constraint over the coefficients. The constraint, given with parameter $p \in \mathbb{N}^*$, is the following:

$$\forall i \quad \sum_{j \neq i}^{J} (\tau_{j,i})^p = 1, \qquad p \in \mathbb{N}^* \tag{15.4}$$

15.2.2 Optimizing the Collaboration

We propose to solve the following optimization system: given the $\Delta(S^i, S^j) \geq 0$ and $p \in \mathbb{N}^*$, we are trying to find the matrix $T = \{\tau_{i,j}\}_{J \times J}$ solving the following

optimization problem:

$$\begin{aligned} \underset{T}{\text{minimize}} \quad & \sum_{j=1}^{J} \sum_{i \neq j} \tau_{i,j} \cdot \Delta(S^i, S^j) \\ \text{subject to} \quad & \sum_{i \neq j} (\tau_{i,j})^p = 1, \qquad \forall j, \\ & \tau_{i,j} \geq 0 \qquad \forall (i,j). \end{aligned} \tag{15.5}$$

The solution of this problem for $p > 1$ is given in the following proposition.

Proposition 11. *Any solution of system 15.5 for $p > 1$ verifies:*

$$\forall j, \forall i \notin arg\min_{k \neq j} \Delta(S^k, S^j), \quad \tau_{j,i} = 0 \tag{15.6}$$

Proof. We solve this problem by considering the Karush–Kuhn–Tucker (KKT) conditions (Kuhn and Tucker, 1951). The five conditions form the following system:

$$\forall (i,j), i \neq j \begin{cases} (1) & \tau_{i,j} \geq 0 \qquad \text{(primal feasibility)} \\ (2) & \sum_{i \neq j}^{J} (\tau_{i,j}) = 1 \qquad \text{(primal feasibility)} \\ (3) & \lambda_{i,j} \geq 0 \qquad \text{(dual feasibility)} \\ (4) & \tau_{i,j} \cdot \lambda_{i,j} = 0 \qquad \text{(complementary slackness)} \\ (5) & \Delta(S^i, S^j) - \lambda_{i,j} + \nu_j = 0 \qquad \text{(stationarity)} \end{cases} \tag{15.7}$$

We fix j. Consider k_j such that $\tau_{k_j,j} > 0$ (such a k_j necessarily exists, from the primal feasibility condition). Complementary slackness imposes that $\lambda_{k_j,j} = 0$ and thus:

$$\nu_j = -\Delta(S^{k_j}, S^j) \tag{15.8}$$

For other collaborators $i \neq k_j$, two cases are possible: either $\tau_{i,j} > 0$ or $\tau_{i,j} = 0$ (the case $\tau_{i,j} < 0$ is discarded by the primal feasibility).
Case 1: $\tau_{i,j} > 0$. In this case, we can use the stationarity condition in the same way as done for k. We obtain that $\nu_j = -\Delta(S^i, S^j)$. Using Equation 15.8, we have that all positive coefficients correspond to views i that have the same dissimilarity value $\Delta(S^i, S^j)$ with view j.
Case 2: $\tau_{i,j} = 0$. In this case, the stationarity condition gives the following value for $\lambda_{i,j}$:

$$\lambda_{i,j} = \nu_j + \Delta(S^i, S^j) = \Delta(S^i, S^j) - \Delta(S^{k_j}, S^j)$$

Since $\lambda_{i,j} \geq 0$ (feasibility on the dual), we have the condition $\Delta(S^i, S^j) > \Delta(S^{k_j}, S^j)$, which means that k_j minimizes the dissimilarity value. □

We notice in this result that only views with minimal dissimilarity can collaborate. However, if several views i have the same dissimilarity with a view j, their coefficients are not necessarily uniform. Any possible weighting satisfying condition 15.6 is a solution of the problem 15.5. The corresponding minimal value is equal to $\sum_{j=1}^{J} \min_{k \neq j} \Delta(S^k, S^j)$.

The summary of this proposition is the following: In the context of collaborative clustering, the results should be better if each individual algorithm collaborates only with the algorithm that has the most similar solution. If several algorithms have the same most similar solution, they can be given any weight. However, this result

has to be taken with care: it depends on the choice of the regularization. Here, we considered that the coefficients must sum to 1, but other feasibility restrictions could be considered.

15.2.3 Discussion

These results are interesting because they go against the common idea that collaboration works best between collaborators having an average diversity (Grozavu, Cabanes, and Bennani, 2014; Rastin, Cabanes, Grozavu, and Bennani, 2015). Indeed, common sense would want us to think that a low diversity means not much room for improvement since everyone agrees, and a high diversity not enough common ground to reach an agreement, thus making average diversity the best case scenario.

However it is our opinion that this interpretation carries the bias of supervised learning. If we think about the goal of collaboration in the context of unsupervised learning, these mathematical results make sense: We are in a situation where each algorithm does an exploratory task and has no supervised index to rely on to guess quality of its solution. Therefore, when several algorithms find solutions that are similar, it is quite likely that they have actually found a structure in the data. As a consequence collaborating with algorithms that have solutions similar to the local partitioning is a convenient way to avoid the risk of negative collaboration. There are actually good reasons not to collaborate with an algorithm the results of which are too different from the local partition: Such collaborators may be in a feature space where the clusters to be found are completely different even for the same objects. The dissimilarity of a solution with all others may simply mean that this solution is a poor one.

These results can also be linked to recent works on clustering stability (Ben-David, Von Luxburg, and Pál, 2006). A clustering is said to be stable if the partition remains similar when the data set or the clustering process are perturbed. In the context of collaborative clustering, the perturbations would be that (1) we observe the same data in different feature spaces, and (2) we use different algorithms. With our proposed weighting methods, the algorithms with the strongest influence will be these with solutions most often similar to the other algorithms' solutions. It matches with the definition of stability: Such solutions that highlight common structures and clusters through several feature spaces with different algorithms are the most stable. The problem of stability in collaborative clustering will be discussed in more details in next section.

15.3 Stability of Collaborative Clustering

In this section, we introduce the problem of stability in collaborative clustering. The presented results are prospective on-going works.

15.3.1 Reminder: Clustering Stability

Before we propose our analysis of stability in collaborative clustering, we propose to introduce the original definitions proposed for classical clustering stability. This section exposes the notions introduced in (Ben-David, Von Luxburg, and Pál, 2006).

We consider a data space $\mathbb{X}$ endowed with probability measure P. If $\mathbb{X}$ happens to be a metric space, we denote by ℓ its metric. A sample $S = \{x_1, \ldots, x_m\}$ is drawn i.i.d from $(\mathbb{X}, P)$.

A clustering $\mathcal{C}$ of a subset $X \subseteq \mathbb{X}$ is a function $\mathcal{C} : X \to \mathbb{N}$ which to any data subset $X \subseteq \mathbb{X}$ associates a solution vector in the form of matching clusters: $S = \mathcal{C}(X)$. The clusters are defined by $\mathcal{C}_i = \mathcal{C}^{-1}(\{i\}) = \{x \in X; \mathcal{C}(x) = i\}$. A clustering algorithm $\mathcal{A}$ is a function that computes a clustering of X for any given finite sample $\mathcal{S} \subseteq X$.

The proposed definition of clustering is very different from the approaches followed until now. In previous approaches (and in particular in Chapter 14), we considered that clustering regards partitioning one precise dataset without any aim to generalize. In the described approach, a clustering is a partitioning of the entire data space and not only of the observed dataset. The space can be chosen to be the dataset, but this trivial case is not interesting in the theory of stability.

A large class of clustering algorithms choose the clustering by optimizing some risk function. The large class of centroid based algorithms falls into this category, and spectral clustering can also be interpreted in this way as well.

Definition 16 (Risk optimization scheme)**.** *A risk optimization scheme is defined by a quadruple $(\mathbb{X}, \Sigma, \mathcal{P}, R)$, where $\mathbb{X}$ is some domain set, Σ is a set of legal clusterings of $\mathcal{X}$, and $\mathcal{P}$ is a set of probability distributions over $\mathbb{X}$, and $R : \mathcal{P} \times \Sigma \to [0, \infty)$ is an objective function (or risk) that the clustering algorithm aims to minimize.*

Denote $opt(P) := \inf_{\mathcal{C} \in \Sigma} R(P, \mathcal{C})$. For a sample $X \subseteq \mathbb{X}$, we call $R(P_X, \mathcal{C})$ the empirical risk of $\mathcal{C}$, where P_X is the uniform probability distribution over S. A clustering algorithm $\mathcal{A}$ is called R-minimizing, if $R(P_X, \mathcal{A}(X)) = opt(P_X)$, for any sample X.

In the context of this theoretical analysis, we aim to compare different clustering solutions. For this purpose, we define clustering distances.

Definition 17 (Clustering distance)**.** *Let $\mathcal{P}$ be family of probability distributions over some domain $\mathbb{X}$. Let Σ be a family of clusterings of $\mathbb{X}$. A clustering distance is function $d : \mathcal{P} \times \Sigma \times \Sigma \to [0, 1]$ satisfying for any $P \in \mathcal{P}$ and any $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3 \in \Sigma$:*

1. $d_P(\mathcal{C}_1, \mathcal{C}_1) = 0$
2. $d_P(\mathcal{C}_1, \mathcal{C}_2) = d_P(\mathcal{C}_2, \mathcal{C}_1)$ *(symmetry)*
3. $d_P(\mathcal{C}_1, \mathcal{C}_3) \leq d_P(\mathcal{C}_1, \mathcal{C}_2) + d_P(\mathcal{C}_2, \mathcal{C}_3)$ *(triangle inequality)*

We do not require that a clustering distance satisfies the implication $d_P(\mathcal{C}_1, \mathcal{C}_2) = 0 \Rightarrow \mathcal{C}_1 = \mathcal{C}_2$.

Stability measures how a perturbation in data affects the result of a clustering algorithm. It is possible to define the **stability** of an algorithm $\mathcal{A}$ for a sample size m with respect to a probability distribution P as follows:

Definition 18 (Stability)**.** *Let P be a probability distribution over $\mathcal{X}$. Let d be a clustering distance. Let $\mathcal{A}$ be a clustering algorithm. The stability of the algorithm $\mathcal{A}$ for the sample size m with respect to the probability distribution P is*

$$stab(\mathcal{A}, P, m) = \mathop{\mathbb{E}}_{\substack{X_1 \sim P^m \\ X_2 \sim P^m}} [d_P(\mathcal{A}(X_1), \mathcal{A}(X_2))] \tag{15.9}$$

The stability of the algorithm $\mathcal{A}$ with respect to the probability distribution P is

$$stab(\mathcal{A}, P) = \limsup_{m \to \infty} stab(\mathcal{A}, P, m)$$

We say that algorithm $\mathcal{A}$ is stable for P, if $stab(\mathcal{A}, P) = 0$.

15.3.2 Definitions: Collaborative Clustering

In the context of collaborative clustering, we consider that the total space $\mathbb{X}$ can be decomposed into the product $\mathbb{X}^1 \times \ldots \times \mathbb{X}^J$ of J view spaces $\mathbb{X}^j$.

Definition 19 (Global clustering)**.** *A global clustering is defined as a combination of local clustering in the following sense: A global clustering $\mathcal{C}$ of the subset $\mathcal{X} \subseteq \mathbb{X}$ is a function $\mathcal{C} : \mathcal{X} \to \mathbb{N}^J$. The i-th local cluster for view j, denoted $\mathcal{C}_i^j$, is defined as:*

$$\mathcal{C}_i^j = \{x \in \mathcal{X}; \left(\mathcal{C}(x)\right)^j = i\} \subseteq \mathbb{X} \tag{15.10}$$

A collaborative clustering algorithm $\mathcal{A} = \langle \mathcal{A}^1, \ldots, \mathcal{A}^J \rangle$ is a function which computes a global clustering based on local clustering algorithms $\mathcal{C}^j$ on $\mathbb{X}^j$. More formally, if we denote by $\mathbb{A}^j$ the set of clustering algorithms on $\mathbb{X}^j$, $\mathcal{C}$ the set of global clusterings on $\mathcal{X} \subseteq \mathbb{X}$ and Σ the set of finite partitions of $\mathcal{X}$, a collaborative clustering algorithm is defined as a mapping $\mathbb{A}^1 \times \ldots \times \mathbb{A}^J \times \Sigma \to \mathcal{C}$.

In general, the projection of the clustering obtained by a collaborative algorithm onto one of the views j is distinct of the local clustering obtained by the local algorithm $\mathcal{A}^j$: If $\mathcal{C} = \mathcal{A}(X)$, then in general $\mathcal{C}^j \neq \mathcal{A}^j(X^j)$. We define a very particular case of collaborative clustering algorithms for which this general property does not hold.

Definition 20 (Concatenation of local clustering algorithms)**.** *The concatenation of local clustering algorithms $\mathcal{A}^1$ to $\mathcal{A}^J$, denoted by $\bigoplus_{j=1}^J \mathcal{A}^j$ is defined as follows: If $\mathcal{C}$ is the global clustering induced by $\mathcal{A} = \bigoplus_{j=1}^J \mathcal{A}^j$ on a data set X, then*

$$\forall x \in \mathbb{X}, \forall j \in \{1, \ldots, J\}, \qquad \mathcal{C}^j(x^j) = \left(\mathcal{A}^j(X^j)\right)(x^j) \tag{15.11}$$

This definition means that the local clustering on each view j obtained with the collaborative algorithm is exactly the same as the local clustering obtained with the local algorithm $\mathcal{A}^j$ only.

Since $\mathbb{N}^J$ is isomorphic to $\mathbb{N}$, a global clustering can be interpreted as a clustering of $X \subseteq \mathbb{X}$. Consider the isomorphism $\nu_J : \mathbb{N}^J \to \mathbb{N}$ (which will be denoted by ν when the value J is obvious in the context). Then the mapping $\nu \circ \mathcal{C}$ is a a clustering of $X \subseteq \mathbb{X}$.

Using this equivalence, the notions of risk optimization scheme and clustering distance hold for global clustering.

Proposition 12. *Let $\mathbb{X} = \mathbb{X}^1 \times \ldots \times \mathbb{X}^J$ be a domain and d^j clustering distances on $\mathbb{X}^j$. We define the function $d : \mathcal{P} \times \mathcal{S} \times \mathcal{S} \to [0, 1]$ such that $d_P(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{J} \sum_{j=1}^J d_{P_j}^j(\mathcal{C}_1^j, \mathcal{C}_2^j)$. Then d defines a clustering distance on $\mathcal{X}$. We call it the canonical collaborative clustering distance.*

Proof. The clustering distance properties follow from the linearity in terms of d^j and from the properties of the local clustering distances. □

15.3.3 Stability of Collaborative Clustering

As noticed above, a global clustering can be interpreted as a standard clustering. Hence, the definition of stability given in the standard case can be extended to the collaborative case.

Proposition 13 above shows a direct adaptation of Ben-David's key theorem on clustering stability (Theorem 10 in (Ben-David, Von Luxburg, and Pál, 2006)) to collaborative clustering.

Proposition 13. *If P has a unique minimizer $\mathcal{C}^*$ for risk R, then any R-minimizing collaborative clustering algorithm which is risk converging is stable on P.*

Proof. Let $\mathcal{A}$ be a collaborative clustering algorithm on $\mathbb{X} = \mathbb{X}^1 \times \ldots \times \mathbb{X}^J$.

Consider an isomorphism $\nu : \mathbb{N}^J \to \mathbb{N}$. Based on collaborative algorithm $\mathcal{A}$, one can build a clustering algorithm $\tilde{\mathcal{A}}$ such that the clustering $\tilde{\mathcal{C}}$ induced by sample S for $\tilde{\mathcal{A}}$ is such that $\tilde{\mathcal{C}} = \nu \circ (\mathcal{A}(S))$. For simplicity purpose, we will denote this algorithm $\tilde{\mathcal{A}} = \nu \circ \mathcal{A}$. We call d_P the global clustering distance and $\tilde{d}_P$ its associated local distance such that $\tilde{d}_P(\tilde{\mathcal{C}}_1, \tilde{\mathcal{C}}_2) = d_P(\nu^{-1} \circ \tilde{\mathcal{C}}_1, \nu^{-1} \circ \tilde{\mathcal{C}}_2)$.

Using these two clustering distances in Equation 15.9, the following lemma is straightforward:

Lemma 1. *If $\tilde{\mathcal{A}} = \nu \circ \mathcal{A}$ is stable (for distance $\tilde{d}_P$), then $\mathcal{A}$ is stable (for distance d_P).*

If $\mathcal{A}$ is R-minimizing, then $\tilde{\mathcal{A}}$ is $\tilde{R}$-minimizing with $\tilde{R}(P, \tilde{\mathcal{C}}) = R(P, \nu^{-1} \circ \tilde{\mathcal{C}})$. It is direct that $opt_{\tilde{R}}(P) = opt_R(P)$ and that $\tilde{\mathcal{A}}$ is risk-converging. It is also direct that P has a unique minimizer $\tilde{\mathcal{C}}^*$ associated to $\tilde{R}$.

Combining all the previous results together, we have that $\tilde{\mathcal{A}}$ is $\tilde{R}$-minimizing and risk converging. Since P has a unique minimizer for $\tilde{R}$, it follows from (Ben-David, Von Luxburg, and Pál, 2006) that $\tilde{\mathcal{A}}$ is stable. Lemma 1 guarantees the result. □

The idea of the proof is simply to build a clustering algorithm from the collaborative clustering algorithm based on the isomorphism ν. From this proposition, it comes that collaborative clustering algorithms can be treated exactly the same way as standard clustering algorithms when it comes to stability analysis.

A first result can be shown about the concatenation of clustering algorithms. Proposition 14 states that a concatenation of local algorithms is stable provided that the local algorithms are stable.

Proposition 14. *Suppose that the local algorithms $\mathcal{A}^j$ are stable for distance $d^j_{P^j}$. Then the collaborative algorithm $\mathcal{A} = \bigoplus_{j=1}^J \mathcal{A}^j$ is stable for canonical distance.*

Proof. If X_1 and X_2 are two samples drawn from distribution P, then we have:

$$d_P\left(\mathcal{A}(X_1), \mathcal{A}(X_2)\right) = \frac{1}{J}\sum_{j=1}^{J} d^j_{P^j}\left(\left(\mathcal{A}(X_1)\right)^j, \left(\mathcal{A}(X_2)\right)^j\right) = \frac{1}{J}\sum_{j=1}^{J} d^j_{P^j}\left(\mathcal{A}^j(X_1^j), \mathcal{A}^j(X_2^j)\right) \tag{15.12}$$

From the linearity of the expected value, it comes that

$$stab(\mathcal{A}, P, m) = \frac{1}{J}\sum_{j=1}^{J} stab(\mathcal{A}^j, P^j, m)$$

hence the stability of $\mathcal{A}$. □

This result is rather intuitive, since concatenation corresponds to an absence of collaboration. From this point of view, it is expected that, when stable local algorithms do not collaborate, the result of the non-collaboration remains stable. More interestingly, the same proof can be applied to get a more general result.

Before we present this general result, we have to introduce a key notion of collaborative clustering, **consistency**. Consistency is a desired property of collaborative clustering algorithms which states that the updated results must be somehow similar to the original local results. We formalize this notion in the following definition:

Definition 21. *Let P be a probability distribution over $\mathcal{X}$. Let d be a clustering distance. Let $\mathcal{A}$ be a collaborative clustering algorithm. The consistency of the collaborative algorithm $\mathcal{A} = \langle \mathcal{A}^1, \ldots, \mathcal{A}^J \rangle$ for the sample size m with respect to the probability distribution P is*

$$cons(\mathcal{A}, P, m) = \mathop{\mathbb{E}}_{X \sim P^m} \left[d_P \left(\mathcal{A}(X), \bigoplus_{j=1}^{J} \mathcal{A}^j(X^j) \right) \right].$$

The consistency of algorithm $\mathcal{A}$ with respect to the probability distribution P is

$$cons(\mathcal{A}, P) = \limsup_{m \to \infty} cons(\mathcal{A}, P, m)$$

Intuitively, consistency measures the distance of the global clustering produced by the collaboration to the clustering produced by concatenation of local algorithms. We recall that the clustering distances are pseudo-distance and do not satisfy the property $d(\mathcal{C}_1, \mathcal{C}_2) = 0 \Rightarrow \mathcal{C}_1 = \mathcal{C}_2$. For instance, it can be easily verified that, with the Hamming distance $d_P(\mathcal{C}_1, \mathcal{C}_2) = Pr_{X,Y \sim P}[(x \sim_{\mathcal{C}_1} y) \oplus (x \sim_{\mathcal{C}_2} y)]$, where $\oplus$ designates the XOR operation, two clusterings have a zero distance if they differ only a zero measure set. As a consequence, consistent algorithms are not necessarily concatenations.

As an example, consider two collaborators $\mathcal{A}^1$ and $\mathcal{A}^2$ working on $\mathbb{X}_1 = \mathbb{X}_2 = \mathbb{R}$. We define a collaborative clustering algorithm $\mathcal{A} = \langle \mathcal{A}^1, \mathcal{A}^2 \rangle$ that produces local clusterings of the form:

$$\mathcal{C}(x, y) = \begin{cases} \langle \mathcal{C}^2(x), \mathcal{C}^2(y) \rangle & \text{if } x \in \mathbb{Q} \\ \langle \mathcal{C}^1(x), \mathcal{C}^2(y) \rangle & \text{otherwise} \end{cases} \tag{15.13}$$

where $\mathbb{Q}$ is the set of rational numbers, and $\mathcal{C}^1$ is the local clustering computed by algorithm $\mathcal{A}^i$. This clustering differs from the simple concatenation on the set of rational numbers for the first collaborator, hence has a zero Hamming distance toward concatenation. As a consequence, algorithm $\mathcal{A}$ is consistent.

Consistency is naturally involved in a fundamental result on stability:

Theorem 10. *Let $\mathcal{A} = \langle \mathcal{A}^1, \ldots, \mathcal{A}^J \rangle$ be a collaborative clustering algorithm. Then the stability of $\mathcal{A}$ relatively to the canonical distance is upper-bounded as follows:*

$$stab(\mathcal{A}, P) \leq cons(\mathcal{A}, P) + \frac{1}{J} \sum_{j=1}^{J} stab(\mathcal{A}^j, P^j) \tag{15.14}$$

Proof. Consider X_1 and X_2 two samples drawn from distribution P. Since the canonical distance satisfies the triangular inequality:

$$d_P(\mathcal{A}(X_1), \mathcal{A}(X_1)) \leq d_P\left(\mathcal{A}(X_1), \left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X_1)\right) + d_P\left(\left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X_1), \left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X_2)\right) + d_P\left(\left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X_2), \mathcal{A}(X_2)\right)$$

Taking the expected value of this expression, we obtain:

$$stab(\mathcal{A}, P, m) \leq 2 \times \mathbb{E}_{X \sim P^m}\left[d_P\left(\mathcal{A}(X), \left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X)\right)\right] + \mathbb{E}_{X_1, X_2 \sim P^m}\left[d_P\left(\left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X_1), \left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X_2)\right)\right]$$

which is exactly the desired result. □

This result is really general since it does not require any information on the collaborative process. It has a direct consequence on collaborative clustering stability:

Corollary 3. *Any consistent collaboration of stable algorithms is stable for the canonical distance.*

It is noticeable that these two results are extremely limited. The consistency assumption is extremely strong and does not apply to "reasonable" collaborations, which are expected to find new structures in each view, and thus differ significantly from the concatenation. In practice, collaborative clustering has to find a trade-off between novelty (finding new structures and partitions) and consistency (relying on locally determined structures). Consequently, the result of this Corollary is barely applicable to real situations.

As another remark, Theorem 10 does not mean that non-consistent collaborations cannot be stable, or, equivalently, that a collaboration of unstable local algorithms cannot be stable. However, the result relies on the triangle inequality only, hence is the most precise result that can be obtained based on the definition only. Further investigations on collaborative clustering stability has to be found elsewhere, probably in the direction of Proposition 13.

15.3.4 Perspectives

The results on collaborative clustering stability presented above are the result of preliminary works that has been done with the idea of building a theoretical framework for unsupervised collaboration. At this stage, it is almost only an adaptation of the existing theory to the multi-view context. We proposed two main theorems on clustering stability.

The first theorem (Proposition 13) is based on a risk-minimization scheme and proves the stability of a collaborative clustering algorithm under two conditions: existence of a unique *global* minimizer and risk convergence. The main question that arises from this theorem is to determine if these two conditions can be expressed in a more suitable way for collaborative clustering? This question can be divided in two

parts. The question of the existence of a unique minimizer is relative to the division of the space into local subspaces (the views). However, it is important to keep in mind that the notion of *minimizer* is related to a specific risk. The choice of a risk function is also inherently fundamental in the risk convergence property.

Given the state of the art and the model that has been used in Section 15.2, it seems tempting to consider the generic form of risk as the sum of local risks and of a collaborative term:

$$R(P,\mathcal{C}) = \sum_{j=1}^{J} \left(R^j(P^j,\mathcal{C}^j) + \sum_{i\neq j} \Delta(P,\mathcal{C}^i,\mathcal{C}^j) \right) \tag{15.15}$$

The second theorem (Theorem 10 gives an upper-bound of stability in terms of the stability of local algorithms and global consistency (ie. distance to concatenation). A restricted application of this theorem states that any consistent collaboration of stable algorithms remains stable. However, this algorithm does not solve the other cases: inconsistent collaboration or unstable local algorithms. In particular, several questions remain open: Is one unstable local algorithm enough to make the collaboration of stable algorithms unstable? Can an inconsistent collaboration of stable elements be stable? Such questions are of real interest for the theory of collaboration and should find an answer.

However, we considered here the case of a global stability, which means the stability of the collaborative clustering algorithm considered as a clustering algorithm on $\mathbb{X}$. Another open question is the stability of the local algorithms defined from $\mathcal{A}$ (hence the restriction of $\mathcal{A}$ on one of the views).

15.4 Conclusion

In this chapter, we presented a couple of thoughts relative to the fundamental nature of collaborative clustering. We have shown that the issues raised in this domain are completely different from the questions of supervised ensemble learning, because of the absence of supervision and of objective quality measure. We proposed two models for the quality of a collaboration. In the first model, we asked the question of the choice of good collaborators in a general setting. We have shown that, under some simple but non-restrictive assumptions, the best collaboration (in terms of score, or likelihood) is actually no collaboration at all: The best way for an algorithm to collaborate is to collaborate with the most similar algorithm. The second model we propose is an adaptation of the theory of stability. First theoretical results were introduced but the whole question of collaboration remains open. Such questions will have to be investigated in future works.

Part V

Conclusion

Chapter 16

Conclusion

The general scope of this thesis being rather wide, exploring various domains, we would like to conclude with a short reminder of the main contributions of our work. This recap will leave a sour taste of unfinished work, but the main lesson we have learned from these three years is that, in research, a conclusion is necessarily an opening to new questions: Following this optimistic philosophy, we will conclude this thesis by some perspectives for future works, to be done by ourselves or by other passionate researchers.

16.1 Contributions

We first provide a short list of our main contributions. We divide this list in two parts: general contributions first, followed by contributions specific to one precise domain.

16.1.1 General Contributions

The main direction chosen in this thesis is rather unusual to our knowledge. The ambition was to address multiple research questions, taken from various domains, with only one aspect in common: the notion of knowledge transfer. We defined knowledge transfer in a very informal way as the necessity to share information in the learning process, either from one task to another task (such as in analogical reasoning and in transfer learning), from the past to the present (such as in incremental learning) or from one agent to the other (such as in collaborative clustering). Given this unified definition, many questions arose, to which we tried to find general answers:

- **Is there a general tool to assess the general question of knowledge transfer?** We claimed that Kolmogorov complexity is a perfect candidate to play this role. Complexity of an object is defined as the length of the shortest program, on a Turing machine, that can generate this object. It measures the information contained in an object, and can be used in particular to measure the quantity of transferred information.

- **Is transfer always possible?** We have seen that this question is quite ambiguous and does not correctly illustrate the actual issues of transfer. We gave two answers to this question. The first answer is *yes*: Given a transfer algorithm, it is always possible to apply it to a problem requiring transfer. The second answer is *no*: Sometimes, transferring unrelated knowledge is worse, in terms of the followed objective, than relying on the target problem only. These two observations lead naturally to the next question:

- **When is transfer helpful?** We do not have any answer to this question yet. We proposed a framework to assess this question (Chapter 9) but the suggested definitions are first steps in a direction that remains to be better defined. The proposed solution is the following: Transfer is possible only in case the source knowledge leads to a better compression of the target problem. We will discuss, in the perspectives, the questions opened by this approach.

More technically, one of our main contributions is the definition of Descriptive Graphical Models (DGM), which are generalization of Probabilistic Graphical Models to non-probabilistic Turing machines. These models are based on the use of Kolmogorov complexity and can be interpreted as general machines that can produce complex objects. Based on these machines, we suggested a general methodology for the description of transfer problems. This model is widely inspired by an approach to analogical reasoning developed by (Cornuéjols and Ales-Bianchetti, 1998), its main strength is to rely not on a direct description of data, but on intermediate objects that we called *models*. Models are helpful since they are ideal candidates for the transfer phase. As exposed first by the Structure Mapping Theory (Gentner, 1983), a good transfer has to focus on structural description of objects, and not on local irregularities (noise or high level descriptors), and thus data themselves are not adapted. If all the common information about objects is stored in these models, the transfer can be done at this low level only and be more efficient. We followed this idea of using models in the four domains we have explored: analogy, transfer learning, stream mining and multi-source clustering.

16.1.2 Local Contributions

We choose to group our contributions by domains. These domains can be related to the four parts of the thesis or are more transverse.

16.1.2.1 Analogical Reasoning

Analogical reasoning is a domain that focuses on questions of the form “A is to B as C is to x", where x is unknown and has to be found. As mentioned above, a first model, based on Kolmogorov complexity, was introduced by (Cornuéjols and Ales-Bianchetti, 1998). Our contribution, relative to this model, is to be found in the formalization that we proposed. This formalization is based on the Descriptive Graphical Models, described as a general contribution.

Our second contribution is a formal study of minimum complexity analogies in geometric spaces (Chapter 6). Our idea is that analogies can be defined in terms of transformations, and that some transformations are more “natural" than others. For instance, the very nature of vector spaces makes additions and substractions intuitive, hence simple, operations. In Riemannian spaces, we suggested that parallel transport is a natural operation, defined directly by the metric. We proposed an analysis of the produced analogies and compared them with the axioms of proportional analogy.

Our last contribution follows from the previous one. We showed that the analogy defined with help of parallel transport does not satisfy the axioms of analogical proportion. We thus addressed the following question: Is it possible to define analogical proportions on differential manifolds? The answer to this question is yes. However, we could not provide any evidence that these proportions are “continuous".

16.1.2.2 Transfer Learning

Our main contribution in the domain of transfer learning is to propose an interpretation of transfer in terms of analogical reasoning. Based on this observation, we used the same model as presented to solve analogies. This methodology is based on *models*, hence intermediate objects that encode global low level information about observed objects. This approach is consistent with mapping-based methods, that project source and target data into a common intermediate space.

We illustrated our approach by developing a simple prototype-based model, inspired by Learning Vector Quantization. We illustrated the transfer with multiple simple algorithms based on this model.

16.1.2.3 Data Stream Mining

Data stream mining is a bit different from the previously exposed tasks, since it does not consist of a source and a target, but of a stream of data. We proposed to use the same approach as transfer learning, but to provide a model adaptation based on the past sequence. The main advantage of our description is that it provides a general formalization of the problem of data stream mining, which can be used for theoretical studies of data stream mining.

Our second contribution to the domain of data stream mining is a study of online recommendation, with applications on textual data (Chapter 11). We proposed a very simple algorithm to deal with streams of text and provide an up-to-date topic model. Based on this idea, we proposed an hybrid recommendation algorithm. The results we have shown tend to confirm the intuition that online recommender systems have issues of their own: In particular, we have demonstrated in our experiments that hybrid recommender systems perform worse than only collaborative filtering algorithms when the item model is not regularly updated.

16.1.2.4 A Cognitive Model

Humans are known to be sensible to Kolmogorov complexity (Chater, 1999). We have studied this question in the restricted context of analogies on character strings and their applications.

A first contribution is the formal study of the models in the case of Hofstadter's micro-world. Hofstadter's micro-world is a set of analogical problems defined on character strings. In Chapter 4, we have proposed a way to define the models used in the DGM suggested for analogical reasoning. Our solution is based on a descriptive and memory-based language. We have shown that the complexity results obtained with this language were coherent with the results obtained by humans.

The defined models were then used as an application of our approach of data stream mining (Chapter 12). We studied a well-known phenomenon in cognitive sciences called "U-shaped learning". This phenomenon models a process of learning, un-learning and re-learning that is typical of first language acquisition for instance. We provided simple experiments (on a restricted domain) in order to observe if our model can produce such a phenomenon and we observed that it actually can.

A last contribution was a short interpretation of the phenomenon of syntactic priming, that can be modeled as an analogical reasoning problem.

16.1.2.5 Multi-Source Clustering

As opposed to standard clustering, multi-source clustering regards the cooperation or collaboration between several clustering algorithms operating on the same data but with different views or biases on them.

We first interpreted the problem in terms of DGM and Kolmogorov complexity (Chapter 14). We developed a very simple framework, as well as an algorithm to test it. An advantage of our approach is that it allows a collaboration between any type of algorithm, but also considers local information about the algorithms. To our knowledge, there is no other method that satisfies these two constraints together.

Finally, we proposed a preliminary study of the idea of collaboration in clustering. First, we asked the question of the choice of good collaborators. Using a very simple weighting approach, we demonstrated that the best collaboration strategy for a clustering algorithm is to exchange information with algorithms that gives the same results. This observation is a major difference with collaboration in supervised setting. Finally, we proposed a formalism and several preliminary results for the study of stability for collaborative clustering.

16.2 Perspectives and Future Works

The results presented in this thesis open new perspectives in the domains we have explored. We want to conclude with an incomplete list of research directions that directly follow from our work.

- **Automatic search for structures in Hofstadter's micro-world:** The developed language seems very convenient for the description of character strings, but in this thesis, all the programs were designed at hand. An automatic search for the shortest program will have to be done, maybe following the directions exposed in Section 4.4.
- **Existence of continuous proportional analogies on differential manifolds:** We have shown the existence of continuous non-proportional analogies, and of non-continuous proportional analogies on Riemannian manifolds. The question we have not succeeded to answer regards the existence of continuous proportional analogies on manifolds.
- **Extension of the parallelogrammoid procedure:** The algorithm proposed in Chapter 6 works only when the four terms of the analogy are in the same space. It would be interesting to find a generalization of this procedure when the spaces differ.
- **Study of learnability:** We proposed new notions of learnability and transferability, but in a very preliminary way. More work has to be proposed in order to determine whether this approach is useful. In particular, a strong result that could be obtained would show a link between our notion of learnability and PAC-learnability.
- **Universal Prior for Learning with Concern for Future Questions:** Learning with concern for future questions is a general learning framework, which gives to the learner a prior over the future. This framework is intended to go beyond classical assumptions such as the i.i.d. hypothesis. The question of the definition of the prior remains open, but we suggest to consider universal distribution as a potential prior.

- **Clustering cooperation with MDL principle:** The algorithm we proposed in Chapter 14 works only for collaborative clustering. An adaptation is required for cooperative clustering (ie. to find a consensus between different views).
- **Collaborative clustering stability:** The results presented in Section 15.3 are preliminary results and must be completed.

A larger, and maybe more philosophical perspective, is more a promise than a merely technical question. All along this thesis, we have done our best to merge the spirits of various disciplines and to take the best of all these domains, ranging from clustering to cognitive modeling. We have demonstrated that it is possible to unify them with common principles. In future works, we will do our best to keep this distance in order to propose results as general as possible.

Appendix A

Experiment on Hofstadter's Analogies

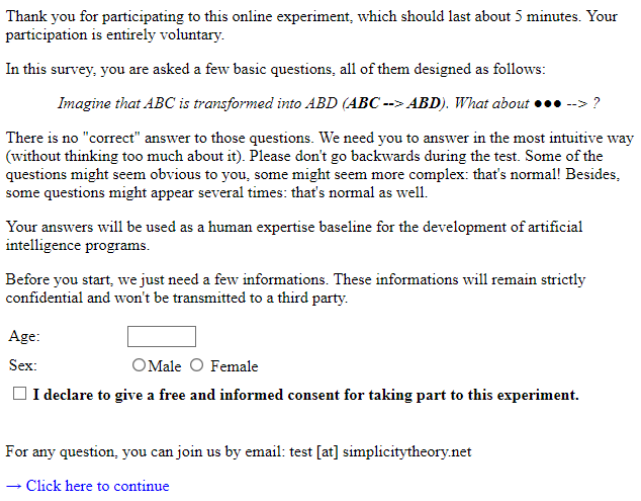
In this chapter, we present the protocol and complete results of the small experiment on Hofstadter's analogies, presented in Chapter 4. This presentation is divided into three sections. In a first section, we expose the whole protocol as well as the complete set of questions. In a second section, we discuss the way we filtered and processed collected data. Finally, we propose detailed results of these experiment as well as a short analysis of the profile of participants.

A.1 Experiment Protocol

In order to evaluate human results on Hofstadter's analogy problems, we proposed an online experiment[1]. The experiment was designed on the IBEX farm platform (Drummond, 2013).

The test was publicly opened and made available online. The link to the test was sent directly to a couple of potential participants (external to the research problem) and shared on social networks in order to reach a broader audience in terms of academic background.

The home page of the survey is reproduced in Figure A.2. The page corresponding to a question is reproduced in Figure



Thank you for participating to this online experiment, which should last about 5 minutes. Your participation is entirely voluntary.

In this survey, you are asked a few basic questions, all of them designed as follows:

*Imagine that ABC is transformed into ABD (**ABC --> ABD**). What about ●●● --> ?*

There is no "correct" answer to those questions. We need you to answer in the most intuitive way (without thinking too much about it). Please don't go backwards during the test. Some of the questions might seem obvious to you, some might seem more complex: that's normal! Besides, some questions might appear several times: that's normal as well.

Your answers will be used as a human expertise baseline for the development of artificial intelligence programs.

Before you start, we just need a few informations. These informations will remain strictly confidential and won't be transmitted to a third party.

Age:

Sex: ○Male ○ Female

☐ **I declare to give a free and informed consent for taking part to this experiment.**

For any question, you can join us by email: test [at] simplicitytheory.net

→ Click here to continue

FIGURE A.1: Home page of the online survey.

[1] Available at the address `http://spellout.net/ibexexps/pam/Analogy/experiment.html`.

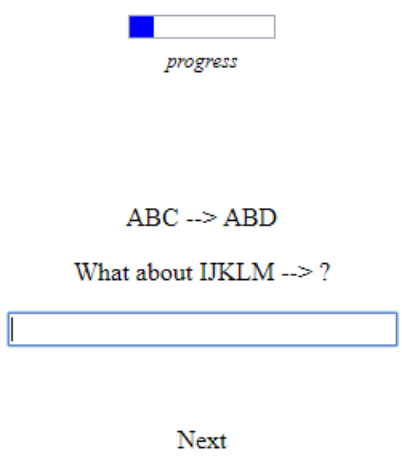


FIGURE A.2: Question page in the online survey.

A.2 Filtering Results

In order to have significant results, we removed all participants who did not provide a "serious" answer to more than three questions. As "not serious", we designate the answers which:

- Correspond to general comments on a question or on the whole experiment (e.g. "this test is random")
- Are purposely out of the context (e.g. "ABC : ABD :: IJK : 42").
- Are sentences or equivalent (e.g. "I believe I can flyyyy" or "LOL")

After this initial filtering step, we removed six participants. Notice that some answers remain that satisfy at least one of the mentioned conditions (which can be seen in the detailed results presented in the next section).

In order to make the results homogeneous, we show the results in capital letters. For instance, we transfered the proposed solution"abd" into "ABD".

No other processing has been made on the data. In particular, we did not fix obvious typos (for instance IJK : **O**JL) nor changed the additional blanks (for instance "1 4 10" is not merged with the answer "1410").

A.3 Detailed Results

We present the results obtained with our experiment. To make them understandable, we first present them in their raw form, and then group results by categories.

A.3.1 Raw Results

The raw results are presented in the following pages. The results are grouped by submission, giving the submission date, the age and the gender, as well as the proposed solutions to the analogy.

Result:

- Date: Friday January 13 2017 16:07:50 UTC
- Age: 24
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	9296
ABC : ABD :: BCA	BCB	12217
ABC : ABD :: AABABC	AABABD	25871
ABC : ABD :: IJKLM	IJKLN	16423
ABC : ABD :: 123	124	5100
ABC : ABD :: KJI	LJI	11499
ABC : ABD :: 135	136	10245
ABC : ABD :: BCD	BCE	12451
ABC : ABD :: IJJKKK	IJJLLL	11587
ABC : ABD :: XYZ	XYA	5202
ABC : ABD :: 122333	122444	9532
ABC : ABD :: RSSTTT	RSSUUU	10543
ABC : ABD :: IJJKKK	IJJLLL	6792
ABC : ABD :: AABABC	AABABB	26272
ABC : ABD :: MRRJJJ	MRRKKK	12610
ABC : ABD :: 147	148	7053

Result:

- Date: Friday January 13 2017 16:08:14 UTC
- Age: 21
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	9285
ABC : ABD :: BCA	BCB	27811
ABC : ABD :: AABABC	AACABD	27265
ABC : ABD :: IJKLM	IJLLM	9216
ABC : ABD :: 123	124	4472
ABC : ABD :: KJI	KJJ	16146
ABC : ABD :: 135	136	4720
ABC : ABD :: BCD	BCE	6573
ABC : ABD :: IJJKKK	IJKKKL	9525
ABC : ABD :: XYZ	XYA	5813
ABC : ABD :: 122333	123334	6661
ABC : ABD :: RSSTTT	RSTTTU	17073
ABC : ABD :: IJJKKK	IJKKKL	13133
ABC : ABD :: AABABC	AACABD	10773
ABC : ABD :: MRRJJJ	MRSJJK	13505
ABC : ABD :: 147	148	14247

Result:

- Date: Friday January 13 2017 16:18:26 UTC
- Age: 23
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJD	8382
ABC : ABD :: BCA	BCD	10748
ABC : ABD :: AABABC	AADABC	34092
ABC : ABD :: IJKLM	IJKLD	16961
ABC : ABD :: 123	123	6818
ABC : ABD :: KJI	KJD	18691
ABC : ABD :: 135	13D	16643
ABC : ABD :: BCD	BCD	8900
ABC : ABD :: IJJKKK	IJDDDD	58865
ABC : ABD :: XYZ	XYA	12246
ABC : ABD :: 122333	122334	14910
ABC : ABD :: RSSTTT	RSDTTT	11039
ABC : ABD :: IJJKKK	IJJDDD	12706
ABC : ABD :: AABABC	AABDBD	24916
ABC : ABD :: MRRJJJ	MRDJJD	29014
ABC : ABD :: 147	148	14198

Result:

- Date: Friday January 13 2017 16:25:19 UTC
- Age: 24
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	35122
ABC : ABD :: BCA	BCB	47165
ABC : ABD :: AABABC	AABABD	25501
ABC : ABD :: IJKLM	IJKLN	6597
ABC : ABD :: 123	124	6381
ABC : ABD :: KJI	KJJ	6190
ABC : ABD :: 135	136	5308
ABC : ABD :: BCD	BCE	13289
ABC : ABD :: IJJKKK	IJJKKL	8097
ABC : ABD :: XYZ	XYA	12364
ABC : ABD :: 122333	122334	7236
ABC : ABD :: RSSTTT	RSSTTU	5694
ABC : ABD :: IJJKKK	IJJKKL	7430
ABC : ABD :: AABABC	AABABD	7262
ABC : ABD :: MRRJJJ	MRRJJK	9062
ABC : ABD :: 147	148	5422

Result:

- Date: Friday January 13 2017 16:27:08 UTC
- Age: 23
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	18384
ABC : ABD :: BCA	BCB	9272
ABC : ABD :: AABABC	AABABD	58382
ABC : ABD :: IJKLM	IJJKLM	17079
ABC : ABD :: 123	124	9192
ABC : ABD :: KJI	KJI	6712
ABC : ABD :: 135	145	11286
ABC : ABD :: BCD	BDD	9019
ABC : ABD :: IJJKKK	IJJKKK	9507
ABC : ABD :: XYZ	XYZ	8505
ABC : ABD :: 122333	122433	16922
ABC : ABD :: RSSTTT	RSSTTT	7365
ABC : ABD :: IJJKKK	IJJKKK	7733
ABC : ABD :: AABABC	AABABD	18104
ABC : ABD :: MRRJJJ	MRRJJJ	5331
ABC : ABD :: 147	147	6790

Result:

- Date: Friday January 13 2017 16:32:34 UTC
- Age: 25
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	6218
ABC : ABD :: BCA	BDA	8034
ABC : ABD :: AABABC	AABABD	8900
ABC : ABD :: IJKLM	IJLMN	16037
ABC : ABD :: 123	124	4717
ABC : ABD :: KJI	LJI	5922
ABC : ABD :: 135	137	3866
ABC : ABD :: BCD	BCE	9358
ABC : ABD :: IJJKKK	IJJLLL	6835
ABC : ABD :: XYZ	XYA	8131
ABC : ABD :: 122333	122444	6691
ABC : ABD :: RSSTTT	RSSUUU	5158
ABC : ABD :: IJJKKK	IJJLLL	3924
ABC : ABD :: AABABC	AABABD	3761
ABC : ABD :: MRRJJJ	MRRLLL	18779
ABC : ABD :: 147	1410	6964

Result:

- Date: Friday January 13 2017 16:49:48 UTC
- Age: 22
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	12615
ABC : ABD :: BCA	BCB	11204
ABC : ABD :: AABABC	AACABD	15706
ABC : ABD :: IJKLM	IJLLM	15646
ABC : ABD :: 123	124	6169
ABC : ABD :: KJI	KJJ	6777
ABC : ABD :: 135	136	5069
ABC : ABD :: BCD	BCE	7280
ABC : ABD :: IJJKKK	IJKKKL	13533
ABC : ABD :: XYZ	XYA	7175
ABC : ABD :: 122333	123334	8369
ABC : ABD :: RSSTTT	RSSTTTU	14351
ABC : ABD :: IJJKKK	IJKKKL	9871
ABC : ABD :: AABABC	AACABD	103440
ABC : ABD :: MRRJJJ	MRSJJK	9189
ABC : ABD :: 147	148	5247

Result:

- Date: Friday January 13 2017 16:59:13 UTC
- Age: 21
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	11331
ABC : ABD :: BCA	BDA	36865
ABC : ABD :: AABABC	AABABD	14020
ABC : ABD :: IJKLM	IJLLM	18310
ABC : ABD :: 123	124	4513
ABC : ABD :: KJI	LJI	9401
ABC : ABD :: 135	136	3627
ABC : ABD :: BCD	BCE	6388
ABC : ABD :: IJJKKK	IJJLLL	5865
ABC : ABD :: XYZ	XYA	5217
ABC : ABD :: 122333	122444	5390
ABC : ABD :: RSSTTT	RSSUUU	8748
ABC : ABD :: IJJKKK	IJJLLL	4781
ABC : ABD :: AABABC	AABABD	6157
ABC : ABD :: MRRJJJ	MSSJJJ	22565
ABC : ABD :: 147	148	3357

Result:

- Date: Friday January 13 2017 17:06:34 UTC
- Age: 54
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	23580
ABC : ABD :: BCA	DBA	13741
ABC : ABD :: AABABC	AABABD	38607
ABC : ABD :: IJKLM	IJLLM	82164
ABC : ABD :: 123	124	6050
ABC : ABD :: KJI	LJI	17936
ABC : ABD :: 135	137	18110
ABC : ABD :: BCD	BCE	9625
ABC : ABD :: IJJKKK	IJJLLL	18578
ABC : ABD :: XYZ	XYA	11441
ABC : ABD :: 122333	122444	14694
ABC : ABD :: RSSTTT	RSSUUU	12600
ABC : ABD :: IJJKKK	IJJLLL	10090
ABC : ABD :: AABABC	AABABD	16645
ABC : ABD :: MRRJJJ	MRRLLL	68325
ABC : ABD :: 147	140	20549

Result:

- Date: Friday January 13 2017 17:08:25 UTC
- Age: 25
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	14284
ABC : ABD :: BCA	BCB	21688
ABC : ABD :: AABABC	AACABD	39292
ABC : ABD :: IJKLM	IJKLN	13682
ABC : ABD :: 123	124	5134
ABC : ABD :: KJI	KJH	23245
ABC : ABD :: 135	136	5185
ABC : ABD :: BCD	BCE	12538
ABC : ABD :: IJJKKK	IJJKKL	8479
ABC : ABD :: XYZ	XYA	6389
ABC : ABD :: 122333	122334	7606
ABC : ABD :: RSSTTT	RSSTTU	7600
ABC : ABD :: IJJKKK	IJJKKL	6607
ABC : ABD :: AABABC	AABABC	8263
ABC : ABD :: MRRJJJ	MRRJJK	8557
ABC : ABD :: 147	148	8372

Result:

- Date: Friday January 13 2017 17:16:25 UTC
- Age: 22
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IKL	18426
ABC : ABD :: BCA	BCB	11544
ABC : ABD :: AABABC	AABABD	36563
ABC : ABD :: IJKLM	IJKLN	21096
ABC : ABD :: 123	124	5553
ABC : ABD :: KJI	KJJ	7642
ABC : ABD :: 135	136	5864
ABC : ABD :: BCD	BCE	5426
ABC : ABD :: IJJKKK	IJJKL	8137
ABC : ABD :: XYZ	XYA	13009
ABC : ABD :: 122333	122334	7821
ABC : ABD :: RSSTTT	RSSTTU	7190
ABC : ABD :: IJJKKK	IJJKKL	7360
ABC : ABD :: AABABC	AABABD	6576
ABC : ABD :: MRRJJJ	MRRJJK	9412
ABC : ABD :: 147	148	4409

Result:

- Date: Friday January 13 2017 17:25:20 UTC
- Age: 25
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	8412
ABC : ABD :: BCA	BDA	12029
ABC : ABD :: AABABC	AABABD	6338
ABC : ABD :: IJKLM	IJLMN	10479
ABC : ABD :: 123	124	3260
ABC : ABD :: KJI	LJI	6950
ABC : ABD :: 135	146	4962
ABC : ABD :: BCD	BDE	5386
ABC : ABD :: IJJKKK	IJJLLL	8696
ABC : ABD :: XYZ	XYA	6277
ABC : ABD :: 122333	122444	2990
ABC : ABD :: RSSTTT	RSSUUU	5344
ABC : ABD :: IJJKKK	IJJLLL	5909
ABC : ABD :: AABABC	AABABD	3844
ABC : ABD :: MRRJJJ	MRRJJ	48148
ABC : ABD :: 147	158	2683

Result:

- Date: Friday January 13 2017 18:28:42 UTC
- Age: 25
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	16344
ABC : ABD :: BCA	BCB	26013
ABC : ABD :: AABABC	AABABD	32465
ABC : ABD :: IJKLM	IJKLN	8777
ABC : ABD :: 123	123	3962
ABC : ABD :: KJI	KJJ	9506
ABC : ABD :: 135	135	5220
ABC : ABD :: BCD	BCE	5587
ABC : ABD :: IJJKKK	IJJKKL	6329
ABC : ABD :: XYZ	XYA	5982
ABC : ABD :: 122333	122333	6475
ABC : ABD :: RSSTTT	RSSTTU	8298
ABC : ABD :: IJJKKK	IJJKKL	5226
ABC : ABD :: AABABC	AABABD	5711
ABC : ABD :: MRRJJJ	MRRJJK	7781
ABC : ABD :: 147	147	6557

Result:

- Date: Friday January 13 2017 18:38:03 UTC
- Age: 22
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	11248
ABC : ABD :: BCA	BDA	17609
ABC : ABD :: AABABC	AABABD	28406
ABC : ABD :: IJKLM	IJKLN	59997
ABC : ABD :: 123	124	8859
ABC : ABD :: KJI	KJH	67923
ABC : ABD :: 135	137	20330
ABC : ABD :: BCD	BCE	18551
ABC : ABD :: IJJKKK	IJJLLL	41267
ABC : ABD :: XYZ	XYA	8104
ABC : ABD :: 122333	122444	40548
ABC : ABD :: RSSTTT	RSSUUU	10920
ABC : ABD :: IJJKKK	IJJLLL	10593
ABC : ABD :: AABABC	AABABD	12371
ABC : ABD :: MRRJJJ	MRREEE	138518
ABC : ABD :: 147	1 4 10	321388

Result:

- Date: Friday January 13 2017 19:46:44 UTC
- Age: 25
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	6438
ABC : ABD :: BCA	BCB	8632
ABC : ABD :: AABABC	AABABD	8557
ABC : ABD :: IJKLM	IJKLN	5226
ABC : ABD :: 123	124	2746
ABC : ABD :: KJI	KJJ	5157
ABC : ABD :: 135	136	4529
ABC : ABD :: BCD	BCE	3831
ABC : ABD :: IJJKKK	IJJKKL	4431
ABC : ABD :: XYZ	XYA	4346
ABC : ABD :: 122333	122334	3561
ABC : ABD :: RSSTTT	RSSTTU	5357
ABC : ABD :: IJJKKK	IJJKKL	4869
ABC : ABD :: AABABC	AABABD	6651
ABC : ABD :: MRRJJJ	MRRJJK	4817
ABC : ABD :: 147	148	2045

Result:

- Date: Friday January 13 2017 20:12:32 UTC
- Age: 24
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	14409
ABC : ABD :: BCA	BCB	33572
ABC : ABD :: AABABC	AABBD	23395
ABC : ABD :: IJKLM	IJKLN	25401
ABC : ABD :: 123	124	6317
ABC : ABD :: KJI	KJK	15208
ABC : ABD :: 135	136	9715
ABC : ABD :: BCD	BCE	11632
ABC : ABD :: IJJKKK	IJJKL	8506
ABC : ABD :: XYZ	XYA	7535
ABC : ABD :: 122333	1224444	24557
ABC : ABD :: RSSTTT	RSSUUU	10172
ABC : ABD :: IJJKKK	IJLLL	10006
ABC : ABD :: AABABC	AABACD	18653
ABC : ABD :: MRRJJJ	MRRLLLL	14354
ABC : ABD :: 147	148	9052

Result:

- Date: Friday January 13 2017 23:33:25 UTC
- Age: 23
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	21705
ABC : ABD :: BCA	BDA	85800
ABC : ABD :: AABABC	AABABD	26153
ABC : ABD :: IJKLM	IJKLN	16036
ABC : ABD :: 123	124	5926
ABC : ABD :: KJI	LJI	48820
ABC : ABD :: 135	137	15169
ABC : ABD :: BCD	BCE	755995
ABC : ABD :: IJJKKK	IJJLLL	12319
ABC : ABD :: XYZ	XYA	23299
ABC : ABD :: 122333	122444	9123
ABC : ABD :: RSSTTT	RSSUUU	9422
ABC : ABD :: IJJKKK	IJJLLL	12087
ABC : ABD :: AABABC	AABABD	9680
ABC : ABD :: MRRJJJ	MRRKKK	226000
ABC : ABD :: 147	148	27414

Result:

- Date: Saturday January 14 2017 02:59:21 UTC
- Age: 26
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	18110
ABC : ABD :: BCA	BCB	15976
ABC : ABD :: AABABC	AACABD	22266
ABC : ABD :: IJKLM	IJLLM	14518
ABC : ABD :: 123	124	7144
ABC : ABD :: KJI	KJJ	18050
ABC : ABD :: 135	136	4328
ABC : ABD :: BCD	BCE	8835
ABC : ABD :: IJJKKK	IJKKKL	12427
ABC : ABD :: XYZ	XYA	8993
ABC : ABD :: 122333	123334	7917
ABC : ABD :: RSSTTT	RSTTTU	14135
ABC : ABD :: IJJKKK	IJKKKL	13472
ABC : ABD :: AABABC	AACABD	10679
ABC : ABD :: MRRJJJ	MRSJJK	15913
ABC : ABD :: 147	148	3669

Result:

- Date: Saturday January 14 2017 08:32:53 UTC
- Age: 27
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	7835
ABC : ABD :: BCA	BDA	18189
ABC : ABD :: AABABC	AABABD	16898
ABC : ABD :: IJKLM	IJKLN	9680
ABC : ABD :: 123	124	4218
ABC : ABD :: KJI	LJI	9687
ABC : ABD :: 135	136	8304
ABC : ABD :: BCD	BCE	8992
ABC : ABD :: IJJKKK	IJJLLL	7533
ABC : ABD :: XYZ	XYA	11559
ABC : ABD :: 122333	122444	5619
ABC : ABD :: RSSTTT	RSSUUU	7433
ABC : ABD :: IJJKKK	IJJLLL	8370
ABC : ABD :: AABABC	AABABD	6763
ABC : ABD :: MRRJJJ	MSSJJJ	18354
ABC : ABD :: 147	148	4553

Result:

- Date: Saturday January 14 2017 08:40:14 UTC
- Age: 27
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	22623
ABC : ABD :: BCA	BCB	19339
ABC : ABD :: AABABC	AABABD	40208
ABC : ABD :: IJKLM	IJKLN	33272
ABC : ABD :: 123	124	8122
ABC : ABD :: KJI	KJJ	26485
ABC : ABD :: 135	136	9773
ABC : ABD :: BCD	BCE	13645
ABC : ABD :: IJJKKK	IJJLLL	33025
ABC : ABD :: XYZ	XYA1	40983
ABC : ABD :: 122333	122444	16021
ABC : ABD :: RSSTTT	RSSUUU	16811
ABC : ABD :: IJJKKK	IJJLLL	11794
ABC : ABD :: AABABC	AABABD	20219
ABC : ABD :: MRRJJJ	MRRKKK	13671
ABC : ABD :: 147	148	9625

Result:

- Date: Saturday January 14 2017 11:08:43 UTC
- Age: 23
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	18429
ABC : ABD :: BCA	BDA	20353
ABC : ABD :: AABABC	AABABD	13758
ABC : ABD :: IJKLM	IJKLN	12219
ABC : ABD :: 123	124	5227
ABC : ABD :: KJI	LJI	20804
ABC : ABD :: 135	137	18316
ABC : ABD :: BCD	BCE	16920
ABC : ABD :: IJJKKK	IJJLLL	12536
ABC : ABD :: XYZ	XYA	16401
ABC : ABD :: 122333	122444	12212
ABC : ABD :: RSSTTT	RSSUUU	12707
ABC : ABD :: IJJKKK	IJJLLL	5645
ABC : ABD :: AABABC	AABABD	7959
ABC : ABD :: MRRJJJ	MRRLLL	37846
ABC : ABD :: 147	1410	16147

Result:

- Date: Saturday January 14 2017 12:37:44 UTC
- Age: 24
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	18417
ABC : ABD :: BCA	BCB	10142
ABC : ABD :: AABABC	AABABD	19801
ABC : ABD :: IJKLM	IJKLN	9516
ABC : ABD :: 123	124	5851
ABC : ABD :: KJI	KJJ	10121
ABC : ABD :: 135	136	6505
ABC : ABD :: BCD	BCE	5634
ABC : ABD :: IJJKKK	IJJKKL	7490
ABC : ABD :: XYZ	XYA	7632
ABC : ABD :: 122333	122334	7985
ABC : ABD :: RSSTTT	RSSTTU	5247
ABC : ABD :: IJJKKK	IJJKKL	5256
ABC : ABD :: AABABC	AABABD	5592
ABC : ABD :: MRRJJJ	MRRJJK	7545
ABC : ABD :: 147	148	3906

Result:

- Date: Saturday January 14 2017 12:48:58 UTC
- Age: 23
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	8494
ABC : ABD :: BCA	BDA	10007
ABC : ABD :: AABABC	AABABD	27012
ABC : ABD :: IJKLM	IJLLM	15836
ABC : ABD :: 123	124	2977
ABC : ABD :: KJI	KJI	7612
ABC : ABD :: 135	135	5022
ABC : ABD :: BCD	BCE	9227
ABC : ABD :: IJJKKK	IJKKKL	9646
ABC : ABD :: XYZ	XYA	3718
ABC : ABD :: 122333	123334	6046
ABC : ABD :: RSSTTT	RSTTTU	7745
ABC : ABD :: IJJKKK	IJKKKL	6749
ABC : ABD :: AABABC	AACABD	5783
ABC : ABD :: MRRJJJ	MRRJJJ	18445
ABC : ABD :: 147	148	3677

Result:

- Date: Saturday January 14 2017 12:49:09 UTC
- Age: 25
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	7588
ABC : ABD :: BCA	BDA	9590
ABC : ABD :: AABABC	AABABD	12021
ABC : ABD :: IJKLM	IJLLM	17279
ABC : ABD :: 123	124	5491
ABC : ABD :: KJI	KJJ	9978
ABC : ABD :: 135	136	3304
ABC : ABD :: BCD	BCE	5090
ABC : ABD :: IJJKKK	IJKKKL	8328
ABC : ABD :: XYZ	XYA	5947
ABC : ABD :: 122333	123334	5545
ABC : ABD :: RSSTTT	RSTTTU	7268
ABC : ABD :: IJJKKK	IJKKKL	7798
ABC : ABD :: AABABC	AACABD	15746
ABC : ABD :: MRRJJJ	MROJJK	13748
ABC : ABD :: 147	148	2688

Result:

- Date: Saturday January 14 2017 14:29:17 UTC
- Age: 23
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	20479
ABC : ABD :: BCA	BCB	10539
ABC : ABD :: AABABC	AABABD	15604
ABC : ABD :: IJKLM	IJKLN	10034
ABC : ABD :: 123	124	3908
ABC : ABD :: KJI	KJJ	5829
ABC : ABD :: 135	136	2529
ABC : ABD :: BCD	BCE	3539
ABC : ABD :: IJJKKK	IJJKKL	10661
ABC : ABD :: XYZ	XYA	5609
ABC : ABD :: 122333	122334	5448
ABC : ABD :: RSSTTT	RSSTTU	6139
ABC : ABD :: IJJKKK	IJJKKL	5460
ABC : ABD :: AABABC	AABABD	5677
ABC : ABD :: MRRJJJ	MRRJJK	8132
ABC : ABD :: 147	148	2897

Result:

- Date: Saturday January 14 2017 20:59:26 UTC
- Age: 25
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	5400
ABC : ABD :: BCA	BCB	5801
ABC : ABD :: AABABC	BBCBCD	8320
ABC : ABD :: IJKLM	IJKLN	11651
ABC : ABD :: 123	124	3839
ABC : ABD :: KJI	KJJ	3885
ABC : ABD :: 135	136	2487
ABC : ABD :: BCD	BCE	3250
ABC : ABD :: IJJKKK	IJJKKL	4424
ABC : ABD :: XYZ	XYA	9550
ABC : ABD :: 122333	122334	3175
ABC : ABD :: RSSTTT	RSSTTU	4851
ABC : ABD :: IJJKKK	IJJKKL	4095
ABC : ABD :: AABABC	AABABD	4716
ABC : ABD :: MRRJJJ	MRRJJK	4632
ABC : ABD :: 147	148	2150

Result:

- Date: Saturday January 14 2017 21:47:43 UTC
- Age: 27
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	11708
ABC : ABD :: BCA	BDA	9195
ABC : ABD :: AABABC	AABABD	43157
ABC : ABD :: IJKLM	IJKLN	7573
ABC : ABD :: 123	124	5558
ABC : ABD :: KJI	KJL	22756
ABC : ABD :: 135	136	4160
ABC : ABD :: BCD	BDE	6376
ABC : ABD :: IJJKKK	IJJLLL	20304
ABC : ABD :: XYZ	XY0	8958
ABC : ABD :: 122333	122444	9520
ABC : ABD :: RSSTTT	RSSUUU	6235
ABC : ABD :: IJJKKK	IJJLLL	6195
ABC : ABD :: AABABC	AABABD	18374
ABC : ABD :: MRRJJJ	MRRKKK	7494
ABC : ABD :: 147	148	13840

Result:

- Date: Sunday January 15 2017 22:28:27 UTC
- Age: 22
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	13296
ABC : ABD :: BCA	BDA	18938
ABC : ABD :: AABABC	AABABD	24628
ABC : ABD :: IJKLM	IJLMN	19072
ABC : ABD :: 123	124	4415
ABC : ABD :: KJI	LJI	9103
ABC : ABD :: 135	137	12626
ABC : ABD :: BCD	BCE	11543
ABC : ABD :: IJJKKK	IJJLLL	9184
ABC : ABD :: XYZ	XYA	8966
ABC : ABD :: 122333	122444	8888
ABC : ABD :: RSSTTT	RSSUUU	8519
ABC : ABD :: IJJKKK	IJJLLL	7651
ABC : ABD :: AABABC	AABABD	9940
ABC : ABD :: MRRJJJ	MRRIII	45999
ABC : ABD :: 147	1410	8875

Result:

- Date: Monday January 16 2017 10:29:22 UTC
- Age: 21
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	11277
ABC : ABD :: BCA	BCZ	11188
ABC : ABD :: AABABC	AABABCABD	21088
ABC : ABD :: IJKLM	IJKLN	18925
ABC : ABD :: 123	124	3193
ABC : ABD :: KJI	KJH	10638
ABC : ABD :: 135	137	6768
ABC : ABD :: BCD	BCE	4983
ABC : ABD :: IJJKKK	IJJKKKMMMM	6616
ABC : ABD :: XYZ	XYA	6206
ABC : ABD :: 122333	1224444	8235
ABC : ABD :: RSSTTT	RSSUUUU	23541
ABC : ABD :: IJJKKK	IJJLLLL	4786
ABC : ABD :: AABABC	AABABCABD	7185
ABC : ABD :: MRRJJJ	MRRSSSS	54646
ABC : ABD :: 147	141	9338

Result:

- Date: Thursday January 26 2017 17:12:48 UTC
- Age: 25
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	17609
ABC : ABD :: BCA	BDA	23333
ABC : ABD :: AABABC	AABBCD	37205
ABC : ABD :: IJKLM	IJKLN	26723
ABC : ABD :: 123	124	6169
ABC : ABD :: KJI	LJI	18242
ABC : ABD :: 135	137	10343
ABC : ABD :: BCD	BCE	7711
ABC : ABD :: IJJKKK	IJJLLL	18997
ABC : ABD :: XYZ	XYA	11460
ABC : ABD :: 122333	1224444	13296
ABC : ABD :: RSSTTT	RSSUUU	14901
ABC : ABD :: IJJKKK	IJJLLL	9304
ABC : ABD :: AABABC	AABABD	16606
ABC : ABD :: MRRJJJ	MRRKKK	30393
ABC : ABD :: 147	1410	18961

Result:

- Date: Friday January 27 2017 12:58:40 UTC
- Age: 67
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	20943
ABC : ABD :: BCA	BCB	55635
ABC : ABD :: AABABC	AABABD	63417
ABC : ABD :: IJKLM	IJKLN	16510
ABC : ABD :: 123	124	8395
ABC : ABD :: KJI	KJJ	32697
ABC : ABD :: 135	136	7370
ABC : ABD :: BCD	BCE	11837
ABC : ABD :: IJJKKK	IJJLLL	19661
ABC : ABD :: XYZ	XY.	38204
ABC : ABD :: 122333	122444	10991
ABC : ABD :: RSSTTT	RSSUUU	10933
ABC : ABD :: IJJKKK	IJJLLL	8942
ABC : ABD :: AABABC	AABABD	10925
ABC : ABD :: MRRJJJ	MRRKKK	10927
ABC : ABD :: 147	148	4689

Result:

- Date: Friday January 27 2017 13:41:24 UTC
- Age: 22
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	8807
ABC : ABD :: BCA	BCB	10416
ABC : ABD :: AABABC	AABACD	16054
ABC : ABD :: IJKLM	IJKMN	10228
ABC : ABD :: 123	124	4968
ABC : ABD :: KJI	KJJ	15528
ABC : ABD :: 135	136	4971
ABC : ABD :: BCD	BCE	4880
ABC : ABD :: IJJKKK	IJJKLL	14926
ABC : ABD :: XYZ	XYA	5355
ABC : ABD :: 122333	122344	8883
ABC : ABD :: RSSTTT	RSSTWW	15366
ABC : ABD :: IJJKKK	IJJKLL	6490
ABC : ABD :: AABABC	AABACD	6917
ABC : ABD :: MRRJJJ	MRRJKK	13551
ABC : ABD :: 147	148	5478

Result:

- Date: Friday January 27 2017 23:26:48 UTC
- Age: 22
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	27470
ABC : ABD :: BCA	BDA	26433
ABC : ABD :: AABABC	AABABD	11579
ABC : ABD :: IJKLM	IJKLM	11003
ABC : ABD :: 123	124	6890
ABC : ABD :: KJI	KJJ	7467
ABC : ABD :: 135	136	7039
ABC : ABD :: BCD	BDD	18332
ABC : ABD :: IJJKKK	IJJKKK	6891
ABC : ABD :: XYZ	XYA	8189
ABC : ABD :: 122333	122333	11164
ABC : ABD :: RSSTTT	RSSTTT	5368
ABC : ABD :: IJJKKK	IJJKKK	7312
ABC : ABD :: AABABC	AABABD	10664
ABC : ABD :: MRRJJJ	MRRJJJ	5185
ABC : ABD :: 147	148	4324

Result:

- Date: Wednesday February 01 2017 17:05:32 UTC
- Age: 21
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	10949
ABC : ABD :: BCA	BDA	15484
ABC : ABD :: AABABC	AABABD	9188
ABC : ABD :: IJKLM	IJKLN	6894
ABC : ABD :: 123	124	3685
ABC : ABD :: KJI	KJG	27173
ABC : ABD :: 135	136	6636
ABC : ABD :: BCD	BDC	33497
ABC : ABD :: IJJKKK	IJJKKM	11046
ABC : ABD :: XYZ	XYA	8390
ABC : ABD :: 122333	122334	7183
ABC : ABD :: RSSTTT	RSSTTU	8270
ABC : ABD :: IJJKKK	IJKKL	4461
ABC : ABD :: AABABC	AABABD	6512
ABC : ABD :: MRRJJJ	MRRJJK	10759
ABC : ABD :: 147	148	4263

Result:

- Date: Wednesday February 01 2017 17:09:25 UTC
- Age: 26
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	13519
ABC : ABD :: BCA	BDA	16917
ABC : ABD :: AABABC	AABABD	9201
ABC : ABD :: IJKLM	IJKLN	23954
ABC : ABD :: 123	124	6362
ABC : ABD :: KJI	LJI	34550
ABC : ABD :: 135	136	8417
ABC : ABD :: BCD	BCE	10189
ABC : ABD :: IJJKKK	IJJLLL	10422
ABC : ABD :: XYZ	XYA	9418
ABC : ABD :: 122333	122444	4729
ABC : ABD :: RSSTTT	RSSUUU	8026
ABC : ABD :: IJJKKK	IJJLLL	7223
ABC : ABD :: AABABC	AABABD	6157
ABC : ABD :: MRRJJJ	MRRKKK	12545
ABC : ABD :: 147	148	5652

Result:

- Date: Wednesday February 01 2017 17:09:58 UTC
- Age: 24
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	12530
ABC : ABD :: BCA	BCB	11233
ABC : ABD :: AABABC	AACABD	10508
ABC : ABD :: IJKLM	IJLLM	12200
ABC : ABD :: 123	124	4016
ABC : ABD :: KJI	KJJ	5216
ABC : ABD :: 135	136	4183
ABC : ABD :: BCD	BCE	4791
ABC : ABD :: IJJKKK	IJKKKL	7442
ABC : ABD :: XYZ	XYA	5277
ABC : ABD :: 122333	123334	6209
ABC : ABD :: RSSTTT	RSTTTU	8359
ABC : ABD :: IJJKKK	IJKKKL	8186
ABC : ABD :: AABABC	AACABD	7851
ABC : ABD :: MRRJJJ	MRSJJK	6296
ABC : ABD :: 147	148	5647

Result:

- Date: Wednesday February 01 2017 17:11:28 UTC
- Age: 21
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	24640
ABC : ABD :: BCA	BDA	12560
ABC : ABD :: AABABC	AABABD	9886
ABC : ABD :: IJKLM	IJLKM	41982
ABC : ABD :: 123	124	4419
ABC : ABD :: KJI	KJI	25280
ABC : ABD :: 135	135	2687
ABC : ABD :: BCD	BCD	3403
ABC : ABD :: IJJKKK	IJJKKK	3114
ABC : ABD :: XYZ	XYZ	3411
ABC : ABD :: 122333	122333	4466
ABC : ABD :: RSSTTT	RSSTTT	4055
ABC : ABD :: IJJKKK	IJJKKK	3567
ABC : ABD :: AABABC	AABABC	3071
ABC : ABD :: MRRJJJ	MRRJJJ	3211
ABC : ABD :: 147	147	2443

Result:

- Date: Wednesday February 01 2017 17:14:02 UTC
- Age: 23
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	12202
ABC : ABD :: BCA	BCB	15756
ABC : ABD :: AABABC	AABABD	30659
ABC : ABD :: IJKLM	IJKLN	9143
ABC : ABD :: 123	124	4759
ABC : ABD :: KJI	KJL	4892
ABC : ABD :: 135	136	3580
ABC : ABD :: BCD	BCE	6241
ABC : ABD :: IJJKKK	IIJJKKL	7164
ABC : ABD :: XYZ	XYA	5453
ABC : ABD :: 122333	122334	4066
ABC : ABD :: RSSTTT	RSSTTU	4991
ABC : ABD :: IJJKKK	IJJKKL	5457
ABC : ABD :: AABABC	AABABD	4404
ABC : ABD :: MRRJJJ	MRRJJK	5019
ABC : ABD :: 147	148	7256

Result:

- Date: Wednesday February 01 2017 17:14:21 UTC
- Age: 24
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	11780
ABC : ABD :: BCA	BCB	19005
ABC : ABD :: AABABC	AABACD	17475
ABC : ABD :: IJKLM	IJKLN	18445
ABC : ABD :: 123	124	5623
ABC : ABD :: KJI	LJI	14819
ABC : ABD :: 135	137	4725
ABC : ABD :: BCD	BCE	6637
ABC : ABD :: IJJKKK	IJJLLL	9473
ABC : ABD :: XYZ	XYA	8908
ABC : ABD :: 122333	122444	8252
ABC : ABD :: RSSTTT	RSSUUU	7655
ABC : ABD :: IJJKKK	IJJLLL	13052
ABC : ABD :: AABABC	AABACD	11440
ABC : ABD :: MRRJJJ	MRRKKK	16723
ABC : ABD :: 147	148	5953

Result:

- Date: Wednesday February 01 2017 17:17:51 UTC
- Age: 18
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	ADE	18055
ABC : ABD :: BCA	BCB	39511
ABC : ABD :: AABABC	AABBAD	23157
ABC : ABD :: IJKLM	IJKLMN	19895
ABC : ABD :: 123	124	6214
ABC : ABD :: KJI	KJJ	35801
ABC : ABD :: 135	136	4800
ABC : ABD :: BCD	BCE	5706
ABC : ABD :: IJJKKK	IJJKKK	5011
ABC : ABD :: XYZ	XYA	11980
ABC : ABD :: 122333	1122444	7672
ABC : ABD :: RSSTTT	RSSUUU	20793
ABC : ABD :: IJJKKK	IJJLLL	21970
ABC : ABD :: AABABC	AABABD	8052
ABC : ABD :: MRRJJJ	MRRKKK	6495
ABC : ABD :: 147	148	8401

Result:

- Date: Wednesday February 01 2017 17:22:08 UTC
- Age: 25
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	12928
ABC : ABD :: BCA	BCB	26200
ABC : ABD :: AABABC	AABABD	25984
ABC : ABD :: IJKLM	IJKLN	9526
ABC : ABD :: 123	124	3710
ABC : ABD :: KJI	KJK	9142
ABC : ABD :: 135	136	4328
ABC : ABD :: BCD	BCE	6028
ABC : ABD :: IJJKKK	IJJKKL	8104
ABC : ABD :: XYZ	XYA	3871
ABC : ABD :: 122333	122334	4589
ABC : ABD :: RSSTTT	RSSTTU	11572
ABC : ABD :: IJJKKK	IJJKKL	7356
ABC : ABD :: AABABC	AABABD	10230
ABC : ABD :: MRRJJJ	MRRJJK	8004
ABC : ABD :: 147	148	3097

Result:

- Date: Wednesday February 01 2017 17:25:57 UTC
- Age: 25
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	8904
ABC : ABD :: BCA	BDA	18194
ABC : ABD :: AABABC	AABABD	15603
ABC : ABD :: IJKLM	IJKLN	9539
ABC : ABD :: 123	124	3354
ABC : ABD :: KJI	LJI	22057
ABC : ABD :: 135	137	7870
ABC : ABD :: BCD	BCE	5040
ABC : ABD :: IJJKKK	IJJLLL	16576
ABC : ABD :: XYZ	XY	6946
ABC : ABD :: 122333	122444	8592
ABC : ABD :: RSSTTT	RSSUUU	11611
ABC : ABD :: IJJKKK	IJJLLL	6655
ABC : ABD :: AABABC	AABABD	7937
ABC : ABD :: MRRJJJ	MRRJJJ	9786
ABC : ABD :: 147	14A	17713

Result:

- Date: Wednesday February 01 2017 17:26:03 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	9997
ABC : ABD :: BCA	BDA	16256
ABC : ABD :: AABABC	AABABD	15492
ABC : ABD :: IJKLM	IJKLN	14361
ABC : ABD :: 123	124	3645
ABC : ABD :: KJI	KJJ	8060
ABC : ABD :: 135	136	8036
ABC : ABD :: BCD	BCE	9432
ABC : ABD :: IJJKKK	IJJKKL	13213
ABC : ABD :: XYZ	XYA	14526
ABC : ABD :: 122333	122334	7395
ABC : ABD :: RSSTTT	RSSTTU	7431
ABC : ABD :: IJJKKK	IJJKKL	6484
ABC : ABD :: AABABC	AABABD	7655
ABC : ABD :: MRRJJJ	MRRJJK	6145
ABC : ABD :: 147	148	3898

Result:

- Date: Wednesday February 01 2017 17:43:13 UTC
- Age: 21
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	6367
ABC : ABD :: BCA	BDA	25907
ABC : ABD :: AABABC	AACABD	16107
ABC : ABD :: IJKLM	IJLLM	12552
ABC : ABD :: 123	124	3957
ABC : ABD :: KJI	LJI	18573
ABC : ABD :: 135	136	5451
ABC : ABD :: BCD	BCE	13112
ABC : ABD :: IJJKKK	IJKKKL	19793
ABC : ABD :: XYZ	XYA	5822
ABC : ABD :: 122333	123334	7427
ABC : ABD :: RSSTTT	RSTTTU	10900
ABC : ABD :: IJJKKK	IJKKKL	6149
ABC : ABD :: AABABC	AACABD	9326
ABC : ABD :: MRRJJJ	MRSJJK	18177
ABC : ABD :: 147	148	21250

Result:

- Date: Wednesday February 01 2017 17:43:35 UTC
- Age: 21
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	14650
ABC : ABD :: BCA	BDA	10056
ABC : ABD :: AABABC	AABABD	19760
ABC : ABD :: IJKLM	IJKLN	19622
ABC : ABD :: 123	124	3630
ABC : ABD :: KJI	LJI	9252
ABC : ABD :: 135	137	12381
ABC : ABD :: BCD	BCE	31000
ABC : ABD :: IJJKKK	IJJLLL	7506
ABC : ABD :: XYZ	XYA	7200
ABC : ABD :: 122333	122444	6446
ABC : ABD :: RSSTTT	RSSUUU	27380
ABC : ABD :: IJJKKK	IJJLLL	11655
ABC : ABD :: AABABC	AABABD	9126
ABC : ABD :: MRRJJJ	:(	68133
ABC : ABD :: 147	1410	9986

Result:

- Date: Wednesday February 01 2017 17:47:07 UTC
- Age: 25
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	21270
ABC : ABD :: BCA	BCB	58722
ABC : ABD :: AABABC	AACABD	24809
ABC : ABD :: IJKLM	IJKLN	21834
ABC : ABD :: 123	124	9113
ABC : ABD :: KJI	KJJ	31531
ABC : ABD :: 135	137	15981
ABC : ABD :: BCD	BCC	23049
ABC : ABD :: IJJKKK	IJJLLL	22396
ABC : ABD :: XYZ	XYZS	35735
ABC : ABD :: 122333	122444	16452
ABC : ABD :: RSSTTT	RSSUUU	22362
ABC : ABD :: IJJKKK	IJJLLL	9462
ABC : ABD :: AABABC	AABACD	29083
ABC : ABD :: MRRJJJ	MRRKKK	12945
ABC : ABD :: 147	1410	37495

Result:

- Date: Wednesday February 01 2017 17:47:52 UTC
- Age: 22
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJF	20578
ABC : ABD :: BCA	BCZS	47469
ABC : ABD :: AABABC	ABE	32944
ABC : ABD :: IJKLM	IJKLO	25060
ABC : ABD :: 123	124	16210
ABC : ABD :: KJI	KJH	16931
ABC : ABD :: 135	137	13632
ABC : ABD :: BCD	BCE	8872
ABC : ABD :: IJJKKK	IJJKKF	26918
ABC : ABD :: XYZ	XYA	21615
ABC : ABD :: 122333	122334	8654
ABC : ABD :: RSSTTT	RSSTTU	13180
ABC : ABD :: IJJKKK	IJJKKL	16494
ABC : ABD :: AABABC	AABABE	48649
ABC : ABD :: MRRJJJ	MRRJJH	14513
ABC : ABD :: 147	141	14992

Result:

- Date: Wednesday February 01 2017 17:49:19 UTC
- Age: 20
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	18134
ABC : ABD :: BCA	BCB	19512
ABC : ABD :: AABABC	AABABD	36044
ABC : ABD :: IJKLM	IJKLN	22179
ABC : ABD :: 123	124	9271
ABC : ABD :: KJI	KJK	6977
ABC : ABD :: 135	136	4562
ABC : ABD :: BCD	BCE	11161
ABC : ABD :: IJJKKK	IJJKKL	13031
ABC : ABD :: XYZ	XYA	8311
ABC : ABD :: 122333	122334	21927
ABC : ABD :: RSSTTT	RSSTTU	8081
ABC : ABD :: IJJKKK	IJJKKL	7529
ABC : ABD :: AABABC	AABABD	9742
ABC : ABD :: MRRJJJ	MRRJJK	8142
ABC : ABD :: 147	148	6239

Result:

- Date: Wednesday February 01 2017 17:52:48 UTC
- Age: 23
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	10033
ABC : ABD :: BCA	BDA	50919
ABC : ABD :: AABABC	AABABD	19095
ABC : ABD :: IJKLM	IJKLN	14539
ABC : ABD :: 123	124	4752
ABC : ABD :: KJI	LJI	12019
ABC : ABD :: 135	136	4804
ABC : ABD :: BCD	BCE	10498
ABC : ABD :: IJJKKK	IJJLLL	11036
ABC : ABD :: XYZ	XYA	9250
ABC : ABD :: 122333	122444	8388
ABC : ABD :: RSSTTT	RSSUUU	19215
ABC : ABD :: IJJKKK	IJJLLL	8090
ABC : ABD :: AABABC	AABABD	13131
ABC : ABD :: MRRJJJ	MRRLLL	12276
ABC : ABD :: 147	148	6318

Result:

- Date: Wednesday February 01 2017 17:58:50 UTC
- Age: 21
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	18119
ABC : ABD :: BCA	BDA	17172
ABC : ABD :: AABABC	AABABD	13856
ABC : ABD :: IJKLM	IJKLN	13652
ABC : ABD :: 123	124	4425
ABC : ABD :: KJI	LJI	12139
ABC : ABD :: 135	136	4580
ABC : ABD :: BCD	BDE	8410
ABC : ABD :: IJJKKK	IJJLLL	6697
ABC : ABD :: XYZ	XYA	4540
ABC : ABD :: 122333	122444	5214
ABC : ABD :: RSSTTT	RSSUUU	8032
ABC : ABD :: IJJKKK	IJJLLL	6055
ABC : ABD :: AABABC	AABABD	6132
ABC : ABD :: MRRJJJ	MSSJJJ	12550
ABC : ABD :: 147	148	4565

Result:

- Date: Wednesday February 01 2017 17:59:55 UTC
- Age: 23
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	14893
ABC : ABD :: BCA	BCB	23234
ABC : ABD :: AABABC	AABABD	15678
ABC : ABD :: IJKLM	IJKMN	9509
ABC : ABD :: 123	124	3361
ABC : ABD :: KJI	KJJ	18442
ABC : ABD :: 135	136	3909
ABC : ABD :: BCD	BCE	7567
ABC : ABD :: IJJKKK	IJJKKL	6845
ABC : ABD :: XYZ	XYA	6270
ABC : ABD :: 122333	122334	6836
ABC : ABD :: RSSTTT	RSSTTU	4973
ABC : ABD :: IJJKKK	IJJKKL	7974
ABC : ABD :: AABABC	AABABD	4847
ABC : ABD :: MRRJJJ	MRRJJK	8972
ABC : ABD :: 147	148	2810

Result:

- Date: Wednesday February 01 2017 18:23:51 UTC
- Age: 19
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	8446
ABC : ABD :: BCA	BCB	11092
ABC : ABD :: AABABC	AABABD	17417
ABC : ABD :: IJKLM	IJKLN	12611
ABC : ABD :: 123	124	6562
ABC : ABD :: KJI	KJJ	6819
ABC : ABD :: 135	136	3680
ABC : ABD :: BCD	BCE	5787
ABC : ABD :: IJJKKK	IJJKKL	6903
ABC : ABD :: XYZ	XYA	6956
ABC : ABD :: 122333	122334	6398
ABC : ABD :: RSSTTT	RSSTTU	8857
ABC : ABD :: IJJKKK	IJJKKL	7308
ABC : ABD :: AABABC	AABABD	8692
ABC : ABD :: MRRJJJ	MRRJJK	10360
ABC : ABD :: 147	148	3494

Result:

- Date: Wednesday February 01 2017 19:01:42 UTC
- Age: 31
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	29694
ABC : ABD :: BCA	BCB	24914
ABC : ABD :: AABABC	AABABD	22848
ABC : ABD :: IJKLM	IJKLN	12492
ABC : ABD :: 123	124	47918
ABC : ABD :: KJI	LJI	88186
ABC : ABD :: 135	136	10442
ABC : ABD :: BCD	BCE	10218
ABC : ABD :: IJJKKK	IJJLLL	88715
ABC : ABD :: XYZ	XYA	28735
ABC : ABD :: 122333	122444	74218
ABC : ABD :: RSSTTT	RSSUUU	17606
ABC : ABD :: IJJKKK	IJJLLL	15499
ABC : ABD :: AABABC	AABABD	14480
ABC : ABD :: MRRJJJ	MSSJJJ	42900
ABC : ABD :: 147	148	12001

Result:

- Date: Wednesday February 01 2017 19:04:19 UTC
- Age: 21
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	14689
ABC : ABD :: BCA	BDA	24641
ABC : ABD :: AABABC	AABABD	14517
ABC : ABD :: IJKLM	IJLP	33019
ABC : ABD :: 123	124	3746
ABC : ABD :: KJI	LJI	12128
ABC : ABD :: 135	137	33109
ABC : ABD :: BCD	BCE	8284
ABC : ABD :: IJJKKK	IJJLLL	11323
ABC : ABD :: XYZ	XYA	10687
ABC : ABD :: 122333	122444	7360
ABC : ABD :: RSSTTT	RSSUUU	12099
ABC : ABD :: IJJKKK	IJJLLL	9444
ABC : ABD :: AABABC	AABABD	9497
ABC : ABD :: MRRJJJ	MRRKKK	27402
ABC : ABD :: 147	148	8235

Result:

- Date: Wednesday February 01 2017 19:42:32 UTC
- Age: 23
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	9593
ABC : ABD :: BCA	BDA	6681
ABC : ABD :: AABABC	AABABD	17378
ABC : ABD :: IJKLM	IJKMN	14004
ABC : ABD :: 123	124	6823
ABC : ABD :: KJI	JJI	10478
ABC : ABD :: 135	136	7727
ABC : ABD :: BCD	BCE	9502
ABC : ABD :: IJJKKK	IJJKLL	11823
ABC : ABD :: XYZ	XYA	7144
ABC : ABD :: 122333	122344	8326
ABC : ABD :: RSSTTT	RSTUU	6150
ABC : ABD :: IJJKKK	IJKLL	7251
ABC : ABD :: AABABC	AABACD	6789
ABC : ABD :: MRRJJJ	MRJII	7020
ABC : ABD :: 147	148	4165

Result:

- Date: Wednesday February 01 2017 19:44:54 UTC
- Age: 25
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	8671
ABC : ABD :: BCA	BDA	19076
ABC : ABD :: AABABC	AABABD	20305
ABC : ABD :: IJKLM	IJKLN	19560
ABC : ABD :: 123	124	8719
ABC : ABD :: KJI	LJI	16514
ABC : ABD :: 135	137	68012
ABC : ABD :: BCD	BED	62938
ABC : ABD :: IJJKKK	IJJLLL	23493
ABC : ABD :: XYZ	XYA	6341
ABC : ABD :: 122333	122444	8029
ABC : ABD :: RSSTTT	RSSUUU	7254
ABC : ABD :: IJJKKK	IJJLLL	8105
ABC : ABD :: AABABC	AABABD	6708
ABC : ABD :: MRRJJJ	?	17494
ABC : ABD :: 147	?	3062

Result:

- Date: Wednesday February 01 2017 19:48:34 UTC
- Age: 19
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	11339
ABC : ABD :: BCA	BDA	26919
ABC : ABD :: AABABC	AABABD	20360
ABC : ABD :: IJKLM	IJKLN	9724
ABC : ABD :: 123	124	4318
ABC : ABD :: KJI	LJI	10594
ABC : ABD :: 135	137	7247
ABC : ABD :: BCD	BCE	8273
ABC : ABD :: IJJKKK	IJJLLL	7600
ABC : ABD :: XYZ	XYA	11853
ABC : ABD :: 122333	122444	6139
ABC : ABD :: RSSTTT	RSSUUU	6672
ABC : ABD :: IJJKKK	IJJLLL	7621
ABC : ABD :: AABABC	AABABD	6291
ABC : ABD :: MRRJJJ	MRRLLL	10440
ABC : ABD :: 147	149	10128

Result:

- Date: Wednesday February 01 2017 19:50:04 UTC
- Age: 72
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	12087
ABC : ABD :: BCA	BCB	26900
ABC : ABD :: AABABC	AABABD	17040
ABC : ABD :: IJKLM	IJKLN	12754
ABC : ABD :: 123	124	10402
ABC : ABD :: KJI	KJJ	11815
ABC : ABD :: 135	136	9319
ABC : ABD :: BCD	BCE	14685
ABC : ABD :: IJJKKK	IJJKKL	13601
ABC : ABD :: XYZ	XYA	8774
ABC : ABD :: 122333	122334	10446
ABC : ABD :: RSSTTT	RSSTTU	11694
ABC : ABD :: IJJKKK	OJJKKL	9902
ABC : ABD :: AABABC	AABABD	13430
ABC : ABD :: MRRJJJ	MRRJJK	10526
ABC : ABD :: 147	148	7501

Result:

- Date: Wednesday February 01 2017 19:58:13 UTC
- Age: 23
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	22882
ABC : ABD :: BCA	BCB	18270
ABC : ABD :: AABABC	AABABD	23823
ABC : ABD :: IJKLM	IJKLN	14170
ABC : ABD :: 123	124	5479
ABC : ABD :: KJI	KJJ	27526
ABC : ABD :: 135	136	6623
ABC : ABD :: BCD	BCE	8487
ABC : ABD :: IJJKKK	IJJKKL	9977
ABC : ABD :: XYZ	XYA	19824
ABC : ABD :: 122333	122334	7176
ABC : ABD :: RSSTTT	RSSTTU	12240
ABC : ABD :: IJJKKK	IJJKKL	5214
ABC : ABD :: AABABC	AABABD	6918
ABC : ABD :: MRRJJJ	MRRJJK	6729
ABC : ABD :: 147	148	3776

Result:

- Date: Wednesday February 01 2017 20:03:55 UTC
- Age: 28
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	21253
ABC : ABD :: BCA	BCB	12020
ABC : ABD :: AABABC	AABABD	32957
ABC : ABD :: IJKLM	IJKLN	34314
ABC : ABD :: 123	124	6856
ABC : ABD :: KJI	KJJ	8382
ABC : ABD :: 135	136	5247
ABC : ABD :: BCD	BCE	8594
ABC : ABD :: IJJKKK	IJJKKL	11274
ABC : ABD :: XYZ	XYA	9673
ABC : ABD :: 122333	122334	3694
ABC : ABD :: RSSTTT	RSSTTU	7331
ABC : ABD :: IJJKKK	IJJKKL	5615
ABC : ABD :: AABABC	AABABD	7825
ABC : ABD :: MRRJJJ	MRRJJK	9536
ABC : ABD :: 147	148	3998

Result:

- Date: Wednesday February 01 2017 20:09:10 UTC
- Age: 31
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	19714
ABC : ABD :: BCA	BDA	27305
ABC : ABD :: AABABC	AABABD	22628
ABC : ABD :: IJKLM	IJKLN	22144
ABC : ABD :: 123	124	7962
ABC : ABD :: KJI	LJI	51049
ABC : ABD :: 135	136	16810
ABC : ABD :: BCD	BCE	15240
ABC : ABD :: IJJKKK	IJJLLL	13999
ABC : ABD :: XYZ	XYA	10232
ABC : ABD :: 122333	122444	10664
ABC : ABD :: RSSTTT	RSSUUU	11287
ABC : ABD :: IJJKKK	IJJLLL	10229
ABC : ABD :: AABABC	AABABD	11475
ABC : ABD :: MRRJJJ	MRRKKK	14905
ABC : ABD :: 147	148	8260

Result:

- Date: Wednesday February 01 2017 20:30:00 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	13294
ABC : ABD :: BCA	BCB	11466
ABC : ABD :: AABABC	AABABD	16565
ABC : ABD :: IJKLM	IJKLM	12250
ABC : ABD :: 123	124	6496
ABC : ABD :: KJI	KJI	14340
ABC : ABD :: 135	145	9352
ABC : ABD :: BCD	BDD	8040
ABC : ABD :: IJJKKK	IJJKKK	5007
ABC : ABD :: XYZ	XYZ	6279
ABC : ABD :: 122333	122444	6238
ABC : ABD :: RSSTTT	RSSTTT	4752
ABC : ABD :: IJJKKK	IJJKKK	4840
ABC : ABD :: AABABC	AABABD	7798
ABC : ABD :: MRRJJJ	MRJJ	4583
ABC : ABD :: 147	147	6092

Result:

- Date: Wednesday February 01 2017 20:37:38 UTC
- Age: 23
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	39505
ABC : ABD :: BCA	BDA	46348
ABC : ABD :: AABABC	AABABD	41253
ABC : ABD :: IJKLM	IJKLN	22753
ABC : ABD :: 123	124	6536
ABC : ABD :: KJI	LJI	14151
ABC : ABD :: 135	136	12040
ABC : ABD :: BCD	BCE	10639
ABC : ABD :: IJJKKK	IJJKKL	12048
ABC : ABD :: XYZ	XYA	30566
ABC : ABD :: 122333	122334	7141
ABC : ABD :: RSSTTT	RSSTTU	7522
ABC : ABD :: IJJKKK	IJJKKL	8032
ABC : ABD :: AABABC	AABABD	21039
ABC : ABD :: MRRJJJ	MRRJJW	84508
ABC : ABD :: 147	14711	53820

Result:

- Date: Wednesday February 01 2017 20:49:01 UTC
- Age: 23
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	10302
ABC : ABD :: BCA	DBA	7545
ABC : ABD :: AABABC	AABABD	9755
ABC : ABD :: IJKLM	IJKLN	8844
ABC : ABD :: 123	124	6042
ABC : ABD :: KJI	KJH	28075
ABC : ABD :: 135	136	7828
ABC : ABD :: BCD	BCE	5291
ABC : ABD :: IJJKKK	IJJLLL	10604
ABC : ABD :: XYZ	XYA	6180
ABC : ABD :: 122333	1224444	9344
ABC : ABD :: RSSTTT	RSSUUU	9197
ABC : ABD :: IJJKKK	IJJLLL	7180
ABC : ABD :: AABABC	AABABD	5608
ABC : ABD :: MRRJJJ	MRRKKK	11814
ABC : ABD :: 147	148	5747

Result:

- Date: Wednesday February 01 2017 20:50:02 UTC
- Age: 22
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	21043
ABC : ABD :: BCA	BDA	48194
ABC : ABD :: AABABC	AABABD	22754
ABC : ABD :: IJKLM	VRJTN	399996
ABC : ABD :: 123	124	7200
ABC : ABD :: KJI	LJI	56775
ABC : ABD :: 135	1416	57686
ABC : ABD :: BCD	BDH	26185
ABC : ABD :: IJJKKK	IJJKKKK	29705
ABC : ABD :: XYZ	XYA	13743
ABC : ABD :: 122333	1223333	12803
ABC : ABD :: RSSTTT	RSSTTTT	12846
ABC : ABD :: IJJKKK	IJJKKKK	15652
ABC : ABD :: AABABC	AABABD	14457
ABC : ABD :: MRRJJJ	MRRJJJJ	10943
ABC : ABD :: 147	1410	25894

Result:

- Date: Wednesday February 01 2017 20:51:40 UTC
- Age: 21
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	16872
ABC : ABD :: BCA	BCB	13812
ABC : ABD :: AABABC	AACABD	18004
ABC : ABD :: IJKLM	IJLLM	8024
ABC : ABD :: 123	124	3933
ABC : ABD :: KJI	KJJ	14830
ABC : ABD :: 135	136	4887
ABC : ABD :: BCD	BCE	5409
ABC : ABD :: IJJKKK	IJKKKL	8451
ABC : ABD :: XYZ	XYA	6086
ABC : ABD :: 122333	123334	5967
ABC : ABD :: RSSTTT	RSTTTU	9706
ABC : ABD :: IJJKKK	IJKKKL	7292
ABC : ABD :: AABABC	AACABD	6528
ABC : ABD :: MRRJJJ	MRSJJK	7763
ABC : ABD :: 147	148	3302

Result:

- Date: Wednesday February 01 2017 20:52:39 UTC
- Age: 18
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	22147
ABC : ABD :: BCA	BCB	23027
ABC : ABD :: AABABC	AABABD	11582
ABC : ABD :: IJKLM	IJKLN	11404
ABC : ABD :: 123	124	4491
ABC : ABD :: KJI	KJJ	11072
ABC : ABD :: 135	136	6135
ABC : ABD :: BCD	BCE	5691
ABC : ABD :: IJJKKK	IJJKKL	7186
ABC : ABD :: XYZ	XYA	8413
ABC : ABD :: 122333	122334	6150
ABC : ABD :: RSSTTT	RSSTTU	5959
ABC : ABD :: IJJKKK	IJJKKL	4589
ABC : ABD :: AABABC	AABABD	4608
ABC : ABD :: MRRJJJ	MRRJJK	5622
ABC : ABD :: 147	148	5276

Result:

- Date: Wednesday February 01 2017 21:16:00 UTC
- Age: 21
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	21467
ABC : ABD :: BCA	BDA	24263
ABC : ABD :: AABABC	AABABD	43361
ABC : ABD :: IJKLM	IJKLN	20825
ABC : ABD :: 123	124	6626
ABC : ABD :: KJI	KJH	42076
ABC : ABD :: 135	137	11769
ABC : ABD :: BCD	BCE	10531
ABC : ABD :: IJJKKK	IJJLLL	1932229
ABC : ABD :: XYZ	XYA	12774
ABC : ABD :: 122333	1122444	12973
ABC : ABD :: RSSTTT	RSSUUU	115286
ABC : ABD :: IJJKKK	IJJLLL	11653
ABC : ABD :: AABABC	AABABD	16310
ABC : ABD :: MRRJJJ	MRRPPP	119159
ABC : ABD :: 147	1410	85610

Result:

- Date: Wednesday February 01 2017 21:25:12 UTC
- Age: 19
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	33475
ABC : ABD :: BCA	BDA	64007
ABC : ABD :: AABABC	AABABD	22223
ABC : ABD :: IJKLM	IJKLN	57824
ABC : ABD :: 123	124	12878
ABC : ABD :: KJI	MJI	35437
ABC : ABD :: 135	136	57884
ABC : ABD :: BCD	BCE	92921
ABC : ABD :: IJJKKK	IJJLLL	32232
ABC : ABD :: XYZ	XYä	60588
ABC : ABD :: 122333	122444	23518
ABC : ABD :: RSSTTT	RSSUUU	14221
ABC : ABD :: IJJKKK	IJJLLL	14336
ABC : ABD :: AABABC	AABABD	24525
ABC : ABD :: MRRJJJ	I BELIEVE I CAN FLYYYYY	65813
ABC : ABD :: 147	1410	150393

Result:

- Date: Wednesday February 01 2017 21:31:21 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJF	12721
ABC : ABD :: BCA	BDA	9518
ABC : ABD :: AABABC	AABABD	12653
ABC : ABD :: IJKLM	IJKLN	17308
ABC : ABD :: 123	124	2595
ABC : ABD :: KJI	LJI	21154
ABC : ABD :: 135	137	4719
ABC : ABD :: BCD	BCE	10252
ABC : ABD :: IJJKKK	IJJLLLL	8121
ABC : ABD :: XYZ	XYA	7696
ABC : ABD :: 122333	1224444	5091
ABC : ABD :: RSSTTT	RSSUUUU	10889
ABC : ABD :: IJJKKK	IJJLLLL	5504
ABC : ABD :: AABABC	AABABD	10034
ABC : ABD :: MRRJJJ	MRRKKKK	18978
ABC : ABD :: 147	1411	20698

Result:

- Date: Wednesday February 01 2017 21:31:44 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	8018
ABC : ABD :: BCA	BAC	67151
ABC : ABD :: AABABC	AABB	7119
ABC : ABD :: IJKLM	OPQRS	9025
ABC : ABD :: 123	456	6059
ABC : ABD :: KJI	LMN	5915
ABC : ABD :: 135	136	6020
ABC : ABD :: BCD	EFG	6093
ABC : ABD :: IJJKKK	LLLLMMMMMNNNNNN	17586
ABC : ABD :: XYZ	ABC	4775
ABC : ABD :: 122333	444455555666666	13992
ABC : ABD :: RSSTTT	UUUUVVVVVWWWWWW	26976
ABC : ABD :: IJJKKK	LLLLMMMMMNNNNNN	12717
ABC : ABD :: AABABC	ABBA	4014
ABC : ABD :: MRRJJJ	MDR	3956
ABC : ABD :: 147	148	5553

Result:

- Date: Wednesday February 01 2017 21:33:46 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	15227
ABC : ABD :: BCA	BCB	18381
ABC : ABD :: AABABC	AACABD	26746
ABC : ABD :: IJKLM	IJLLM	28962
ABC : ABD :: 123	124	5422
ABC : ABD :: KJI	KJJ	10338
ABC : ABD :: 135	136	4234
ABC : ABD :: BCD	BCE	7285
ABC : ABD :: IJJKKK	IJKKKL	8941
ABC : ABD :: XYZ	XYA	5071
ABC : ABD :: 122333	123334	8692
ABC : ABD :: RSSTTT	RSTTTU	11714
ABC : ABD :: IJJKKK	IJKKKL	7909
ABC : ABD :: AABABC	AACABD	8087
ABC : ABD :: MRRJJJ	MRSJJK	12492
ABC : ABD :: 147	148	3440

Result:

- Date: Wednesday February 01 2017 21:41:46 UTC
- Age: 26
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	12295
ABC : ABD :: BCA	BCB	22500
ABC : ABD :: AABABC	AACABD	7821
ABC : ABD :: IJKLM	IJKLN	18036
ABC : ABD :: 123	124	3981
ABC : ABD :: KJI	KJJ	6852
ABC : ABD :: 135	136	2853
ABC : ABD :: BCD	BCE	5021
ABC : ABD :: IJJKKK	IJKKKL	7653
ABC : ABD :: XYZ	XYA	4341
ABC : ABD :: 122333	123334	4714
ABC : ABD :: RSSTTT	RSTTTU	5414
ABC : ABD :: IJJKKK	IJKKKL	8829
ABC : ABD :: AABABC	AACABD	5262
ABC : ABD :: MRRJJJ	MRSJJK	8645
ABC : ABD :: 147	148	4053

Result:

- Date: Wednesday February 01 2017 22:14:01 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	15765
ABC : ABD :: BCA	DAB	62291
ABC : ABD :: AABABC	AABABD	18051
ABC : ABD :: IJKLM	IJLMN	28241
ABC : ABD :: 123	124	7393
ABC : ABD :: KJI	LKI	33011
ABC : ABD :: 135	136	5659
ABC : ABD :: BCD	BCE	6997
ABC : ABD :: IJJKKK	IJJLLL	28826
ABC : ABD :: XYZ	XYA	21925
ABC : ABD :: 122333	122444	5772
ABC : ABD :: RSSTTT	RSSUUU	7934
ABC : ABD :: IJJKKK	IJJLLL	6008
ABC : ABD :: AABABC	AAABABD	9607
ABC : ABD :: MRRJJJ	MRRKKK	106102
ABC : ABD :: 147	369	61982

Result:

- Date: Wednesday February 01 2017 22:18:58 UTC
- Age: 18
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	17334
ABC : ABD :: BCA	BCB	14359
ABC : ABD :: AABABC	AABABD	29527
ABC : ABD :: IJKLM	IJKLM	6546
ABC : ABD :: 123	123	5630
ABC : ABD :: KJI	KJI	3316
ABC : ABD :: 135	135	4389
ABC : ABD :: BCD	BCD	40136
ABC : ABD :: IJJKKK	IJJKKK	6644
ABC : ABD :: XYZ	XYZ	4578
ABC : ABD :: 122333	122333	4879
ABC : ABD :: RSSTTT	RSSTTT	4592
ABC : ABD :: IJJKKK	IJJKKK	4395
ABC : ABD :: AABABC	AABABD	10306
ABC : ABD :: MRRJJJ	MRRJJJ	4916
ABC : ABD :: 147	147	4301

Result:

- Date: Wednesday February 01 2017 22:21:21 UTC
- Age: 23
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	3996
ABC : ABD :: BCA	BCB	4578
ABC : ABD :: AABABC	AACABD	5238
ABC : ABD :: IJKLM	AJLLM	8311
ABC : ABD :: 123	124	4813
ABC : ABD :: KJI	KJJ	3361
ABC : ABD :: 135	136	5420
ABC : ABD :: BCD	BCE	3176
ABC : ABD :: IJJKKK	IJKKKL	6454
ABC : ABD :: XYZ	XYZS	5267
ABC : ABD :: 122333	123334	5440
ABC : ABD :: RSSTTT	RSTTTU	4384
ABC : ABD :: IJJKKK	IJKKKL	5110
ABC : ABD :: AABABC	AACABD	4301
ABC : ABD :: MRRJJJ	MRSJJK	7475
ABC : ABD :: 147	148	3806

Result:

- Date: Wednesday February 01 2017 23:03:22 UTC
- Age: 22
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	21976
ABC : ABD :: BCA	BCB	13099
ABC : ABD :: AABABC	AABABD	16463
ABC : ABD :: IJKLM	IJKLN	9873
ABC : ABD :: 123	124	4839
ABC : ABD :: KJI	KJJ	13159
ABC : ABD :: 135	136	4615
ABC : ABD :: BCD	BCE	5692
ABC : ABD :: IJJKKK	IJJKKL	6547
ABC : ABD :: XYZ	XYA	6725
ABC : ABD :: 122333	122334	8009
ABC : ABD :: RSSTTT	RSSTTTUUUU	18833
ABC : ABD :: IJJKKK	IJJKKKLLLL	8566
ABC : ABD :: AABABC	AABABCB	24548
ABC : ABD :: MRRJJJ	MRRJJJSSSS	18367
ABC : ABD :: 147	14711	9404

Result:

- Date: Wednesday February 01 2017 23:03:28 UTC
- Age: 21
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	12880
ABC : ABD :: BCA	BDA	13036
ABC : ABD :: AABABC	AABABD	19920
ABC : ABD :: IJKLM	IJLLN	14154
ABC : ABD :: 123	124	9298
ABC : ABD :: KJI	KJI	23457
ABC : ABD :: 135	136	7063
ABC : ABD :: BCD	CDE	15202
ABC : ABD :: IJJKKK	IJJLLL	17334
ABC : ABD :: XYZ	XYZ	14045
ABC : ABD :: 122333	1224444	15762
ABC : ABD :: RSSTTT	RSSUUU	26300
ABC : ABD :: IJJKKK	IJJLLL	12772
ABC : ABD :: AABABC	AABABD	35890
ABC : ABD :: MRRJJJ	MRRKKK	17308
ABC : ABD :: 147	148	9386

Result:

- Date: Wednesday February 01 2017 23:14:20 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	10521
ABC : ABD :: BCA	BCB	18406
ABC : ABD :: AABABC	AABABD	14970
ABC : ABD :: IJKLM	IJKLN	9010
ABC : ABD :: 123	124	5918
ABC : ABD :: KJI	KJJ	9055
ABC : ABD :: 135	136	5938
ABC : ABD :: BCD	BCE	9546
ABC : ABD :: IJJKKK	IJJKKL	9853
ABC : ABD :: XYZ	XYA	8512
ABC : ABD :: 122333	122334	7168
ABC : ABD :: RSSTTT	RSSTTU	9061
ABC : ABD :: IJJKKK	IJJKKL	11726
ABC : ABD :: AABABC	AABABD	8115
ABC : ABD :: MRRJJJ	MRRJJK	9615
ABC : ABD :: 147	148	4332

Result:

- Date: Wednesday February 01 2017 23:16:39 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	26442
ABC : ABD :: BCA	CDA	33388
ABC : ABD :: AABABC	AABABD	20633
ABC : ABD :: IJKLM	IJKLN	15300
ABC : ABD :: 123	124	6483
ABC : ABD :: KJI	LJI	8850
ABC : ABD :: 135	136	13938
ABC : ABD :: BCD	CDE	10517
ABC : ABD :: IJJKKK	IJJKKL	15941
ABC : ABD :: XYZ	YZA	27439
ABC : ABD :: 122333	122334	8951
ABC : ABD :: RSSTTT	RSSTTU	7169
ABC : ABD :: IJJKKK	IJJKKL	11550
ABC : ABD :: AABABC	AACABD	16224
ABC : ABD :: MRRJJJ	MRSJJK	11869
ABC : ABD :: 147	148	7136

Result:

- Date: Thursday February 02 2017 06:29:06 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	17912
ABC : ABD :: BCA	BCB	37716
ABC : ABD :: AABABC	AABABD	29088
ABC : ABD :: IJKLM	IJKLN	21746
ABC : ABD :: 123	124	9784
ABC : ABD :: KJI	KJJ	12403
ABC : ABD :: 135	136	16567
ABC : ABD :: BCD	BCE	8512
ABC : ABD :: IJJKKK	IJKKL	7833
ABC : ABD :: XYZ	XZA	37571
ABC : ABD :: 122333	122334	8817
ABC : ABD :: RSSTTT	RSSTTU	20544
ABC : ABD :: IJJKKK	IJJKKL	23801
ABC : ABD :: AABABC	AABABD	15902
ABC : ABD :: MRRJJJ	MRRJJK	16661
ABC : ABD :: 147	148	16935

Result:

- Date: Thursday February 02 2017 08:31:52 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	20443
ABC : ABD :: BCA	CBA	38838
ABC : ABD :: AABABC	AABABD	33237
ABC : ABD :: IJKLM	IJKLN	10389
ABC : ABD :: 123	124	4663
ABC : ABD :: KJI	LJI	9181
ABC : ABD :: 135	136	12771
ABC : ABD :: BCD	BCE	7855
ABC : ABD :: IJJKKK	IJJLLL	12301
ABC : ABD :: XYZ	XYA	7690
ABC : ABD :: 122333	122444	6943
ABC : ABD :: RSSTTT	RSSUUU	7426
ABC : ABD :: IJJKKK	IJJLLL	14484
ABC : ABD :: AABABC	AABABD	8799
ABC : ABD :: MRRJJJ	MRRKKK	10505
ABC : ABD :: 147	148	4988

Result:

- Date: Thursday February 02 2017 09:45:39 UTC
- Age: 23
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	10730
ABC : ABD :: BCA	BCB	16478
ABC : ABD :: AABABC	AABABD	9841
ABC : ABD :: IJKLM	IJLMN	9596
ABC : ABD :: 123	124	7396
ABC : ABD :: KJI	KJJ	15114
ABC : ABD :: 135	136	6318
ABC : ABD :: BCD	BCE	4348
ABC : ABD :: IJJKKK	IJKLLL	11670
ABC : ABD :: XYZ	XZZ	10415
ABC : ABD :: 122333	122444	5457
ABC : ABD :: RSSTTT	RSSUUU	6045
ABC : ABD :: IJJKKK	IJKLLL	6019
ABC : ABD :: AABABC	AABABD	5354
ABC : ABD :: MRRJJJ	MRRKKK	8740
ABC : ABD :: 147	148	5389

Result:

- Date: Thursday February 02 2017 10:43:21 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	13474
ABC : ABD :: BCA	BDA	41879
ABC : ABD :: AABABC	AABABD	17399
ABC : ABD :: IJKLM	IJKLN	50800
ABC : ABD :: 123	124	3879
ABC : ABD :: KJI	KJH	18552
ABC : ABD :: 135	136	8516
ABC : ABD :: BCD	BCE	10040
ABC : ABD :: IJJKKK	IJJKKKK	15297
ABC : ABD :: XYZ	XYA	14582
ABC : ABD :: 122333	1224444	9279
ABC : ABD :: RSSTTT	RSSUUUU	10495
ABC : ABD :: IJJKKK	IJJLLLL	15285
ABC : ABD :: AABABC	AABABD	18618
ABC : ABD :: MRRJJJ	MRRJJD	25586
ABC : ABD :: 147	148	13126

Result:

- Date: Thursday February 02 2017 10:48:58 UTC
- Age: 24
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	6723
ABC : ABD :: BCA	BCZ	6450
ABC : ABD :: AABABC	AACABD	11294
ABC : ABD :: IJKLM	IJLLM	6089
ABC : ABD :: 123	124	3004
ABC : ABD :: KJI	KJH	10429
ABC : ABD :: 135	136	4462
ABC : ABD :: BCD	BCE	2744
ABC : ABD :: IJJKKK	IJKKKL	5887
ABC : ABD :: XYZ	XYA	3175
ABC : ABD :: 122333	123334	3831
ABC : ABD :: RSSTTT	RSTTTU	4125
ABC : ABD :: IJJKKK	IJKKKL	5690
ABC : ABD :: AABABC	AACABD	4832
ABC : ABD :: MRRJJJ	MRSJJK	7870
ABC : ABD :: 147	148	2269

Result:

- Date: Thursday February 02 2017 12:28:02 UTC
- Age: 29
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	25252
ABC : ABD :: BCA	BCB	16043
ABC : ABD :: AABABC	AABABD	16088
ABC : ABD :: IJKLM	IJKLN	9549
ABC : ABD :: 123	124	5091
ABC : ABD :: KJI	KJK	16751
ABC : ABD :: 135	136	6460
ABC : ABD :: BCD	BCE	6105
ABC : ABD :: IJJKKK	IJJKKL	8258
ABC : ABD :: XYZ	XYA	7036
ABC : ABD :: 122333	122334	6009
ABC : ABD :: RSSTTT	RSSTTU	10282
ABC : ABD :: IJJKKK	IJJKKL	7595
ABC : ABD :: AABABC	AABAABD	7260
ABC : ABD :: MRRJJJ	MRRJJK	7428
ABC : ABD :: 147	148	7607

Result:

- Date: Thursday February 02 2017 13:03:15 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	14381
ABC : ABD :: BCA	BDA	24101
ABC : ABD :: AABABC	AABABD	13355
ABC : ABD :: IJKLM	IJLLM	12095
ABC : ABD :: 123	124	6273
ABC : ABD :: KJI	LJI	44882
ABC : ABD :: 135	136	6452
ABC : ABD :: BCD	BCE	13310
ABC : ABD :: IJJKKK	IJJLLL	64941
ABC : ABD :: XYZ	XYA	6954
ABC : ABD :: 122333	122444	5661
ABC : ABD :: RSSTTT	RSSUUU	11779
ABC : ABD :: IJJKKK	IJJLLL	10877
ABC : ABD :: AABABC	AABABD	101031
ABC : ABD :: MRRJJJ	MRRKKK	25249
ABC : ABD :: 147	148	5120

Result:

- Date: Thursday February 02 2017 13:36:24 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	12002
ABC : ABD :: BCA	BCB	17113
ABC : ABD :: AABABC	AABABD	14849
ABC : ABD :: IJKLM	IJKLN	7315
ABC : ABD :: 123	124	3443
ABC : ABD :: KJI	KJJ	9851
ABC : ABD :: 135	136	7324
ABC : ABD :: BCD	BCE	7187
ABC : ABD :: IJJKKK	IJJKKL	5537
ABC : ABD :: XYZ	XYA	7083
ABC : ABD :: 122333	122334	5766
ABC : ABD :: RSSTTT	RSSTTU	6074
ABC : ABD :: IJJKKK	IJJKKL	4706
ABC : ABD :: AABABC	AABABD	8462
ABC : ABD :: MRRJJJ	MRRJJK	5267
ABC : ABD :: 147	148	3405

Result:

- Date: Thursday February 02 2017 14:59:03 UTC
- Age: 23
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	10208
ABC : ABD :: BCA	BCB	13918
ABC : ABD :: AABABC	AABABD	11127
ABC : ABD :: IJKLM	IJKLN	9909
ABC : ABD :: 123	124	4892
ABC : ABD :: KJI	KJJ	3793
ABC : ABD :: 135	136	6266
ABC : ABD :: BCD	BCE	4291
ABC : ABD :: IJJKKK	IJJKKL	13148
ABC : ABD :: XYZ	XYA	4885
ABC : ABD :: 122333	122334	4036
ABC : ABD :: RSSTTT	RSSTTU	7046
ABC : ABD :: IJJKKK	IJJKKL	5071
ABC : ABD :: AABABC	AABABD	4243
ABC : ABD :: MRRJJJ	MRRJJK	5427
ABC : ABD :: 147	148	2717

Result:

- Date: Thursday February 02 2017 15:11:27 UTC
- Age: 20
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	17338
ABC : ABD :: BCA	BCB	22786
ABC : ABD :: AABABC	AABABD	38754
ABC : ABD :: IJKLM	IJKLN	23407
ABC : ABD :: 123	124	12117
ABC : ABD :: KJI	KJJ	12166
ABC : ABD :: 135	136	7512
ABC : ABD :: BCD	BCE	8021
ABC : ABD :: IJJKKK	IJJKKL	10607
ABC : ABD :: XYZ	XYA	7517
ABC : ABD :: 122333	122334	9450
ABC : ABD :: RSSTTT	RSSTTU	9192
ABC : ABD :: IJJKKK	IJKKKL	11751
ABC : ABD :: AABABC	AACABD	7984
ABC : ABD :: MRRJJJ	MRSJJK	10426
ABC : ABD :: 147	148	8398

Result:

- Date: Thursday February 02 2017 16:32:57 UTC
- Age: 19
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	7968
ABC : ABD :: BCA	DCA	8331
ABC : ABD :: AABABC	AABABD	7990
ABC : ABD :: IJKLM	IJKLN	7843
ABC : ABD :: 123	124	3663
ABC : ABD :: KJI	LJI	14490
ABC : ABD :: 135	135	6819
ABC : ABD :: BCD	DCB	17939
ABC : ABD :: IJJKKK	IJJLLL	76006
ABC : ABD :: XYZ	XYA	4638
ABC : ABD :: 122333	1224444	9098
ABC : ABD :: RSSTTT	RSSUUU	39569
ABC : ABD :: IJJKKK	IJJLLL	7199
ABC : ABD :: AABABC	AABABD	14657
ABC : ABD :: MRRJJJ	MRJJJ	7822
ABC : ABD :: 147	147	4053

Result:

- Date: Thursday February 02 2017 16:37:05 UTC
- Age: 19
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	20667
ABC : ABD :: BCA	BCB	15044
ABC : ABD :: AABABC	AACABD	14158
ABC : ABD :: IJKLM	IJLLN	18316
ABC : ABD :: 123	124	7408
ABC : ABD :: KJI	KJJ	8280
ABC : ABD :: 135	136	6779
ABC : ABD :: BCD	BCE	8290
ABC : ABD :: IJJKKK	IJKLL	18236
ABC : ABD :: XYZ	XYA	7763
ABC : ABD :: 122333	123444	25913
ABC : ABD :: RSSTTT	RSTUUU	12780
ABC : ABD :: IJJKKK	IJKLLL	5795
ABC : ABD :: AABABC	AACBCD	31453
ABC : ABD :: MRRJJJ	MRSKKK	7322
ABC : ABD :: 147	148	5032

Result:

- Date: Thursday February 02 2017 16:45:13 UTC
- Age: 22
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	20113
ABC : ABD :: BCA	BDA	33611
ABC : ABD :: AABABC	AABABD	29971
ABC : ABD :: IJKLM	IJLMN	40741
ABC : ABD :: 123	124	7205
ABC : ABD :: KJI	LJI	37716
ABC : ABD :: 135	137	24918
ABC : ABD :: BCD	BCE	14230
ABC : ABD :: IJJKKK	IJJLLL	19145
ABC : ABD :: XYZ	XYA	7189
ABC : ABD :: 122333	122444	9732
ABC : ABD :: RSSTTT	RSSUUU	17441
ABC : ABD :: IJJKKK	IJJLLL	11258
ABC : ABD :: AABABC	AABABD	28230
ABC : ABD :: MRRJJJ	MRRKKK	28746
ABC : ABD :: 147	1410	28884

Result:

- Date: Thursday February 02 2017 18:21:01 UTC
- Age: 19
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IKL	9096
ABC : ABD :: BCA	BCB	25344
ABC : ABD :: AABABC	AABABD	18809
ABC : ABD :: IJKLM	IJKLN	13477
ABC : ABD :: 123	124	5152
ABC : ABD :: KJI	LJI	10098
ABC : ABD :: 135	136	6261
ABC : ABD :: BCD	BCE	7018
ABC : ABD :: IJJKKK	IJJLLL	12502
ABC : ABD :: XYZ	XYA	7483
ABC : ABD :: 122333	123444	5836
ABC : ABD :: RSSTTT	RSSUUU	5819
ABC : ABD :: IJJKKK	IJJLLL	5870
ABC : ABD :: AABABC	AABABD	9169
ABC : ABD :: MRRJJJ	MRRKKK	13798
ABC : ABD :: 147	148	4509

Result:

- Date: Thursday February 02 2017 23:32:40 UTC
- Age: 21
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	41117
ABC : ABD :: BCA	BDA	75011
ABC : ABD :: AABABC	AABABD	23195
ABC : ABD :: IJKLM	IJKLN	121816
ABC : ABD :: 123	124	9761
ABC : ABD :: KJI	LJI	65275
ABC : ABD :: 135	136	18000
ABC : ABD :: BCD	BCE	14578
ABC : ABD :: IJJKKK	IJJLLL	39866
ABC : ABD :: XYZ	XYA	17660
ABC : ABD :: 122333	122444	19024
ABC : ABD :: RSSTTT	RSUUU	20013
ABC : ABD :: IJJKKK	IJJLLL	29972
ABC : ABD :: AABABC	AABABD	14789
ABC : ABD :: MRRJJJ	MRRKKK	113744
ABC : ABD :: 147	148	23078

Result:

- Date: Friday February 03 2017 00:21:31 UTC
- Age: 19
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	12982
ABC : ABD :: BCA	BCB	10519
ABC : ABD :: AABABC	AACABD	9057
ABC : ABD :: IJKLM	IJLLM	13059
ABC : ABD :: 123	124	5714
ABC : ABD :: KJI	KJJ	5907
ABC : ABD :: 135	136	4026
ABC : ABD :: BCD	BCE	4770
ABC : ABD :: IJJKKK	IJKKKL	12981
ABC : ABD :: XYZ	XYA	6917
ABC : ABD :: 122333	123334	10684
ABC : ABD :: RSSTTT	RSTTTU	6940
ABC : ABD :: IJJKKK	IJKKKL	11403
ABC : ABD :: AABABC	AACABD	9773
ABC : ABD :: MRRJJJ	MRSJJK	9407
ABC : ABD :: 147	148	5043

Result:

- Date: Friday February 03 2017 14:43:34 UTC
- Age: 19
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	11602
ABC : ABD :: BCA	DBA	10218
ABC : ABD :: AABABC	AABABD	12242
ABC : ABD :: IJKLM	IJKLMO	13223
ABC : ABD :: 123	124	7932
ABC : ABD :: KJI	LJI	15332
ABC : ABD :: 135	137	15492
ABC : ABD :: BCD	BCE	12572
ABC : ABD :: IJJKKK	IJJLLL	17772
ABC : ABD :: XYZ	XYA	7076
ABC : ABD :: 122333	122444	6584
ABC : ABD :: RSSTTT	RSSUUU	8342
ABC : ABD :: IJJKKK	IJJLLL	7400
ABC : ABD :: AABABC	AABABD	8473
ABC : ABD :: MRRJJJ	MRRKKK	17010
ABC : ABD :: 147	150	13468

Result:

- Date: Friday February 03 2017 16:29:34 UTC
- Age: 19
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	14267
ABC : ABD :: BCA	BCB	18297
ABC : ABD :: AABABC	AABABD	12035
ABC : ABD :: IJKLM	IJKLN	7964
ABC : ABD :: 123	124	5364
ABC : ABD :: KJI	KJK	5670
ABC : ABD :: 135	136	10404
ABC : ABD :: BCD	BCE	8046
ABC : ABD :: IJJKKK	IJJKKL	6550
ABC : ABD :: XYZ	XYA	5161
ABC : ABD :: 122333	122334	6590
ABC : ABD :: RSSTTT	RSSTTU	7033
ABC : ABD :: IJJKKK	IJJKKL	5764
ABC : ABD :: AABABC	AABABD	6582
ABC : ABD :: MRRJJJ	MRRJJK	11245
ABC : ABD :: 147	148	4564

Result:

- Date: Sunday February 05 2017 12:33:27 UTC
- Age: 14
- Gender: female

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	17704
ABC : ABD :: BCA	BDA	12851
ABC : ABD :: AABABC	AABABD	56625
ABC : ABD :: IJKLM	IJKLN	33381
ABC : ABD :: 123	124	4694
ABC : ABD :: KJI	45	23324
ABC : ABD :: 135	136	4888
ABC : ABD :: BCD	BCE	8335
ABC : ABD :: IJJKKK	IJJLLL	19736
ABC : ABD :: XYZ	XYA	6352
ABC : ABD :: 122333	122444	6805
ABC : ABD :: RSSTTT	RSSUUU	7097
ABC : ABD :: IJJKKK	IJJLLL	12932
ABC : ABD :: AABABC	AABABD	42851
ABC : ABD :: MRRJJJ	MRRKKK	17162
ABC : ABD :: 147	148	5506

Result:

- Date: Wednesday March 01 2017 17:24:37 UTC
- Age: 19
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	12960
ABC : ABD :: BCA	BCB	16342
ABC : ABD :: AABABC	AACABD	9508
ABC : ABD :: IJKLM	IJLLM	24006
ABC : ABD :: 123	124	7492
ABC : ABD :: KJI	KJJ	4500
ABC : ABD :: 135	136	3966
ABC : ABD :: BCD	BCE	3661
ABC : ABD :: IJJKKK	IJKKKL	9258
ABC : ABD :: XYZ	XYA	6598
ABC : ABD :: 122333	123334	6150
ABC : ABD :: RSSTTT	RSTTTU	9623
ABC : ABD :: IJJKKK	IJKKKL	8721
ABC : ABD :: AABABC	AACABD	7677
ABC : ABD :: MRRJJJ	MRSJJK	9172
ABC : ABD :: 147	148	6378

Result:

- Date: Saturday July 21 2018 10:03:40 UTC
- Age: 36
- Gender: male

Problem	Solution	Time (ms)
ABC : ABD :: IJK	IJL	21702
ABC : ABD :: BCA	BCB	21259
ABC : ABD :: AABABC	AABABD	31015
ABC : ABD :: IJKLM	IJKLN	20648
ABC : ABD :: 123	124	8382
ABC : ABD :: KJI	KJJ	13585
ABC : ABD :: 135	136	7493
ABC : ABD :: BCD	BCE	9792
ABC : ABD :: IJJKKK	IJJKKL	22036
ABC : ABD :: XYZ	XYA	14340
ABC : ABD :: 122333	122334	7466
ABC : ABD :: RSSTTT	RSSTTU	17111
ABC : ABD :: IJJKKK	IJJKKL	11289
ABC : ABD :: AABABC	AABABD	22691
ABC : ABD :: MRRJJJ	MRRJJK	19535
ABC : ABD :: 147	148	4561

A.3.2 Ages

We show the age distribution in Table A.1. The large number of participants aged 15-30 is a bias inherent to the diffusion method for the experiment (through social networks).

14	*1*	**18**	*3*	**19**	*10*	**20**	*15*	**21**	*13*	**22**	*10*
23	*16*	**24**	*7*	**25**	*12*	**26**	*3*	**27**	*3*	**28**	*1*
29	*1*	**31**	*2*	**36**	*1*	**54**	*1*	**67**	*1*	**72**	*1*

TABLE A.1: Age distribution. In a row, the ages (in bold font) are followed by the corresponding number of participants (in italic).

A.3.3 Results by Question

We now give the full statistics of the results grouped by question. Each one of the following tables corresponds to an analogy equation (displayed in first row). The proposed answers for the equation are given in the table, followed by the number of times it has been suggested.

ABC : ABD :: IJK : x					
IJL	95	IKL	2	IJF	2
IJD	1	ADE	1		

ABC : ABD :: BCA : x					
BCB	51	BDA	38	DBA	3
BCZ	2	BCD	1	BCZS	1
BAC	1	DAB	1	CDA	1
CBA	1	DCA	1		

ABC : ABD :: AABABC : x					
AABABD	76	AACABD	15	AABACD	2
AADABC	1	AABBD	1	BBCBCD	1
AABB	1	ABE	1	AABBAD	1
AABBCD		1	AABABCABD		1

ABC : ABD :: IJKLM : x					
IJKLN	62	IJLLM	15	IJLMN	6
IJKMN	3	IJKLM	3	IJLLN	2
IJKLD	1	IJJKLM	1	IJLKM	1
IJKLMN	1	IJKLO	1	IJLP	1
VRJTN	1	OPQRS	1	AJLLM	1
IJKLMO	1				

ABC : ABD :: 123 : x					
124	97	123	3	456	1

ABC : ABD :: KJI : x					
KJJ	40	LJI	33	KJH	8
KJI	6	KJK	5	KJL	2
KJD	1	KJG	1	JJI	1
MJI	1	LMN	1	LKI	1
45	1				

ABC : ABD :: 135 : x					
136	71	137	20	135	5
145	2	13D	1	146	1
1416	1				

ABC : ABD :: BCD : x					
BCE	84	BCD	3	BDD	3
BDE	3	CDE	2	BDC	1
BCC	1	BED	1	BDH	1
EFG	1	DCB	1		

ABC : ABD :: IJJKKK : x					
IJJLLL	39	IJJKKL	26	IJKKKL	14
IJJKKK	6	IJJKL	2	IJJKLL	2
IJJKKKK	2	IJDDDD	1	IJJKKKMMMM	1
IJJKKM	1	IIJJKKL	1	IJJKKF	1
IJJLLLL	1	IJKLL	1	IJKKL	1
IJKLLL		1	LLLLMMMMMNNNNNN		1

ABC : ABD :: XYZ : x					
XYA	85	XYZ	5	XYZS	2
XYA1	1	XY0	1	XY.	1
XY	1	XYä	1	ABC	1
YZA	1	XZA	1	XZZ	1

ABC : ABD :: 122333 : x					
122444	35	122334	31	123334	14
1224444	8	122333	4	122344	2
1122444	2	123444	2	122433	1
1223333		1	444455555666666		1

ABC : ABD :: RSSTTT : x					
RSSUUU	41	RSSTTU	30	RSTTTU	13
RSSTTT	5	RSSUUUU	3	RSDTTT	1
RSSTTTU	1	RSSTWW	1	RSTUU	1
RSSTTTT		1	UUUUVVVVVWWWWWW		1
RSSTTTUUUU		1	RSUUU		1

ABC : ABD :: IJJKKK : x					
IJJLLL	40	IJJKKL	27	IJKKKL	15
IJJKKK	5	IJJLLLL	3	IJKLLL	2
IJJDDD	1	IJLLL	1	IJJKLL	1
IJKKL		1	IJKLL		1
LLLLMMMMMNNNNNN		1	OJJKKL		1
IJJKKKK		1	IJJKKKLLLL		1

ABC : ABD :: AABABC : x					
AABABD	69	AACABD	16	AABACD	5
AABABC	2	AABABB	1	AABDBD	1
AABABCABD	1	AABABE	1	ABBA	1
AAABABD	1	AABABCB	1	AABAABD	1
AACBCD	1				

ABC : ABD :: MRRJJJ : x					
MRRJJK	26	MRRKKK	23	MRSJJK	14
MRRJJJ	6	MRRLLL	5	MSSJJJ	4
MRDJJD	1	MRRJJ	1	MRREEE	1
MRRLLLL	1	MROJJK	1	MRRIII	1
MRRSSSS	1	MRRJKK	1	:(	1
MRRJJH	1	MRJII	1	?	1
MRJJ	1	MRRJJW	1	MRRJJJJ	1
MRRPPP	1	I BELIEVE I CAN FLY-YYYY	1	MRRKKKK	1
MDR	1	MRRJJJSSSS	1	MRRJJD	1
MRJJJ	1	MRSKKK	1		

ABC : ABD :: 147 : x					
148	72	1410	10	147	6
141	2	14711	2	140	1
158	1	1 4 10	1	14A	1
?	1	149	1	1411	1
369	1	150	1		

Appendix B

Résumé en Français

Dans ce chapitre, nous proposons un résumé de la thèse en langue française. Le plan du résumé suit le plan général de la thèse.

In this chapter, we propose a summary of the thesis in French. The outline of the summary is the same as the general outline of the thesis.

B.1 Un problème fondamental: Le Raisonnement par Analogie

Dans cette section, nous présentons un premier problème d'apprentissage, le raisonnement par analogie. Après une brève présentation de cette forme de raisonnement, nous analysons l'utilisation d'un principe de description minimale dans le cadre particulier des analogies sur chaînes de caractères, puis dans un cadre plus général, avant de présenter une réflexion sur l'analogie dans des espaces géométriques complexes.

B.1.1 Introduction au Raisonnement par Analogie

Le terme *raisonnement par analogie* désigne toute forme de raisonnement établissant des parallèles entre deux domaines distincts et *a priori* décorrélés. L'idée fondamentale sous-jacente à l'analogie est que, si deux situations sont similaires sous certains aspects, elles doivent être similaires sous d'autres aspects. Partant de cette définition informelle, il est clair que les questions levées par l'analogie s'articulent en particulier autour d'une définition formelle de la similarité, et des conditions de validité de cette conjecture.

La capacité de produire et de comprendre des analogies est une aptitude fondamentale partagée par les êtres humains, à tel point qu'elle est même utilisée comme une mesure de l'intelligence humaine (avec en particulier les tests de QI qui s'appuient majoritairement sur des analogies). On peut trouver des formes d'analogie dans diverses activités quotidiennes, telles que les métaphores, l'humour, ou même la méthode scientifique.

Partant de ces observations, les sciences cognitives ont proposé différentes approches afin de décrire et d'imiter cette aptitude. Parmi elles, on peut citer en particulier la *théorie du liage structurel* (*Structure Mapping Theory*, ou SMT) proposée par (Gentner, 1983) et les modèles connectionistes ACME (Holyoak and Thagard, 1989) ou LISA (Hummel and Holyoak, 1997).

D'un point de vue moins cognitif, une théorie très importante est celle de l'analogie proportionnelle. Dans ce formalisme, une analogie est une relation 4-aire (correspondant aux quatre termes de l'analogie "A est à B ce que C est à D") satisfaisant les trois axiomes suivants (Miclet, Bayoudh, and Delhay, 2008):

- Déterminisme: $A(a,b,a,b)$ est toujours vrai.
- Symétrie: $A(a,b,c,d) \Rightarrow A(c,d,a,b)$
- Permutation centrale: $A(a,b,c,d) \Rightarrow A(a,c,b,d)$

Ces trois axiomes ont été utilisés pour décrire des analogies entre objets dans différents espaces discrets (analogies entre vecteurs booléens (Prade and Richard, 2013), entre ensemble finis (Miclet, Bayoudh, and Delhay, 2008), entre caractères sur un alphabet (Lepage, 1998)) mais aussi sur des espaces vectoriels continus. Une application majeure de cette idée est la règle du parallèlogramme ($d = c + b - a$) utilisée récemment dans le modèle word2vec (Mikolov, Chen, Corrado, and Dean, 2013).

Dans le cadre de cette thèse, nous avons défini l'analogie comme toute relation 4-aire $A : \mathcal{X}_S \times \mathcal{Y}_S \times \mathcal{X}_T \times \mathcal{Y}_T$. Nous appelons domaine source l'ensemble $\mathcal{X}_S \times \mathcal{Y}_S$ et domaine cible l'ensemble $\mathcal{X}_T \times \mathcal{Y}_T$. Par extension, nous qualifions de *source* la donnée de $(a,b) \in \mathcal{X}_S \times \mathcal{Y}_S$ et de *cible* la donnée de $(c,d) \in \mathcal{X}_T \times \mathcal{Y}_T$.

Nous proposons dans cette thèse de discuter la notion de "bonne" analogie, qui est une idée mal posée. L'analogie proportionnelle est une première tentative de définition, mais nous montrerons dans la Section B.1.4 qu'elle ne s'étend pas directement à des analogies dans des espaces courbes (variétés riemanniennes).

B.1.2 Analogies à longueur de description minimale sur les chaînes de caractères

Afin d'introduire les idées principales de cette thèse, nous proposons tout d'abord de considére un exemple simple d'analogies, le micro-monde de Hofstadter (Hofstadter and Mitchell, 1995). Ce micro-monde consiste en des analogies sur l'alphabet latin considéré comme une entité ordonnée. Ainsi, dans cette optique, la chaîne de caractères `abc` est comprise comme la succession de trois lettres de l'alphabet et non comme un groupe de trois lettres non corrélées. Un exemple d'analogie dans le micro-monde de Hofstadter serait "**ABC** est à **ABD** ce que **IJK** est à **IJL**" (aussi noté **ABC** : **ABD** :: **IJK** : **IJL**).

Nous avons proposé d'étendre l'usage de ce micro-monde à trois cas non traités originellement. Tout d'abord, nous considérons que le domaine est alphanumérique et non uniquement alphabétique. Dans le cadre strict du micro-monde de Hofstadter, la numération n'est pas prise en compte, ce qui rend impossible la résolution d'analogies du type **ABC** : **ABD** :: **IJJKKK** : x où x est l'inconnue. Ensuite, nous autorisons un changement de domaine, C'est-à-dire un cas où les domaines $\mathcal{X}_S \times \mathcal{Y}_S$ et $\mathcal{X}_T \times \mathcal{Y}_T$ sont distincts. Cela permet de considérer par exemple des analogies du type **ABC** : **ABD** :: **123** : **124**. Il devient entre autres possible de considérer des analogies sur n'importe quel domaine, tant que celui-ci est ordonné. Enfin, nous nous autorisons à ne pas utiliser l'ordre de l'alphabet pour résoudre les analogies. Cela permet en particulier d'étendre la méthode aux applications d'analogie sur des mots (ce qui rend compte par exemple du phénomène de déclinaison: **ROSA** : **ROSAM** :: **VITA** : **VITAM**).

Afin de résoudre ce problème, nous avons développé un petit langage génératif capable de produire des chaînes de caractères. Par exemple, dans ce langage simple, l'instruction `alphabet, sequence, 3` renvoie la séquence des trois premières lettres de l'alphabet. La caractéristique principale de ce langage est le recours à une opération de mémorisation qui permet d'enregistrer des chaînes de caractères ainsi que des opérateurs. La mémorisation d'opérateurs est essentielle dans ce que nous

considérons être une bonne analogie, car transférabilité signifie qu'il existe une abstraction de la transformation liant question et réponse et s'appliquant à la fois au domaine source et au domaine cible. Pour une analogie A:B::C:D donnée, plusieurs programmes permettent de générer les quatre termes. Notre hypothèse est que le programme de référence est celui dont l'expression binaire est la plus courte. Ainsi, dans le cadre de notre langage, nous pouvons associer à chaque analogie une quantité, que nous appelons sa *longueur de description*, ou *complexité*. Dans ce contexte, inspiré par le principe du rasoir d'Occam, mais aussi par de récentes études cognitives (Chater, 1999), nous avons émis l'hypothèse que la meilleure analogie correspond à celle de longueur de description minimale.

Pour valider cette hypothèse, il conviendrait de pouvoir connaître, pour chaque analogie, sa longueur de description, et donc *a fortiori* le programme génératif le plus court. En l'état, le langage développé ne permet pas de procéder à une recherche de programme minimal, et donc de valider complètement l'hypothèse selon laquelle les meilleures analogies sont celles de complexité minimale. Néanmoins, nous avons testé manuellement un certain nombre de programmes et de solutions, et nous avons constaté que la solution de moindre complexité coïncide en général avec la réponse majoritaire proposée par des sujets humains. Les résultats détaillés de ces expériences sont proposés en Annexe A.

B.1.3 Analogies de complexité minimale

L'idée d'une analogie de complexité minimale, introduite plus haut avec l'exemple des analogies de Hofstadter, peut être généralisée à un cadre plus large. Nous constatons que l'opération principale de notre langage est la mise en mémoire. Cette mise en mémoire permet une factorisation des éléments communs entre la question et la réponse (càd, dans l'analogie A:B::C:D, entre A et B, et entre C et D), mais aussi entre la source et la cible (càd entre A:B et C:D). En poursuivant cette idée, nous avons proposé d'isoler plusieurs termes, correspondant aux différents éléments à décrire.

Dans ce cadre, nous avons proposé l'utilisation d'un outil plus général : la complexité de Kolmogorov. La complexité d'une chaîne de caractères x conditionnellement à une chaîne y est décrite comme la longueur du plus court programme, sur une machine de Turing universlle fixée, tel que, exécuté à partir de l'entrée y, ce programme renvoie la chaîne x. Cette définition pose un certain nombre de problèmes.

Tout d'abord, elle dépend de la machine de Turing universlle choisie. Autrement dit, changer de machine de référence modifie la valeur de la complexité. Dans le cas de notre étude, nous n'avons pas conçu ce problème comme limitant. Nous acceptons que le choix de la machine conditionne les résultats de l'inférence. Cela peut être interprété comme une notion de biais, inhérente à toute forme de raisonnement non déductif. L'idée n'est donc pas d'évaluer des analogies de façon générale, mais dans un cadre inductif précis (par exemple un cadre cognitif raisonnable). Ensuite, il peut être montré que la dépendance à la machine est limitée. Le théorème d'invariance expose que les complexités relatives à deux machines différentes ne peuvent varier que d'une constante, dépendant des deux machines. Ainsi, une chaîne qui serait simple pour une machine ne peut pas être arbitrairement complexe pour une autre machine.

Un autre problème majeur de la complexité est sa non-calculabilité. Intuitivement, cette idée est liée à l'indécidabilité du problème de l'arrêt : étant donné un programme, il n'est *a priori* pas possible de savoir si son exécution s'arrête en temps fini. Puisque la complexité n'est pas calculable, il semble impossible de l'utiliser en

pratique. À l'instar de l'essentiel des applications de la complexité à l'apprentissage artificiel (en particulier les principes MDL et MML), nous nous intéressons à un sous-ensemble de programmes de la machine de référence, sur lequel nous avons une garantie que la complexité est calculable. L'objet que nous qualifions de *complexité* n'en est pas une *stricto sensu*: du fait de cette restriction sur l'ensemble des programmes, elle ne vérifie plus le théorème d'invariance. Pour les mêmes raisons que précédemment, nous estimons que cette restriction n'est pas limitante et correspond à une hypothèse de biais.

Enfin, il existe deux variantes de la notion de complexité. La notion précédente, notée $C(.)$, considère l'ensemble des programmes s'exécutant sur une machine de Turing universelle. Cette définition a un certain nombre de défauts pratiques, du fait que les programmes ne sont pas délimités. Une notion alternative a été développée. Il s'agit de la complexité *préfixe*, notée $K(.)$. Elle varie de $C(.)$ par le fait qu'elle ne s'applique qu'à une machine préfixe. Intuitivement, une machine préfixe n'admet que des programmes dont chaque couple (p, q) vérifie une propriété additionnelle: p ne peut pas être un préfixe de q. Avec cette seconde définition, des propriétés importantes peuvent être observées, en particulier liant complexité et complexité conditionnelle. Pour tout couple (x, y) de chaînes binaires, il peut être montré que $K(x) \leq K(y) + K(x|y)$ à une constante additive près. Nous utilisons cette règle, appelée *formule des complexités composées* (en anglais *chain rule*), pour définir un nouvel objet, le *modèle graphique génératif*.

Un modèle graphique génératif correspond à une restriction de l'ensemble des programmes considérés, décrite par un graphe orienté acycle. Considérant que chaque sommet du graphe correspond à une variable (observée ou latente), un modèle graphique génératif correspond à l'ensemble des machines décomposées en sous-machines décrivant chaque variable à partir de ses parents dans le graphe. Ce modèle est fortement inspiré des modèles graphiques probabilistes.

Nous proposons un modèle graphique simple, inspiré par les travaux de (Cornuéjols and Ales-Bianchetti, 1998) et illustré en Figure B.1. L'idée est de considérer des variables latentes, appelées *modèles*, dont la fonction est de modéliser chaque domaine. Le transfert de la source à la cible ne s'opère pas au niveau des données directement, mais plutôt au niveau des modèles, soit au niveau d'une abstraction des données. Ce modèle graphique est interprété au regard des programmes minimaux décrits dans la section précédente.

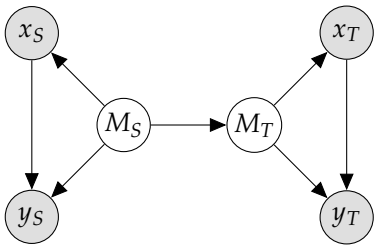


FIGURE B.1: Modèle graphique descriptif pour le raisonnement analogique.

B.1.4 Analogies géométriques

Dans le modèle précédent, nous considérons une description abstraite des données. Nous avons proposé une application au cadre des objets géométriques, en nous concentrant sur des objets de deux natures différentes: les espaces vectoriels et les variétés riemanniennes.

Tous les opérateurs agissant sur de telles structures ne sont pas aussi "naturels". En particulier, les espaces vectoriels sont définis par une relation d'addition, qui correspond donc, dans un codage raisonnable, à une opération de très faible complexité. Poursuivant cette idée, nous avons montré que le principe de minimum de complexité est équivalent, dans un espace vectoriel, à la règle du parallélogramme: $A : B :: C : D \Rightarrow D = C + B - A$. Cette règle est l'une des premières à avoir été utilisées dans le domaine de l'analogie (Rumelhart and Abrahamson, 1973), et est l'une des illustrations principales d'une modélisation classique de l'analogie : l'analogie proportionnelle.

Suivant cette idée, deux opérations élémentaires dans un espace pourraient être l'application exponentielle et le transport parallèle. Nous avons donc proposé un algorithme à partir de ces deux opérations, et qui correspond à la fois à une application du principe de résolution des analogies par minimisation de la complexité, mais aussi à une généralisation de la règle du parallélogramme à des espaces non-euclidiens. Etudiant cet algorithme, nous avons constaté qu'il ne peut pas être utilisé pour définir des analogies proportionnelles, en général. Nous avons montré qu'il ne définit des analogies proportionnelles que dans le cas d'espaces plats au sens de Ricci. Nous avons par ailleurs illustré ce résultat à l'aide d'une application sur la variété de Fisher.

Le résultat précédent stipule qu'un algorithme intuitif, reposant sur la notion de minimisation de la complexité, ne permet pas de définir des analogies proportionnelles en général. Nous concluons notre travail sur l'analogie par une réflexion plus générale : Peut-on définir des analogies proportionnelles sur toute variété riemannienne ? Nous avons montré que cela est en effet possible. Néanmoins, notre preuve propose des relations qui ne sont pas continues : Si A_1 et A_2 sont proches, nous n'avons aucune garantie que les solutions de $A_1 : B :: C : x$ et $A_1 : B :: C : x$, où x est la variable à déterminer, soient également proches. Cette propriété additionnelle peut être cependant importante, mais nous n'avons pu montrer ni infirmer la conjecture selon laquelle il n'est pas toujours possible de définir des analogies proportionnelles continues sur une variété riemannienne.

B.2 De l'analogie à l'apprentissage par transfert

Dans cette section, nous présentons une extension du principe développé pour le raisonnement par analogie à la tâche d'apprentissage par transfert. Pour ce faire, nous présentons tout d'abord la question du transfert en apprentissage automatique, puis nous discutons l'utilisation de notre principe de raisonnement. Enfin, nous cherchons à dépasser la restriction du transfert et à étendre notre principe au raisonnement inductif.

B.2.1 Introduction à l'apprentissage par transfert

L'apprentissage par transfert est un problème d'apprentissage machine partant d'une hypothèse simple : la machine apprend un modèle à partir de données (appelées données *source*), puis doit appliquer ce modèle sur de nouvelles données (appelées

données *cible*), mais de distributions différentes. Il s'oppose à des méthodes plus classiques qui reposent sur l'hypothèse d'une même distribution entre source et cible.

C'est un problème que l'on rencontre fréquemment en pratique. En vision par ordinateur, par exemple, il est fréquent d'avoir des bases de données d'images de résolutions très différentes, ou avec des variations importantes dans l'éclairage, le contexte, le style ou même l'orientation des objets représentés.

Le transfert est important, principalement lorsque les données cible sont en faible quantité et ne permettent pas d'inférer un modèle sûr à partir d'elles seules.

Dans le cadre de cette thèse, nous nous sommes essentiellement intéressés au cas de l'adaptation de domaine non-supervisée. Dans ce problème, les données source sont étiquetées: chaque point est fourni avec une classe. L'enjeu est alors de trouver une fonction représentant correctement la transformation entre un objet et sa classe. En appliquant cette fonction à tout nouvel objet de même distribution, nous pouvons correctement étiqueter un nouvel élément. En adaptation de domaine non-supervisée, le nouvel élément est supposé ne pas suivre la même distribution que les données source. Il n'est pas non plus étiqueté.

L'apprentissage par transfert pose un certain nombre de questions, et ce malgré une formalisation théorique fondée sur la théorie de l'apprentissage statistique. L'une de ces difficultés est le transfert négatif. Dans certains cas, il peut être observé que le transfert nuit à la qualité de la prédiction.

B.2.2 Apprentissage par transfert et principe de longueur de description minimale

L'adaptation de domaine non-supervisée et le raisonnement par analogie ont beaucoup en commun. À partir d'observations d'un problème et d'une solution dans un domaine source, ils interpolent le modèle estimé pour trouver la solution à un problème dans un cas cible, supposé différent de la source. La différence est qu'en apprentissage par transfert, le problème consiste en un jeu de données non étiqueté, et la solution aux étiquettes attribuées.

Cette observation indique qu'un même principe de résolution des deux problèmes semble pouvoir être envisagé. Nous avons donc suggéré d'utiliser le modèle graphique descriptif précédemment décrit dans le cadre de l'apprentissage par transfert. Ce choix est intéressant car il peut s'interpréter dans les termes de principes inductifs connus dans le cadre de l'apprentissage traditionnel, en particulier les principes de minimisation du risque empirique et de maximum de vraisemblance.

Pour nos applications, nous avons proposé un modèle par prototypes, inspiré par les modèles de cartes auto-organisatrices (Kohonen, 1990). Dans ces modèles, un point est représenté relativement à un objet abstrait, appelé *prototype*. On suppose que le prototype porte l'information factorisée entre les différents points qui lui sont raccrochés. Nous avons également proposés différents algorithmes pour estimer les valeurs de ces prototypes, et ce sous différentes hypothèses de transformation entre la source et la cible.

Dans différentes expériences sur des données synthétiques, ces algorithmes ont donné des résultats tout à fait satisfaisants et acceptables. Du fait de la simplicité de l'approche proposée, nous n'avons pas testé sur des jeaux de données plus complexes, sur lesquels le principe proposé pourrait se réléver d'assez faible qualité.

B.2.3 Au-delà du transfert: Apprentissage avec non-indifférence à la question future

Le modèle proposé pour le raisonnement par analogie ainsi que pour l'apprentissage par transfert présente un certain nombre d'avantages. Parmi eux, nous avons constaté qu'il permet de définir simplement certaines notions fondamentales, parmi lesquelles les notions de transférabilité et de transfert négatif.

Nous avons proposé deux définitions alternatives de la transférabilité, appelées respectivement transférabilité forte et transférabilité faible. Elles reposent toutes deux sur une idée commune : une observation source est transférable à un problème cible à la condition qu'il existe une modélisation de la source dont la donnée permette de comprimer la représentation de la cible.

Le transfert négatif ne correspond cependant pas à une non-transférabilité. Au contraire, un transfert négatif est observé à la condition qu'un transfert soit possible, mais l'absence d'étiquettes en cible rend le transfert mauvais. Autrement dit, un transfert est possible, mais son résultat ne correspond pas à la solution attendue.

Nous avons utilisé cette idée de transférabilité pour décrire le problème de la généralisation. Ce problème interroge la possibilité d'inférer une règle générale à partir d'observations. La grande différence avec les problèmes précédemment cités est qu'ici aucune cible n'est *a priori* connue : la règle doit être inférée avant même qu'une cible ne soit observée. Nous pouvons modéliser la généralisation comme le transfert vers un grand nombre de cibles estimées au moment du processus de généralisation. En utilisant des notions d'optimisation multi-objectifs, nous avons montré qu'il est possible de modéliser la généralisation comme un transfert non pas vers une cible, mais vers l'espérance d'une cible. Le problème est donc de définir la probabilité *a priori* sur les tâches futures. Puisque ce principe estime, au moment de l'apprentissage, ce que pourrait être l'utilisation future de son apprentissage, nous avons appelé ce principe *apprentissage avec non-indifférence à la question future*.

B.3 Apprentissage incrémental

Dans cette section, nous nous intéressons à des environnements produisant les données sous forme d'un flux continu, au long duquel les distributions des données sont susceptibles d'évoluer. Nous proposons dans un premier temps une extension directe du principe de transfert par complexité minimale. Nous proposons ensuite deux cadres d'étude : une application aux systèmes de recommandation et une analyse d'un phénomène connu en sciences cognitives, l'apprentissage en forme de U.

B.3.1 De l'apprentissage par transfert à l'apprentissage incrémental

Une problématique nouvelle en sciences des données a émergé du fait d'acquisitions de données sous forme de flux. En apprentissage classique, les données sont toutes accessibles au moment de l'apprentissage. Dans certains systèmes, ce n'est pas le cas, et les données sont produites une à une et en temps réel, potentiellement à de très hautes fréquences. Dans de telles configurations, on observe souvent un phénomène spécifique : la *dérive de concept*. La dérive de concept correspond à l'évolution de la distribution des données au cours du temps. Une telle évolution peut être soudaine (on parle alors de *dérive abrupte*), ou progressive (on parle de *dérive incrémentale*).

L'apprentissage incrémental, sur flux de données, peut être interprété, dans une certaine mesure, comme une succession de transferts, d'un point au suivant, ou dans des fenêtres glissantes. Réutilisant le modèle graphique descriptif proposé pour

l'analogie et le transfert, nous avons proposé une modélisation de l'apprentissage incrémental, dans lequel chaque nouveau point est décrit par rapport à la connaissance du passé plutôt que par rapport à une source fixe. Nous avons pu vérifier que, selon le choix de l'abstraction des données, notre modélisation pouvait couvrir une large classe de méthodes existantes, tant reposant sur la détection des dérives de concept (méthodes dites *actives*) que sur une adaptation continue aux nouvelles données (méthodes dites *passives*).

B.3.2 Recommandation incrémentale hybride

En utilisant notre modélisation de l'apprentissage incrémental, nous avons pu en particulier montrer qu'elle décrit une méthode bien connue dans l'état de l'art, ADWIN (Bifet and Gavalda, 2007). Nous avons proposé une nouvelle application de cet algorithme dans un contexte non encore exploré.

La première application est celle de la détection de changements en modélisation thématique (*topic modeling*). L'enjeu de ce problème est de découvrir des thèmes sous-jacents dans des documents textuels. Lorsque les textes sont tous disponibles, il est possible d'utiliser l'algorithme LDA (Blei, Ng, and Jordan, 2003). Néanmoins, aucune méthode n'existait pour résoudre ce problème dans un contexte de flux de données avec dérive de concepts. Notre idée consiste à utiliser LDA sur les mesures de vraisemblance des données pour adapter notre modèle thématique lors que celui-ci devient obsolète. Nous avons pu tester, tant sur des données synthétiques que réelles, qu'une telle adaptation a un apport important et ne peut être négligée. Nous avons par ailleurs proposé une variante de notre méthode au cas où les dérives de concept seraient récurrentes et où des modèles thématiques reviendraient au cours du temps. Cette modélisation s'inspire de la formalisation de l'apprentissage à partir de cas.

Nous avons intégré cet algorithme dans une technique de recommandation hybride incrémentale. La recommandation est un problème qui consiste à associer des produits à des individus en fonction de leurs goûts. Elle est souvent divisée en deux classes de méthodes : les méthodes basées sur le contenu, qui exploitent une description des produits à recommander, et le filtrage collaboratif qui n'exploite pas les informations des produits mais uniquement les similarités de comportements des individus. Les méthodes hybrides utilisent ces deux approches simultanément. La question de l'incrémental en recommandation est par ailleurs une question assez peu soulevée dans l'état de l'art. Notre contribution a été de proposer un modèle hybride, exploitant la factorisation de matrice incrémentale pour le filtrage collaboratif, et notre algorithme de modélisation thématique incrémentale pour la description des contenus. Nous avons montré, par le biais d'expériences sur données synthétiques et réelles, que cette approche est très supérieure à d'autres approches ne prénant pas en compte la dérive de concept. En particulier, nous avons observé que les méthodes hybrides devenaient moins effectives avec le temps, en particulier moins qu'un simple filtrage collaboratif.

B.3.3 Apprentissage incrémental en forme de U

La deuxième application que nous avons proposée pour l'apprentissage incrémental repose sur les idées développées plus haut pour les analogies sur des chaînes de caractères. Nous avons cherché à modéliser l'acquisition du langage. En particulier, nous avons cherché à reproduire un résultat cognitif bien connu, celui de l'apprentissage en forme de U. Ce terme désigne un phénomène d'apprentissage,

désapprentissage puis réapprentissage, très caractéristique de l'acquisition du langage. Intuitivement, la première phase consiste à établir des règles très précises pour expliquer les observations grammaticales; la seconde à supprimer des règles afin d'obtenir une généralisation; et la troisième à corriger les erreurs faites dans la seconde phase.

Afin de calculer les complexités dans notre modèle graphique descriptif, nous avons exploité le langage de description des chaînes de caractères proposé pour l'analogie. Nous avons proposé un codage simple de la grammaire sous forme de listes de règles écrites comme des fragments de programme. Nous avons également supposé l'existence d'une mémoire à court terme. Une application expérimentale nous a permis de constater que le phénomène d'apprentissage en forme de U est effectivement observé sous ces conditions, et dépend fortement des paramètres de l'apprentissage (en particulier de la taille de la mémoire).

B.4 Transfert d'information en apprentissage non-supervisé

Dans cette section, nous traitons du problème de l'échange d'information entre différents algorithmes d'apprentissage non-supervisé. Après une présentation du problème du clustering multi-sources, nous proposons un échange d'information avec minimum de complexité. Nous concluons cette section par une réflexion sur la possibilité de collaboration entre algorithmes de clustering.

B.4.1 Introduction au clustering multi-sources

Le clustering est une tâche d'apprentissage non-supervisé qui consiste à grouper entre eux des objets similaires. Il s'agit d'un problème mal posé et pourtant essentiel dans nombre d'applications pratiques. Bien que différentes familles de méthodes existent, il a été montré qu'un algorithme de clustering ne peut pas satisfaire tous les critères de qualité à la fois (Kleinberg, 2003). Chaque méthode est donc biaisée vers certains types de problèmes.

Il a donc été proposé d'utiliser de la collaboration afin de surmonter ces biais individuels. Cette idée de collaboration a aussi été introduite dans le cadre d'environnements distribués. Deux classes de méthodes ont donc vu le jour. La première s'intéresse à trouver un consensus entre différents algorithmes de clustering, c'est-à-dire à unifier les solutions de chaque algorithme afin d'avoir une partition commune. La seconde, au contraire, ne cherche qu'à corriger des erreurs locales, mais ne cherche pas de consensus entre les différents agents opérant le clustering. Dans la suite, nous nous intéressons exclusivement à cette seconde classe de méthodes, couramment appelée *clustering collaboratif*.

B.4.2 Clustering multi-source de complexité minimale

Le clustering peut être vu comme une tâche de compression : si deux objets se ressemblent, ils peuvent être co-comprimés. Il est donc tentant d'affirmer qu'un bon clustering devrait minimiser la complexité d'un jeu de données, à un biais près (correspondant au choix de la machine). Suivant cette idée, nous avons utilisé la complexité comme mesure étalon pour comparer les résultats de différents clusterings opérant sur les mêmes données. Cette idée, qui peut se traduire par un modèle graphique descriptif simple, alliée à une simplification, nous permet d'obtenir une méthode générique et très simple pour résoudre le problème du clustering collaboratif.

L'algorithme que nous proposons consiste à établir un système de règles et d'exceptions entre les solutions des différents clusterings, et à procéder à une correction des exceptions, si la correction permet d'abaisser la complexité. Nous avons pu tester cette méthode sur des jeux de données synthétiques et réels et constater ses grandes performances, en particulier comparé à d'autres méthodes de l'état de l'art.

B.4.3 Possibilité de collaboration pour les algorithmes de clustering

La notion de collaboration en clustering est une notion difficile. En apprentissage supervisé, il peut être montré que la collaboration aide à l'apprentissage. Cela est dû en particulier au fait qu'une mesure supervisée est connue par le système apprenant. Le système sait en effet si les résultats sont corrects ou non. En clustering, en revanche, il est impossible de juger de la qualité d'une méthode, sinon par des indices biaisés et sans valeur générale. Nous nous sommes donc interrogés sur le gain réel apporté par une collaboration. Nous avons exploité deux pistes.

La première est celle de la réflexion sur les choix des collaborateurs. Nous avons proposé une modélisation simple du choix de collaborateurs. Dans ce contexte, nous supposons que chaque algorithme de clustering aurait possibilité de donner un poids aux autres algorithmes afin de sélectionner les meilleurs collaborateurs. Nous avons montré théoriquement que, dans ce modèle, un algorithme aurait tendance à privilégier les collaborations avec les algorithmes donnant les solutions les plus semblables. Cette observation peut sembler contre-intuitive : on s'attendrait en effet à ce qu'un algorithme cherche à collaborer avec des méthodes apportant une certaine diversité. Pourtant, ici le choix d'une stabilité est fait.

La seconde est celle d'une réflexion sur la stabilité : l'idée est qu'une collaboration devrait augmenter la stabilité globale de la collaboration. Pour mesurer la stabilité, nous nous sommes appuyés sur les travaux de (Ben-David, Von Luxburg, and Pál, 2006) et les avons étendus pour recouvrir le domaine du clustering collaboratif. Nous avons en outre proposé quelques résultats préliminaires. En particulier, nous avons montré qu'une "faible" collaboration d'algorithmes stables restait stable. Cette notion imprécise de faiblesse est formalisée sous le nom de *consistance* et mesure l'écart entre la collaboration et l'absence de collaboration. La consistance est donc d'autant plus faible que la collaboration ne change rien aux décisions individuelles. Ces résultats sont préliminaires et devront être complétés dans de futurs travaux.

Bibliography

Al-Ghossein, Marie, Pierre-Alexandre Murena, Talel Abdessalem, Anthony Barré, and Antoine Cornuéjols (2018). "Adaptive collaborative topic modeling for online recommendation". In: *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, pp. 338–346.

Al-Ghossein, Marie, Pierre-Alexandre Murena, Antoine Cornuéjols, and Talel Abdessalem. "Online Learning with Reoccurring Drifts: The Perspective of Case-Based Reasoning". In: *ICCBR 2018* (), pp. 133–142.

Alippi, Cesare and Manuel Roveri (2006). "An adaptive cusum-based test for signal change detection". In: *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*. IEEE, 4–pp.

Allison, Lloyd (2018). *Coding Ockham's Razor*. Springer.

AlSumait, Loulwah, Daniel Barbará, and Carlotta Domeniconi (2008). "On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking". In: *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, pp. 3–12.

Amari, Shun-ichi (2012). *Differential-geometrical methods in statistics*. Vol. 28. Springer Science & Business Media.

Ankerst, Mihael, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander (1999). "OPTICS: ordering points to identify the clustering structure". In: *ACM Sigmod record*. Vol. 28. 2. ACM, pp. 49–60.

Ayad, Hanan G and Mohamed S Kamel (2010). "On voting-based consensus of cluster ensembles". In: *Pattern Recognition* 43.5, pp. 1943–1953.

Babcock, Brian, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom (2002). "Models and issues in data stream systems". In: *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, pp. 1–16.

Baliga, Ganesh, John Case, Wolfgang Merkle, Frank Stephan, and Rolf Wiehagen (2008). "When unlearning helps". In: *Information and Computation* 206.5, pp. 694–709.

Bao, Yang, Hui Fang, and Jie Zhang (2014). "TopicMF: simultaneously exploiting ratings and reviews for recommendation". In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, pp. 2–8.

Baxter, Rohan A and Jonathan J Oliver (1994). "MDL and MML: Similarities and differences". In: *Dept. Comput. Sci. Monash Univ., Clayton, Victoria, Australia, Tech. Rep* 207.

Bayoudh, Meriam, Henri Prade, and Gilles Richard (2012). "Evaluation of analogical proportions through Kolmogorov complexity". In: *Knowledge-Based Systems* 29, pp. 20–30.

Ben-David, Shai, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan (2010). "A theory of learning from different domains". In: *Machine learning* 79.1-2, pp. 151–175.

Ben-David, Shai, Ulrike Von Luxburg, and Dávid Pál (2006). "A sober look at clustering stability". In: *International Conference on Computational Learning Theory*. Springer, pp. 5–19.

Ben-Hur, Asa, Andre Elisseeff, and Isabelle Guyon (2001). "A stability based method for discovering structure in clustered data". In: *Biocomputing 2002*. World Scientific, pp. 6–17.

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston (2009). "Curriculum learning". In: *Proceedings of the 26th annual international conference on machine learning*. ACM, pp. 41–48.

Beringer, Jürgen and Eyke Hüllermeier (2007). "Efficient instance-based learning on data streams". In: *Intelligent Data Analysis* 11.6, pp. 627–650.

Berkhin, Pavel (2006). "A survey of clustering data mining techniques". In: *Grouping multidimensional data*. Springer, pp. 25–71.

Besag, Julian (1974). "Spatial interaction and the statistical analysis of lattice systems". In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 192–236.

Bessa, Ricardo Jorge, Vladimiro Miranda, and Joao Gama (2009). "Entropy and correntropy against minimum square error in off-line and on-line 3-day ahead wind power forecasting". In: *IEEE Trans. on Power Syst* 24.4, pp. 1657–1666.

Bickel, Steffen, Michael Brückner, and Tobias Scheffer (2007). "Discriminative learning for differing training and test distributions". In: *Proceedings of the 24th international conference on Machine learning*. ACM, pp. 81–88.

Bickel, Steffen and Tobias Scheffer (2005). "Estimation of mixture models using Co-EM". In: *European Conference on Machine Learning*. Springer, pp. 35–46.

Biederman, Irving (1987). "Recognition-by-components: a theory of human image understanding." In: *Psychological review* 94.2, p. 115.

Bifet, Albert and Ricard Gavalda (2007). "Learning from time-changing data with adaptive windowing". In: *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, pp. 443–448.

Blei, David M and John D Lafferty (2006). "Dynamic topic models". In: *Proceedings of the 23rd international conference on Machine learning*. ACM, pp. 113–120.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Blitzer, John, Ryan McDonald, and Fernando Pereira (2006). "Domain adaptation with structural correspondence learning". In: *Proceedings of the 2006 conference on empirical methods in natural language processing*. Association for Computational Linguistics, pp. 120–128.

Bock, J Kathryn (1986). "Syntactic persistence in language production". In: *Cognitive psychology* 18.3, pp. 355–387.

Bohannon, John N and Laura B Stanowicz (1988). "The issue of negative evidence: Adult responses to children's language errors." In: *Developmental psychology* 24.5, p. 684.

Boothby, William M (1986). *An introduction to differentiable manifolds and Riemannian geometry*. Vol. 120. Academic press.

Bottou, Léon (2010). "Large-scale machine learning with stochastic gradient descent". In: *Proceedings of COMPSTAT'2010*. Springer, pp. 177–186.

Bouchachia, Abdelhamid (2011). "Fuzzy classification in dynamic environments". In: *Soft Computing* 15.5, pp. 1009–1022.

Bounhas, Myriam, Henri Prade, and Gilles Richard (2017). "Analogy-based classifiers for nominal or numerical data". In: *International Journal of Approximate Reasoning* 91, pp. 36–55.

Bowerman, Melissa (1982). "Starting to talk worse: Clues to language acquisition from children's late speech errors". In: *U shaped behavioral growth*. Academic Press, pp. 101–145.

— (1988). "The'no negative evidence'problem: How do children avoid constructing an overly general grammar?" In: *Explaining language universals*. Basil Blackwell, pp. 73–101.

Branigan, Holly P, Martin J Pickering, and Alexandra A Cleland (1999). "Syntactic priming in written production: Evidence for rapid decay". In: *Psychonomic Bulletin & Review* 6.4, pp. 635–640.

Brzezinski, Dariusz and Jerzy Stefanowski (May 2014). "Combining Block-based and Online Methods in Learning Ensembles from Concept Drifting Data Streams". In: *Inf. Sci.* 265, pp. 50–67.

Buhrman, Harry, Lance Fortnow, and Sophie Laplante (2001). "Resource-Bounded Kolmogorov Complexity Revisited". In: *SIAM J. Comput.* 31.3, pp. 887–905.

Cao, Huanhuan, Enhong Chen, Jie Yang, and Hui Xiong (2009). "Enhancing recommender systems under volatile userinterest drifts". In: *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, pp. 1257–1266.

Cao, Liangliang, Zicheng Liu, and Thomas S Huang (2010). "Cross-dataset action detection". In:

Carey, Susan (1982). "Face perception: Anomalies of development". In: *U-shaped behavioral growth*, pp. 169–190.

Carlucci, Lorenzo and John Case (2013). "On the Necessity of U-Shaped Learning". In: *Topics in cognitive Science* 5.1, pp. 56–88.

Carlucci, Lorenzo, John Case, Sanjay Jain, and Frank Stephan (2007). "Results on memory-limited U-shaped learning". In: *Information and Computation* 205, pp. 1551–1573.

Caruana, Rich (1997). "Multitask learning". In: *Machine learning* 28.1, pp. 41–75.

Case, John and Timo Kötzing (2010). "Strongly Non-U-Shaped Learning Results by General Techniques." In: *COLT*, pp. 181–193.

Cavey, Joris Van de and Robert J Hartsuiker (2016). "Is there a domain-general cognitive structuring system? Evidence from structural priming across music, math, action descriptions, and language". In: *Cognition* 146, pp. 172–184.

Cencov, Nikolai Nikolaevich (2000). *Statistical decision rules and optimal inference*. 53. American Mathematical Soc.

Chalmers, David J, Robert M French, and Douglas R Hofstadter (1992). "High-level perception, representation, and analogy: A critique of artificial intelligence methodology". In: *Journal of Experimental & Theoretical Artificial Intelligence* 4.3, pp. 185–211.

Chang, Jonathan, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei (2009). "Reading tea leaves: How humans interpret topic models". In: *Advances in neural information processing systems*, pp. 288–296.

Chater, Nick (1999). "The search for simplicity: A fundamental cognitive principle?" In: *The Quarterly Journal of Experimental Psychology: Section A* 52.2, pp. 273–302.

Chen, Stanley F and Joshua Goodman (1996). "An empirical study of smoothing techniques for language modeling". In: *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 310–318.

Chi, Michelene TH, Robert Glaser, and Marshall J Farr (2014). *The nature of expertise*. Psychology Press.

Cilibrasi, Rudi and Paul Vitanyi (2006). "Automatic meaning discovery using Google". In: *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Cleuziou, Guillaume, Matthieu Exbrayat, Lionel Martin, and Jacques-Henri Sublemontier (2009). "CoFKM: A centralized method for multiple-view clustering". In: *2009 Ninth IEEE International Conference on Data Mining*. IEEE, pp. 752–757.

Cornuéjols, Antoine and Jacques Ales-Bianchetti (1998). "Analogy and Induction : which (missing) link?" In: *Workshop "Advances in Analogy Research : Integration of Theory and Data from Cognitive, Computational and Neural Sciences"*. Sofia, Bulgaria.

Cornuejols, Antoine, Cédric Wemmert, Pierre Gançarski, and Younès Bennani (2018). "Collaborative clustering: Why, when, what and how". In: *Information Fusion* 39, pp. 81–95.

Couceiro, Miguel, Nicolas Hug, Henri Prade, and Gilles Richard (2018). "Behavior of Analogical Inference wrt Boolean Functions." In: *IJCAI*, pp. 2057–2063.

Council, National Research et al. (2013). *Frontiers in massive data analysis*. National Academies Press.

Courty, Nicolas, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy (2017). "Joint distribution optimal transportation for domain adaptation". In: *Advances in Neural Information Processing Systems*, pp. 3730–3739.

Courty, Nicolas, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy (2017). "Optimal transport for domain adaptation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.9, pp. 1853–1865.

Cuturi, Marco (2013). "Sinkhorn distances: Lightspeed computation of optimal transport". In: *Advances in neural information processing systems*, pp. 2292–2300.

Da Silva, Alzennyr, Yves Lechevallier, Fabrice Rossi, and Francisco de Carvalho (2007). "Construction and analysis of evolving data summaries: An application on web usage data". In: *Intelligent Systems Design and Applications, 2007. ISDA 2007. Seventh International Conference on*. IEEE, pp. 377–380.

Davies, David L and Donald W Bouldin (1979). "A cluster separation measure". In: *IEEE transactions on pattern analysis and machine intelligence* 2, pp. 224–227.

Davies, Todd and Stuart Russell (1987). "A Logical Approach to Reasoning by Analogy". In: *Proc. of the 10th IJCAI*. Milan, Italy, pp. 264–270.

De Bot, Kees (2000). "A bilingual production model: Levelt's "speaking" model adapted". In: *The bilingualism reader*, pp. 420–442.

De Mantaras, Ramon Lopez et al. (2005). "Retrieval, reuse, revision and retention in case-based reasoning". In: *The Knowledge Engineering Review* 20.3, pp. 215–240.

Delany, Sarah Jane and Pádraig Cunningham (2004). "An analysis of case-base editing in a spam filtering system". In: *European Conference on Case-Based Reasoning*. Springer, pp. 128–141.

Delany, Sarah Jane, Pádraig Cunningham, Alexey Tsymbal, and Lorcan Coyle (2005). "A case-based technique for tracking concept drift in spam filtering". In: *Knowledge-based systems* 18.4-5, pp. 187–195.

Demetras, Marty J, Kathryn Nolan Post, and Catherine E Snow (1986). "Feedback to first language learners: The role of repetitions and clarification questions". In: *Journal of child language* 13.2, pp. 275–292.

Dempster, Arthur P, Nan M Laird, and Donald B Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38.

Depaire, Benoît, Rafael Falcón, Koen Vanhoof, and Geert Wets (2011). "PSO driven collaborative clustering: A clustering algorithm for ubiquitous environments". In: *Intelligent Data Analysis* 15.1, pp. 49–68.

Dessalles, Jean-Louis (2013). "Algorithmic simplicity and relevance". In: *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence*. Springer, pp. 119–130.

Dietterich, Thomas G (2000). "Ensemble methods in machine learning". In: *International workshop on multiple classifier systems*. Springer, pp. 1–15.

Ding, Yi and Xue Li (2005). "Time weight collaborative filtering". In: *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, pp. 485–492.

Ditzler, Gregory, Gail Rosen, and Robi Polikar (2014). "Domain adaptation bounds for multiple expert systems under concept drift". In: *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, pp. 595–601.

Ditzler, Gregory, Manuel Roveri, Cesare Alippi, and Robi Polikar (2015). "Learning in nonstationary environments: A survey". In: *IEEE Computational Intelligence Magazine* 10.4, pp. 12–25.

Domeniconi, Carlotta and Muna Al-Razgan (2009). "Weighted cluster ensembles: Methods and analysis". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 2.4, p. 17.

Domingos, Pedro and Geoff Hulten (2000). "Mining high-speed data streams". In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 71–80.

Drozd, Aleksandr, Anna Gladkova, and Satoshi Matsuoka (2016). "Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 3519–3530.

Drummond, Alex (2013). "Ibex farm". In: *Online server: http://spellout. net/ibexfarm*.

Du, Lan, Wray Buntine, Huidong Jin, and Changyou Chen (2012). "Sequential latent Dirichlet allocation". In: *Knowledge and information systems* 31.3, pp. 475–503.

Ehrgott, M. (2000). *Multicriteria optimization*. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag.

Elwell, Ryan and Robi Polikar (2009). "Incremental Learning of Variable Rate Concept Drift". In: *Multiple Classifier Systems: 8th International Workshop, MCS 2009, Reykjavik, Iceland, June 10-12, 2009. Proceedings*. Ed. by Jón Atli Benediktsson, Josef Kittler, and Fabio Roli. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 142–151.

— (2011). "Incremental learning of concept drift in nonstationary environments". In: *IEEE Transactions on Neural Networks* 22.10, pp. 1517–1531.

Ester, Martin, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *KDD*. Vol. 96. 34, pp. 226–231.

Falkenhainer, Brian, Kenneth D Forbus, and Dedre Gentner (1989). "The structure-mapping engine: Algorithm and examples". In: *Artificial intelligence* 41.1, pp. 1–63.

Fei-Fei, Li (2006). "Knowledge transfer in learning to recognize visual objects classes". In: *Proceedings of the International Conference on Development and Learning (ICDL)*, p. 11.

Feng, Yansong and Mirella Lapata (2010). "Topic models for image annotation and text illustration". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 831–839.

Fisher, Ronald Aylmer (1925). "Theory of statistical estimation". In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 22. 5. Cambridge University Press, pp. 700–725.

Forbus, K, R Ferguson, and Dedre Gentner (1994). "Incremental structure-mapping". In: *Proceedings of the Cognitive Science Society*.

Fred, Ana (2001). "Finding consistent clusters in data partitions". In: *International Workshop on Multiple Classifier Systems*. Springer, pp. 309–318.

French, Robert M (2002). "The computational modeling of analogy-making". In: *Trends in cognitive Sciences* 6.5, pp. 200–205.

Frey, Brendan J and Delbert Dueck (2007). "Clustering by passing messages between data points". In: *science* 315.5814, pp. 972–976.

Frigó, Erzsébet, Róbert Pálovics, Domokos Kelen, Levente Kocsis, and András A. Benczúr (2017). "Online Ranking Prediction in Non-stationary Environments". In: *Proceedings of the 1st Workshop on Temporal Reasoning in Recommender Systems co-located with 11th International Conference on Recommender Systems (RecSys 2017)*. ACM, pp. 28–34.

Galton, Francis (1907). "Vox populi (The wisdom of crowds)". In: *Nature* 75.7, pp. 450–451.

Gama, João and Petr Kosina (2009). "Tracking recurring concepts with meta-learners". In: *Portuguese Conference on Artificial Intelligence*. Springer, pp. 423–434.

Gama, João, Raquel Sebastião, and Pedro Pereira Rodrigues (2009). "Issues in evaluation of stream learning algorithms". In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 329–338.

Gama, João, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia (2014). "A survey on concept drift adaptation". In: *ACM computing surveys (CSUR)* 46.4, p. 44.

Gammerman, A., V. Vovk, and V. Vapnik (1998). "Learning by Transduction". In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. UAI'98. Madison, Wisconsin: Morgan Kaufmann Publishers Inc., pp. 148–155.

Ganesan, Kavita and ChengXiang Zhai (2012). "Opinion-based entity ranking". In: *Information retrieval* 15.2, pp. 116–150.

Ganin, Yaroslav and Victor Lempitsky (2015). "Unsupervised Domain Adaptation by Backpropagation". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1180–1189.

Gantz, John and David Reinsel (2012). "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east". In: *IDC iView: IDC Analyze the future* 2007.2012, pp. 1–16.

Gärdenfors, Peter (2004). *Conceptual spaces: The geometry of thought*. MIT press.

Gentner, Dedre (1983). "Structure-mapping: A theoretical framework for analogy". In: *Cognitive science* 7.2, pp. 155–170.

Gentner, Dedre and Kenneth D Forbus (2011). "Computational models of analogy". In: *Wiley interdisciplinary reviews: cognitive science* 2.3, pp. 266–276.

Germain, Pascal, Amaury Habrard, François Laviolette, and Emilie Morvant (2013). "A PAC-Bayesian approach for domain adaptation with specialization to linear classifiers". In: *International conference on machine learning*, pp. 738–746.

Ghassany, Mohamad, Nistor Grozavu, and Younès Bennani (2012a). "Collaborative Clustering using Prototype-Based Techniques". In: *International Journal of Computational Intelligence and Applications* 11.3.

— (2012b). "Collaborative generative topographic mapping". In: *International Conference on Neural Information Processing*. Springer, pp. 591–598.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). "Domain adaptation for large-scale sentiment classification: A deep learning approach". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 513–520.

Gold, E Mark (1967). "Language identification in the limit". In: *Information and control* 10.5, pp. 447–474.

Gomes, Heitor Murilo, Jean Paul Barddal, Fabrício Enembreck, and Albert Bifet (2017). "A survey on ensemble learning for data stream classification". In: *ACM Computing Surveys (CSUR)* 50.2, p. 23.

Grbovic, Mihajlo and Slobodan Vucetic (2009). "Learning vector quantization with adaptive prototype addition and removal". In: *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. IEEE, pp. 994–1001.

Griffiths, Thomas L and Mark Steyvers (2004). "Finding scientific topics". In: *Proceedings of the National academy of Sciences* 101.suppl 1, pp. 5228–5235.

Grozavu, Nistor and Younès Bennani (2010). "Topological Collaborative Clustering". In: *Australian Journal of Intelligent Information Processing Systems* 12.3.

Grozavu, Nistor, Guenael Cabanes, and Younes Bennani (2014). "Diversity analysis in collaborative clustering". In: *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, pp. 1754–1761.

Grünwald, Peter D (2007). *The minimum description length principle*. MIT press.

Halkidi, Maria, Yannis Batistakis, and Michalis Vazirgiannis (2002). "Clustering validity checking methods: part II". In: *ACM Sigmod Record* 31.3, pp. 19–27.

Han, Minyeon and Frank C Park (2014). "DTI segmentation and fiber tracking using metrics on multivariate normal distributions". In: *Journal of mathematical imaging and vision* 49.2, pp. 317–334.

Harries, Michael, U Nsw cse tr, and New South Wales (1999). *SPLICE-2 Comparative Evaluation: Electricity Pricing*. Tech. rep.

Harries, Michael Bonnell, Claude Sammut, and Kim Horn (1998). "Extracting hidden context". In: *Machine learning* 32.2, pp. 101–126.

Hartsuiker, Robert J and Herman HJ Kolk (1998). "Syntactic persistence in Dutch". In: *Language and Speech* 41.2, pp. 143–184.

Hartsuiker, Robert J, Martin J Pickering, and Eline Veltkamp (2004). "Is syntax separate or shared between languages? Cross-linguistic syntactic priming in Spanish-English bilinguals". In: *Psychological Science* 15.6, pp. 409–414.

He, Ruining and Julian McAuley (2016). "VBPR: visual Bayesian Personalized Ranking from implicit feedback". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, pp. 144–150.

He, Xiangnan, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua (2016). "Fast matrix factorization for online recommendation with implicit feedback". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, pp. 549–558.

Henley, Nancy M (1969). "A psychological study of the semantics of animal terms". In: *Journal of Memory and Language* 8.2, p. 176.

Hidasi, Balázs and Domonkos Tikk (2012). "Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 67–82.

Hoens, T Ryan, Robi Polikar, and Nitesh V Chawla (2012). "Learning from streaming data with concept drift and imbalance: an overview". In: *Progress in Artificial Intelligence* 1.1, pp. 89–101.

Hoffman, Judy et al. (2018). "CyCADA: Cycle-Consistent Adversarial Domain Adaptation". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed.

by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, pp. 1989–1998.

Hoffman, Matthew, Francis R Bach, and David M Blei (2010). "Online learning for latent dirichlet allocation". In: *advances in neural information processing systems*, pp. 856–864.

Hofmann, Thomas (1999). "Probabilistic latent semantic analysis". In: *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 289–296.

Hofstadter, Douglas (1984). *The Copycat Project: An Experiment in Nondeterminism and Creative Analogies*. AI Memo 755. Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

— (1995). "The architecture of Jumbo". In: *Fluid concepts and creative analogies*. Basic Books, Inc., pp. 97–126.

Hofstadter, Douglas and Melanie Mitchell (1995). "Fluid Concepts and Creative Analogies". In: ed. by Douglas Hofstadter and Corporate Fluid Analogies Research Group. New York, NY, USA: Basic Books, Inc. Chap. The Copycat Project: A Model of Mental Fluidity and Analogy-making, pp. 205–267. ISBN: 0-465-05154-5.

Holyoak, Keith J (2005). "Analogy". In: *The Cambridge Handbook of Thinking and Reasoning*. Ed. by Keith J Holyoak and Robert G Morrison. Cambridge, UK: Cambridge University Press, pp. 117–142.

Holyoak, Keith J and Paul Thagard (1989). "Analogical mapping by constraint satisfaction". In: *Cognitive science* 13.3, pp. 295–355.

Hosseini, Mohammad Javad, Zahra Ahmadi, and Hamid Beigy (2013). "Using a classifier pool in accuracy based tracking of recurring concepts in data stream classification". In: *Evolving Systems* 4.1, pp. 43–60.

Houthuys, Lynn, Zahra Karevan, and Johan AK Suykens (2017). "Multi-view LS-SVM regression for black-box temperature prediction in weather forecasting". In: *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, pp. 1102–1108.

Hsu, Anne S, Nick Chater, and Paul Vitányi (2013). "Language Learning From Positive Evidence, Reconsidered: A Simplicity-Based Approach". In: *Topics in cognitive science* 5.1, pp. 35–55.

Hu, Bo and Martin Ester (2013). "Spatial topic modeling in online social media for location recommendation". In: *Proceedings of the 7th ACM conference on Recommender systems*. ACM, pp. 25–32.

Huang, Jiayuan, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola (2007). "Correcting sample selection bias by unlabeled data". In: *Advances in neural information processing systems*, pp. 601–608.

Hubert, Lawrence and Phipps Arabie (1985). "Comparing partitions". In: *Journal of classification* 2.1, pp. 193–218.

Hug, Nicolas, Henri Prade, and Gilles Richard (2015). "Experimenting analogical reasoning in recommendation". In: *International Symposium on Methodologies for Intelligent Systems*. Springer, pp. 69–78.

Hulten, Geoff, Laurie Spencer, and Pedro Domingos (2001). "Mining time-changing data streams". In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 97–106.

Hummel, John E and Keith J Holyoak (1997). "Distributed representations of structure: A theory of analogical access and mapping." In: *Psychological review* 104.3, p. 427.

Hutter, Marcus (2001). "Towards a universal theory of artificial intelligence based on algorithmic probability and sequential decisions". In: *European Conference on Machine Learning*. Springer, pp. 226–238.

— (2003). "Optimality of universal Bayesian sequence prediction for general loss and alphabet". In: *Journal of Machine Learning Research* 4.Nov, pp. 971–1000.

Hwang, Sung Ju, Kristen Grauman, and Fei Sha (2013). "Analogy-preserving semantic embedding for visual object categorization". In: *International Conference on Machine Learning*, pp. 639–647.

III, Hal Daume (2007). "Frustratingly Easy Domain Adaptation". In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 256–263.

Iwata, Tomoharu, Takeshi Yamada, Yasushi Sakurai, and Naonori Ueda (2010). "Online multiscale dynamic topic models". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 663–672.

Jaber, Ghazal, Antoine Cornuéjols, and Philippe Tarroux (2013). "A new on-line learning method for coping with recurring concepts: the ADACC system". In: *International Conference on Neural Information Processing*. Springer, pp. 595–604.

Jain, Sanjay, Daniel Osherson, James S Royer, and Arun Sharma (1999). *Systems that learn*.

Jelinek, Fred, Robert L Mercer, Lalit R Bahl, and James K Baker (1977). "Perplexity—a measure of the difficulty of speech recognition tasks". In: *The Journal of the Acoustical Society of America* 62.S1, S63–S63.

Kadlec, Petr and Bogdan Gabrys (2011). "Local learning-based adaptive soft sensor for catalyst activation prediction". In: *AIChE Journal* 57.5, pp. 1288–1301.

Kantola, Leila and Roger PG van Gompel (2011). "Between-and within-language priming is the same: Evidence for shared bilingual syntactic representations". In: *Memory & Cognition* 39.2, pp. 276–290.

Kantorovitch, Leonid (1958). "On the translocation of masses". In: *Management Science* 5.1, pp. 1–4.

Katakis, Ioannis, Grigorios Tsouma-kas, and Ioannis Vlahavas (2010). "Tracking recurring contexts using ensemble classifiers: an application to email filtering". In: *Knowledge and Information Systems* 22.3, pp. 371–391.

Kifer, Daniel, Shai Ben-David, and Johannes Gehrke (2004). "Detecting change in data streams". In: *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, pp. 180–191.

Kille, Benjamin, Frank Hopfgartner, Torben Brodt, and Tobias Heintz (2013). "The plista dataset". In: *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*. ACM, pp. 16–23.

Kim, Donghyun, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu (2016). "Convolutional matrix factorization for document context-aware recommendation". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, pp. 233–240.

Kleinberg, Jon M (2003). "An impossibility theorem for clustering". In: *Advances in neural information processing systems*, pp. 463–470.

Koestler, Arthur (1964). "The act of creation". In:

Kohonen, Teuvo (1990). "The self-organizing map". In: *Proceedings of the IEEE* 78.9, pp. 1464–1480.

— (1997). "Learning vector quantization". In: *Self-organizing maps*. Springer, pp. 203–217.

Kolter, J Zico and Marcus A Maloof (2007). "Dynamic weighted majority: An ensemble method for drifting concepts". In: *Journal of Machine Learning Research* 8.Dec, pp. 2755–2790.

Koren, Yehuda (2010). "Collaborative filtering with temporal dynamics". In: *Communications of the ACM* 53.4, pp. 89–97.

Kuehne, Sven, Kenneth Forbus, Dedre Gentner, and Bryan Quinn (2000). "SEQL: Category learning as progressive abstraction using structure mapping". In: *Proceedings of the 22nd annual meeting of the cognitive science society*. Vol. 4.

Kuhn, H. W. and A. W. Tucker (1951). "Nonlinear programming". In: *Proceedings of 2nd Berkeley Symposium*. Ed. by Berkeley University of California Press, pp. 481–492.

Kukar, Matjaž (2003). "Drifting concepts as hidden factors in clinical studies". In: *Conference on Artificial Intelligence in Medicine in Europe*. Springer, pp. 355–364.

Kuo, Mu-Hsing, Liang-Chu Chen, and Chien-Wen Liang (2009). "Building and evaluating a location-based service recommendation system with a preference adjustment mechanism". In: *Expert Systems with Applications* 36.2, pp. 3543–3554.

Kuzborskij, Ilja and Francesco Orabona (2013). "Stability and hypothesis transfer learning". In: *International Conference on Machine Learning*, pp. 942–950.

Lange, Steffen and Thomas Zeugmann (1996). "Incremental learning from positive data". In: *Journal of Computer and System Sciences* 53.1, pp. 88–103.

Langlais, Philippe, François Yvon, and Pierre Zweigenbaum (2009). "Improvements in analogical learning: application to translating multi-terms of the medical domain". In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 487–495.

Lanquillon, Carsten (2001). "Enhancing text classification to improve information filtering". PhD thesis. Otto-von-Guericke-Universität Magdeburg, Universitätsbibliothek.

Lepage, Yves (1998). "Solving analogies on words: an algorithm". In: *Proceedings of the 17th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pp. 728–734.

— (2000). "Languages of analogical strings". In: *Proceedings of the 18th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pp. 488–494.

— (2003). "De l'analogie rendant compte de la commutation en linguistique". PhD thesis. Université Joseph-Fourier-Grenoble I.

— (2004). "Analogy and formal languages". In: *Electronic notes in theoretical computer science* 53, pp. 180–191.

Lepage, Yves and Etienne Denoual (2005). "Purest ever example-based machine translation: Detailed presentation and assessment". In: *Machine Translation* 19.3-4, pp. 251–282.

Leyton, Michael (2001). *A Generative Theory of Shape*. Springer.

Li, Ming and Paul Vitányi (2008). *An introduction to Kolmogorov complexity and its applications*. Vol. 9. Springer, New York.

Loebell, Helga and Kathryn Bock (2003). "Structural priming across languages". In: *Linguistics* 41.5; ISSU 387, pp. 791–824.

Long, Mingsheng, Yue Cao, Jianmin Wang, and Michael I. Jordan (2015). "Learning Transferable Features with Deep Adaptation Networks". In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France, pp. 97–105.

Lu, Ning, Jie Lu, Guangquan Zhang, and Ramon Lopez De Mantaras (2016). "A concept drift-tolerant case-base editing technique". In: *Artificial Intelligence* 230, pp. 108–133.

Luo, Zelun, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei (2017). "Label efficient learning of transferable representations acrosss domains and tasks". In: *Advances in Neural Information Processing Systems*, pp. 165–177.

MacQueen, James et al. (1967). "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA, pp. 281–297.

Mahmud, MM and Sylvian Ray (2008). "Transfer learning using Kolmogorov complexity: Basic theory and empirical evaluations". In: *Advances in neural information processing systems*, pp. 985–992.

Mahmud, MM Hassan (2009). "On universal transfer learning". In: *Theoretical Computer Science* 410.19, pp. 1826–1846.

Mansour, Yishay, Mehryar Mohri, and Afshin Rostamizadeh (2009). "Domain Adaptation: Learning Bounds and Algorithms." In: *COLT*.

Marcus, Gary F (1993). "Negative evidence in language acquisition". In: *Cognition* 46.1, pp. 53–85.

Marcus, Gary F et al. (1992). "Overregularization in language acquisition". In: *Monographs of the society for research in child development*, pp. i–178.

Marshall, James B (2002). "Metacat: A self-watching cognitive architecture for analogy-making". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 24. 24.

Masud, Mohammad, Jing Gao, Latifur Khan, Jiawei Han, and Bhavani M Thuraisingham (2011). "Classification and novel class detection in concept-drifting data streams under time constraints". In: *IEEE Transactions on Knowledge and Data Engineering* 23.6, pp. 859–874.

Matuszyk, Pawel and Myra Spiliopoulou (2014). "Selective forgetting for incremental matrix factorization in recommender systems". In: *International Conference on Discovery Science*. Springer, pp. 204–215.

Matuszyk, Pawel, João Vinagre, Myra Spiliopoulou, Alípio Mário Jorge, and João Gama (2015). "Forgetting methods for incremental matrix factorization in recommender systems". In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. ACM, pp. 947–953.

Miclet, Laurent, Sabri Bayoudh, and Arnaud Delhay (2008). "Analogical Dissimilarity: Definition, Algorithms and Two Experiments in Machine Learning." In: *J. Artif. Intell. Res.(JAIR)* 32, pp. 793–824.

Miclet, Laurent and Henri Prade (2009). "Handling analogical proportions in classical logic and fuzzy logics settings". In: *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*. Springer, pp. 638–650.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig (2013). "Linguistic regularities in continuous space word representations". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751.

Miranda, Catarina and Alípio Mário Jorge (2009). "Item-based and user-based incremental collaborative filtering for web recommendations". In: *Portuguese Conference on Artificial Intelligence*. Springer, pp. 673–684.

Mitchell, Melanie (2001). "Analogy-Making as a Complex Adaptive System". In:

Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard (2017). "Universal Adversarial Perturbations". In: *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, pp. 86–94.

Mu, Xin, Kai Ming Ting, and Zhi-Hua Zhou (2017). "Classification under streaming emerging new classes: A solution using completely-random trees". In: *IEEE Transactions on Knowledge and Data Engineering* 29.8, pp. 1605–1618.

Murena, Pierre-Alexandre, M Al Ghossein, T Abdessalem, and Antoine Cornuéjols (2018). "Adaptive window strategy for topic modeling in document streams". In: *International Joint Conference on Neural Networks*.

Murena, Pierre-Alexandre and Antoine Cornuéjols (2016). "Minimum Description Length Principle applied to structure adaptation for classification under concept drift". In: *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016*, pp. 2842–2849.

Murena, Pierre-Alexandre, Antoine Cornuéjols, and Jean-Louis Dessalles (2017). "Incremental learning with the minimum description length principle". In: *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, pp. 1908–1915.

Murena, Pierre Alexandre, Jean-Louis Dessalles, and Antoine Cornuéjols (2017). "A complexity based approach for solving Hofstadter's analogies". In: *CAW@ ICCBR-2017 Computational Analogy Workshop, at International Conference on Case Based Reasoning*.

Murena, Pierre-Alexandre, Jérémie Sublime, Basarab Matei, and Antoine Cornuéjols (2018). "An Information Theory based Approach to Multisource Clustering." In: *IJCAI*, pp. 2581–2587.

Murez, Zak, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim (2018). "Image to Image Translation for Domain Adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4500–4509.

Muthukrishnan, S, Eric van den Berg, and Yihua Wu (2007). "Sequential change detection on data streams". In: *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*. IEEE, pp. 551–550.

Nagao, Makoto (1984). "A framework of a mechanical translation between Japanese and English by analogy principle". In: *Artificial and human intelligence*, pp. 351–354.

Nasraoui, Olfa, Jeff Cerwinske, Carlos Rojas, and Fabio Gonzalez (2007). "Performance of recommendation systems in dynamic streaming environments". In: *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, pp. 569–574.

Nguyen, Anh, Jason Yosinski, and Jeff Clune (2015). "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436.

Nistor Grozavu, Younès Bennani Mustapha Lebbah (2009). "From variable weighting to cluster characterization in topographic unsupervised learning". In: *in Proc. Proc. of IJCNN09, International Joint Conference on Neural Network*.

Ollivier, Yann (2011). "A visual introduction to Riemannian curvatures and some discrete generalizations". In: *Analysis and Geometry of Metric Measure Spaces: Lecture Notes of the 50th Séminaire de Mathématiques Supérieures (SMS), Montréal*, pp. 197–219.

Orseau, Laurent (2010). "Optimality issues of universal greedy agents with static priors". In: *International Conference on Algorithmic Learning Theory*. Springer, pp. 345–359.

— (2014). "Universal knowledge-seeking agents". In: *Theoretical Computer Science* 519, pp. 127–139.

Pan, Rong et al. (2008). "One-class collaborative filtering". In: *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, pp. 502–511.

Pan, Sinno Jialin and Qiang Yang (2010). "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10, pp. 1345–1359.

Patel, Aniruddh D (2003). "Language, music, syntax and the brain". In: *Nature neuroscience* 6.7, p. 674.

Pechenizkiy, Mykola, Jorn Bakker, Indre Žliobaitė, Andriy Ivannikov, and Tommi Kärkkäinen (2010). "Online mass flow prediction in CFB boilers with explicit detection of sudden concept drift". In: *ACM SIGKDD Explorations Newsletter* 11.2, pp. 109–116.

Pedrycz, Witold (2002). "Collaborative fuzzy clustering". In: *Pattern Recognition Letters* 23.14, pp. 1675–1686.

Pena, José M, Jose Antonio Lozano, and Pedro Larranaga (1999). "An empirical comparison of four initialization methods for the k-means algorithm". In: *Pattern recognition letters* 20.10, pp. 1027–1040.

Peng, Peixi et al. (2016). "Unsupervised cross-dataset transfer learning for person re-identification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1306–1315.

Phan, Xuan-Hieu, Le-Minh Nguyen, and Susumu Horiguchi (2008). "Learning to classify short and sparse text & web with hidden topics from large-scale data collections". In: *Proceedings of the 17th international conference on World Wide Web*. ACM, pp. 91–100.

Pietruczuk, Lena, Leszek Rutkowski, Maciej Jaworski, and Piotr Duda (2017). "How to adjust an ensemble size in stream data mining?" In: *Information Sciences* 381, pp. 46–54.

Pirrelli, Vito and Stefano Federici (1994). "Derivational paradigms in morphonology". In: *Proceedings of the 15th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pp. 234–240.

Plunkett, Kim and Virginia Marchman (1991). "U-shaped learning and frequency effects in a multi-layered perception: Implications for child language acquisition". In: *Cognition* 38.1, pp. 43–102.

Pocock, Adam, Paraskevas Yiapanis, Jeremy Singer, Mikel Luján, and Gavin Brown (2010). "Online non-stationary boosting". In: *International Workshop on Multiple Classifier Systems*. Springer, pp. 205–214.

Potter, Mary C and Linda Lombardi (1998). "Syntactic priming in immediate recall of sentences". In: *Journal of Memory and Language* 38.3, pp. 265–282.

Prade, Henri and Gilles Richard (2009). "Testing Analogical Proportions with Google using Kolmogorov Information Theory." In: *FLAIRS Conference*.

— (2013). "From analogical proportion to logical proportions". In: *Logica Universalis* 7.4, pp. 441–505.

— (2014). *Computational Approaches to Analogical Reasoning: Current Trends*. Vol. 548. Springer.

Prade, Henri, Gilles Richard, and Bing Yao (2012). "Enforcing regularity by means of analogy-related proportions – A new approach to classification". In: *International Journal of Computer Information Systems and Industrial Management Applications* 4, pp. 648–658.

Pratt, Lorien Y, Jack Mostow, Candace A Kamm, and Ace A Kamm (1991). "Direct Transfer of Learned Information Among Neural Networks." In: *AAAI*. Vol. 91, pp. 584–589.

Raina, Rajat, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng (2007). "Self-taught learning: transfer learning from unlabeled data". In: *Proceedings of the 24th international conference on Machine learning*. ACM, pp. 759–766.

Ramamurthy, Sasthakumar and Raj Bhatnagar (2007). "Tracking recurrent concept drift in streaming data using ensemble classifiers". In: *Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on*. IEEE, pp. 404–409.

Rao, Yanghui, Qing Li, Xudong Mao, and Liu Wenyin (2014). "Sentiment topic models for social emotion mining". In: *Information Sciences* 266, pp. 90–100.

Rastin, Parisa, Guénaël Cabanes, Nistor Grozavu, and Younes Bennani (2015). "Collaborative clustering: How to select the optimal collaborators?" In: *Computational Intelligence, 2015 IEEE Symposium Series on*. IEEE, pp. 787–794.

Read, Jesse, Albert Bifet, Bernhard Pfahringer, and Geoff Holmes (2012). "Batch-incremental versus instance-incremental learning in dynamic and evolving data". In: *International Symposium on Intelligent Data Analysis*. Springer, pp. 313–323.

Rissanen, Jorma (1978). "Modeling by shortest data description". In: *Automatica* 14.5, pp. 465–471.

— (1989). *Stochastic complexity in statistical inquiry*. World Scientific.

Rosen-Zvi, Michal, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth (2004). "The author-topic model for authors and documents". In: *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, pp. 487–494.

Rosenstein, Michael T., Zvika Marx, Leslie Pack Kaelbling, and Thomas G. Dietterich (2005). "To transfer or not to transfer". In: *In NIPS'05 Workshop, Inductive Transfer: 10 Years Later*.

Rougier, S. and A. Puissant (2014). "Improvements of urban vegetation segmentation and classification using multi-temporal Pleiades images". In: *5th International Conference on Geographic Object-Based Image Analysis*, p. 6.

Rousseeuw, Peter J and Annick M Leroy (2005). *Robust regression and outlier detection*. Vol. 589. John wiley & sons.

Rumelhart, David E and Adele A Abrahamson (1973). "A model for analogical reasoning". In: *Cognitive Psychology* 5.1, pp. 1 –28. ISSN: 0010-0285. DOI: https://doi.org/10.1016/0010-0285(73)90023-6. URL: http://www.sciencedirect.com/science/article/pii/0010028573900236.

Russakovsky, Olga et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

Rutkowski, Leszek, Lena Pietruczuk, Piotr Duda, and Maciej Jaworski (2013). "Decision trees for mining data streams based on the McDiarmid's bound". In: *IEEE Transactions on Knowledge and Data Engineering* 25.6, pp. 1272–1279.

Sakaguchi, Takatoshi et al. (2011). "Recommendation system with multi-dimensional and parallel-case four-term analogy". In: *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*. IEEE, pp. 3137–3143.

Salganicoff, Marcos (1997). "Tolerating concept and sampling shift in lazy learning using prediction error context switching". In: *Lazy learning*. Springer, pp. 133–155.

Scheepers, Christoph, Patrick Sturt, Catherine J Martin, Andriy Myachykov, Kay Teevan, and Izabela Viskupova (2011). "Structural priming across cognitive domains: From simple arithmetic to relative-clause attachment". In: *Psychological Science* 22.10, pp. 1319–1326.

Schein, Andrew I, Alexandrin Popescul, Lyle H Ungar, and David M Pennock (2002). "Methods and metrics for cold-start recommendations". In: *Proceedings of the 25th*

annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 253–260.

Schuldt, Christian, Ivan Laptev, and Barbara Caputo (2004). "Recognizing human actions: a local SVM approach". In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 3. IEEE, pp. 32–36.

Sebastião, Raquel and João Gama (2007). "Change detection in learning histograms from data streams". In: *Portuguese Conference on Artificial Intelligence*. Springer, pp. 112–123.

Shachter, Ross D (1998). "Bayes-ball: Rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams)". In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 480–487.

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shao, Ling, Fan Zhu, and Xuelong Li (2015). "Transfer learning for visual categorization: A survey". In: *IEEE transactions on neural networks and learning systems* 26.5, pp. 1019–1034.

Shi, Jianbo and Jitendra Malik (2000). "Normalized cuts and image segmentation". In: *IEEE Transactions on pattern analysis and machine intelligence* 22.8, pp. 888–905.

Shimodaira, Hidetoshi (2000). "Improving predictive inference under covariate shift by weighting the log-likelihood function". In: *Journal of statistical planning and inference* 90.2, pp. 227–244.

Siddiqui, Zaigham Faraz, Eleftherios Tiakas, Panagiotis Symeonidis, Myra Spiliopoulou, and Yannis Manolopoulos (2014). "xstreams: Recommending items to users with time-evolving preferences". In: *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*. ACM, p. 22.

Silver, David et al. (2016). "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587, pp. 484–489.

Skovgaard, Lene Theil (1984). "A Riemannian geometry of the multivariate normal model". In: *Scandinavian Journal of Statistics*, pp. 211–223.

Smeaton, Alan F, Wessel Kraaij, and Paul Over (2004). "The TREC video retrieval evaluation (TRECVID): A case study and status report". In: *Coupling approaches, coupling media and coupling languages for information retrieval*. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, pp. 25–37.

Solomonoff, Ray (1978). "Complexity-based induction systems: comparisons and convergence theorems". In: *IEEE transactions on Information Theory* 24.4, pp. 422–432.

Solomonoff, Ray J (1964). "A formal theory of inductive inference. Part I". In: *Information and control* 7.1, pp. 1–22.

Stavy, Ruth (2012). *U-shaped behavioral growth*. Elsevier.

Steen, Gerard (2008). "The paradox of metaphor: Why we need a three-dimensional model of metaphor". In: *Metaphor and Symbol* 23.4, pp. 213–241.

Strannegård, Claes, Abdul Rahim Nizamani, Anders Sjöberg, and Fredrik Engström (2013). "Bounded Kolmogorov Complexity Based on Cognitive Models". In: *Artificial General Intelligence: 6th International Conference, AGI 2013, Beijing, China, July 31 – August 3, 2013 Proceedings*. Ed. by Kai-Uwe Kühnberger, Sebastian Rudolph, and Pei Wang. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 130–139. ISBN: 978-3-642-39521-5.

Street, W. Nick and YongSeog Kim (2001). "A Streaming Ensemble Algorithm (SEA) for Large-scale Classification". In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '01. San Francisco, California: ACM, pp. 377–382.

Strehl, Alexander and Joydeep Ghosh (2002). "Cluster ensembles—a knowledge reuse framework for combining multiple partitions". In: *Journal of machine learning research* 3.Dec, pp. 583–617.

Stroppa, Nicolas and Francçois Yvon (2005). "An analogical learner for morphological analysis". In: *Proceedings of the Ninth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pp. 120–127.

Sublime, Jérémie, Basarab Matei, Guénael Cabanes, Nistor Grozavu, Younès Bennani, and Antoine Cornuéjols (2017). "Entropy Based Probabilistic Collaborative Clustering". In: *Pattern Recognition* 72, pp. 144–157.

Sublime, Jérémie, Basarab Matei, and Pierre-Alexandre Murena (2017). "Analysis of the influence of diversity in collaborative and multi-view clustering". In: *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, pp. 4126–4133.

Sucar, Luis Enrique (2015). "Probabilistic Graphical Models". In: *Advances in Computer Vision and Pattern Recognition. London: Springer London. doi* 10, pp. 978–1.

Sugiyama, Masashi, Neil D Lawrence, Anton Schwaighofer, et al. (2017). *Dataset shift in machine learning*. The MIT Press.

Sugiyama, Masashi, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe (2008). "Direct importance estimation with model selection and its application to covariate shift adaptation". In: *Advances in neural information processing systems*, pp. 1433–1440.

Surowiecki, James (2004). "The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business". In: *Economies, Societies and Nations* 296.

Tenenbaum, Joshua B, Charles Kemp, Thomas L Griffiths, and Noah D Goodman (2011). "How to grow a mind: Statistics, structure, and abstraction". In: *science* 331.6022, pp. 1279–1285.

Thrun, Sebastian and Lorien Pratt, eds. (1998). *Learning to Learn*. Norwell, MA, USA: Kluwer Academic Publishers. ISBN: 0-7923-8047-9.

Tsymbal, Alexey (2004). "The problem of concept drift: definitions and related work". In: *Computer Science Department, Trinity College Dublin* 106.2.

Tsymbal, Alexey, Mykola Pechenizkiy, Padraig Cunningham, and Seppo Puuronen (2006). "Handling local concept drift with dynamic integration of classifiers: Domain of antibiotic resistance in nosocomial infections". In: *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*. IEEE, pp. 679–684.

Tsymbal, Alexey, Mykola Pechenizkiy, Pádraig Cunningham, and Seppo Puuronen (2008). "Dynamic integration of classifiers for handling concept drift". In: *Information fusion* 9.1, pp. 56–68.

Turney, Peter D (2008). "The latent relation mapping engine: Algorithm and experiments". In: *Journal of Artificial Intelligence Research* 33, pp. 615–655.

Ullman, Michael T (2001). "The neural basis of lexicon and grammar in first and second language: The declarative/procedural model". In: *Bilingualism: Language and cognition* 4.2, pp. 105–122.

Vanhaesebrouck, Paul, Aurélien Bellet, and Marc Tommasi (2017). "Decentralized collaborative learning of personalized models over networks". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Vapnik, Vladimir (2006). "Transductive Inference and Semi-Supervised Learning". In: *Semi-supervised learning*. Ed. by Olivier Chapelle, Bernard Schölkopf, and Alexander Zien. MIT Press. Chap. 24, pp. 454–472.

Vapnik, Vladimir and Rauf Izmailov (2015). "Learning Using Privileged Information: Similarity Control and Knowledge Transfer". In: *Journal of Machine Learning Research* 16, pp. 2023–2049. URL: http://jmlr.org/papers/v16/vapnik15b.html.

Vapnik, Vladimir N. (1995). *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc. ISBN: 0-387-94559-8.

Vega-Pons, Sandro and José Ruiz-Shulcloper (2011). "A survey of clustering ensemble algorithms". In: *International Journal of Pattern Recognition and Artificial Intelligence* 25.03, pp. 337–372.

Vendetti, Michael S, Ariel Starr, Elizabeth L Johnson, Kiana Modavi, and Silvia A Bunge (2017). "Eye movements reveal optimal strategies for analogical reasoning". In: *Frontiers in psychology* 8, p. 932.

Vinagre, João, Alípio Mário Jorge, and João Gama (2014a). "Evaluation of recommender systems in streaming environments". In: *Proceedings of the Workshop on Recommender Systems Evaluation: Dimensions and Design in conjunction with the 8th ACM Conference on Recommender Systems (RecSys 2014)*.

— (2014b). "Fast incremental matrix factorization for recommendation with positive-only feedback". In: *International Conference on User Modeling, Adaptation, and Personalization*. Springer, pp. 459–470.

Vorburger, Peter and Abraham Bernstein (2006). "Entropy-based concept shift detection". In: *Data Mining, 2006. ICDM'06. Sixth International Conference on*. IEEE, pp. 1113–1118.

Wald, Abraham (1973). *Sequential analysis*. Courier Corporation.

Wallace, Christopher S and David M Boulton (1968). "An information measure for classification". In: *The Computer Journal* 11.2, pp. 185–194.

Wang, Chong and David M Blei (2011). "Collaborative topic modeling for recommending scientific articles". In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 448–456.

Wang, Hao, Naiyan Wang, and Dit-Yan Yeung (2015). "Collaborative deep learning for recommender systems". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 1235–1244.

Wang, Hua, Feiping Nie, and Heng Huang (2013). "Robust and discriminative self-taught learning". In: *International conference on machine learning*, pp. 298–306.

Wang, Hua-Yan and Qiang Yang (2011). "Transfer Learning by Structural Analogy." In: *AAAI*.

Wang, Xuerui and Andrew McCallum (2006). "Topics over time: a non-Markov continuous-time model of topical trends". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 424–433.

Webb, Geoffrey I, Roy Hyde, Hong Cao, Hai Long Nguyen, and Francois Petitjean (2016). "Characterizing concept drift". In: *Data Mining and Knowledge Discovery* 30.4, pp. 964–994.

Weiner, E Judith and William Labov (1983). "Constraints on the agentless passive". In: *Journal of linguistics* 19.1, pp. 29–58.

Weinland, Daniel, Remi Ronfard, and Edmond Boyer (2006). "Free viewpoint action recognition using motion history volumes". In: *Computer vision and image understanding* 104.2-3, pp. 249–257.

Weiss, Karl, Taghi M Khoshgoftaar, and DingDing Wang (2016). "A survey of transfer learning". In: *Journal of Big Data* 3.1, p. 9.

Wemmert, Cédric and Pierre Gançarski (2002). "A multi-view voting method to combine unsupervised classifications". In: *Artificial Intelligence and Applications*, pp. 447–452.

Wheeler, Donald J, David Smith Chambers, et al. (1992). *Understanding statistical process control*. SPC press.

Widmer, Gerhard and Miroslav Kubat (1996). "Learning in the presence of concept drift and hidden contexts". In: *Machine learning* 23.1, pp. 69–101.

Widyantoro, Dwi H, Thomas R Ioerger, and John Yen (2003). "Tracking changes in user interests with a few relevance judgments". In: *Proceedings of the twelfth international conference on Information and knowledge management*. ACM, pp. 548–551.

Winston, Patrick H (1977). "Learning by creating and justifying transfer frames". In:

— (1980). "Learning and reasoning by analogy". In: *Communications of the ACM* 23.12, pp. 689–703.

Wolpert, David H. (Oct. 1996). "The Lack of a Priori Distinctions Between Learning Algorithms". In: *Neural Comput.* 8.7, pp. 1341–1390. ISSN: 0899-7667. DOI: 10.1162/neco.1996.8.7.1341. URL: http://dx.doi.org/10.1162/neco.1996.8.7.1341.

Wolpert, David H (1997). "On bias plus variance". In: *Neural Computation* 9.6, pp. 1211–1243.

— (2002). "The supervised learning no-free-lunch theorems". In: *Soft computing and industry*. Springer, pp. 25–42.

Xu, Rui and Donald Wunsch (2005). "Survey of clustering algorithms". In: *IEEE Transactions on neural networks* 16.3, pp. 645–678.

Yang, Ying, Xindong Wu, and Xingquan Zhu (2006). "Mining in anticipation for concept change: Proactive-reactive prediction in data streams". In: *Data mining and knowledge discovery* 13.3, pp. 261–289.

Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson (2014). "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems*, pp. 3320–3328.

Yvon, François (2003). "Finite-state machines solving analogies on words". In: *Rapport technique D* 8.

Zhang, Chao, Lei Zhang, and Jieping Ye (2012). "Generalization bounds for domain adaptation". In: *Advances in neural information processing systems*, pp. 3320–3328.

Zhao, Weizhong, Huifang Ma, and Qing He (2009). "Parallel k-means clustering based on mapreduce". In: *IEEE International Conference on Cloud Computing*. Springer, pp. 674–679.

Zimek, Arthur and Jilles Vreeken (2015). "The blind men and the elephant: on meeting the problem of multiple truths in data from clustering and pattern mining perspectives". In: *Machine Learning* 98.1-2, pp. 121–155.

Žliobaitė, Indrė, Mykola Pechenizkiy, and Joao Gama (2016). "An overview of concept drift applications". In: *Big data analysis: new algorithms for a new society*. Springer, pp. 91–114.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Principe de minimum de complexité pour le transfert de connaissances en apprentissage artificiel

Mots clés : Analogie, Transfert, Apprentissage incrémental, Complexité de Kolmogorov

Résumé : Les méthodes classiques d'apprentissage automatique reposent souvent sur une hypothèse simple mais restrictive: les données du passé et du présent sont générées selon une même distribution. Cette hypothèse permet de développer directement des garanties théoriques sur la précision de l'apprentissage. Cependant, elle n'est pas réaliste dans un grand nombre de domaines applicatifs qui ont émergé au cours des dernières années. Dans cette thèse, nous nous intéressons à quatre problèmes différents en intelligence artificielle, unis par un point commun: tous impliquent un transfer de connaissance d'un domaine vers un autre. Le premier problème est le raisonnement par analogie et s'intéresse à des assertions de la forme « A est à B ce que C est à D ». Le second est l'apprentissage par transfert et se concentre sur des problèmes de classification dans des contextes où les données d'entraînement et de test ne sont pas de même distribution (ou n'appartiennent même pas au même espace). Le troisième est l'apprentissage sur flux de données, qui prend en compte des données apparaissant continument une à une à haute fréquence, avec des changements de distribution. Le dernier est le clustering collaboratif et consiste à faire échanger de l'information entre algorithmes de clusterings pour améliorer la qualité de leurs prédictions. La principale contribution de cette thèse est un cadre général pour traiter les problèmes de transfert. Ce cadre s'appuie sur la notion de complexité de Kolmogorov, qui mesure l'information continue dans un objet. Cet outil est particulièrement adapté au problème de transfert, du fait qu'il ne repose pas sur la notion de probabilité tout en étant capable de modéliser les changements de distributions. En plus de cet effort de modélisation, nous proposons dans cette thèse diverses discussions sur d'autres aspects ou applications de ces problèmes. Ces discussions s'articulent autour de la possibilité de transfert dans différents domaines et peuvent s'appuyer sur d'autres outils que la complexité.

Title : Minimum Complexity Principle for Knowledge Transfer in Artificial Learning

Keywords : Analogy, Transfer, Incremental learning, Kolmogorov complexity

Abstract : Classical learning methods are often based on a simple but restrictive assumption: The present and future data are generated according to the same distributions. This hypothesis is particularly convenient when it comes to developing theoretical guarantees that the learning is accurate. However, it is not realistic from the point of view of applicative domains that have emerged in the last years. In this thesis, we focus on four distinct problems in artificial intelligence, that have mainly one common point: All of them imply knowledge transfer from one domain to the other. The first problem is analogical reasoning and concerns statements of the form "A is to B as C is to D". The second one is transfer learning and involves classification problem in situations where the training data and test data do not have the same distribution (nor even belong to the same space). The third one is data stream mining, ie. managing data that arrive one by one in a continuous and high-frequency stream with changes in the distributions. The last one is collaborative and focuses on exchange of information between clustering algorithms to improve the quality of their predictions. The main contribution of this thesis is to present a general framework to deal with these transfer problems. This framework is based on the notion of Kolmogorov complexity, which measures the inner information of an object. This tool is particularly adapted to the problem of transfer, since it does not rely on probability distributions while being able to model the changes in the distributions. Apart from this modeling effort, we propose, in this thesis, various discussions on aspects and applications of the different problems of interest. These discussions all concern the possibility of transfer in multiple domains and are not based on complexity only.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France