Structured machine learning methods for microbiology : mass spectrometry and high-throughput sequencing

Kevin Vervier

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

MINES ParisTech

École doctorale n° 432: Sciences des métiers de l'ingénieur

Doctorat européen ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

l'École nationale supérieure des mines de Paris

Spécialité doctorale "Bio-informatique"

présentée et soutenue publiquement par

Kévin Vervier

le 25 juin 2015

Méthodes d'apprentissage structuré pour la microbiologie: spectrométrie de masse et séquençage haut-débit.

—

Structured machine learning methods for microbiology: mass spectrometry and high-throughput sequencing.

Directeur de thèse : **Jean-Philippe Vert**
Co-encadrant de thèse : **Pierre Mahé**

Jury

Stéphane Canu,	Professeur, INSA de Rouen	Rapporteur
Nicola Segata,	Principal investigator, University of Trento	Rapporteur
Eric Gaussier,	Professeur, Université Joseph Fourier, Grenoble	Examinateur
Stéphane Robin,	Professeur, AgroParisTech	Examinateur
Pierre Mahé,	Ingénieur de recherche, bioMérieux, Grenoble	Examinateur
Jean-Philippe Vert,	Maître de recherche, Centre de Bio-Informatique, Mines ParisTech	Examinateur

MINES ParisTech
Centre de Bio-Informatique (CBIO)
35 rue Saint-Honoré, 77300 Fontainebleau, France

THESE

Acknowledgements/Remerciements

I would like to acknowledge Stéphane Canu and Nicola Segata for accepting to be part of my jury.

I would also like to thank Eric Gaussier and Stéphane Robin for being part of my jury and for fruitful discussions and their comments on my research work during the past three years.

Merci à Jean-Philippe Vert, mon directeur de thèse. Malgré la distance Paris-Lyon, il s'est montré très disponible et à l'écoute.

Merci à Pierre Mahé pour m'avoir encadré au quotidien chez bioMérieux lors de ma thèse. Je ne compterai pas le temps que nous avons pu passer à tordre des problèmes épineux dans tous les sens, ainsi qu'à ecrire des lignes de script ensemble. Il a toujours su trouver les mots justes, pour me relancer lorsque les résultats n'étaient pas très positifs.

Merci à Jean-Baptiste Veyrieras, mon manager à bioMérieux, pour m'avoir accueilli sur les sites de Marcy-l'Etoile et de Grenoble, mais également pour sa disponibilité et son efficacité.

Merci aux membres de l'équipe DKL/BIRD: Audrey, Ghislaine, Magalie, Maud, Nathalie, Bertrand, Christophe, Guillaume, Stéphane, Thomas. Ce que je retiendrai le plus de cette équipe est la pluridisciplinarité de tous ses membres, et des riches discussions qui en découlent. Merci aux biomaths de Grenoble avec qui j'ai partagé d'innombrables pauses-café: Céline, Faustine, Sophie, Véronique, Laurent et Etienne.

Je préfère ne pas faire de liste de peur d'en oublier, mais je remercie également tous les membres du CBIO que j'ai pu croiser lors de mes visites sur Paris. Merci aussi à Emmanuel Barillot de m'avoir accueilli dans les locaux de l'Institut Curie.

Merci à l'ANRT et au dispositif CIFRE qui représente une réelle opportunité de faire de la recherche à la jonction du domaine privé et de l'académique.

En dehors de travail, il y a aussi ceux qui sont là quelque soit le jour, qu'il neige ou qu'il pleuve. Un grand merci aux amis de longue date toujours disponibles pour papoter ou me sortir (je précise que l'ordre ne veut rien dire): Adeline, Camille, Bruno, Cedric, David, Gauthier, Romain, Sebastien, Vincent.

Merci aussi aux deux anciens de classe préparatoire, Clément et Guillaume, qui m'ont accueilli lors de mes séjours sur Paris et qui m'ont fait connaître un peu mieux la capitale.

Merci à ma famille, et en particulier à mes parents, pour m'avoir appuyé depuis toujours dans mes décisions.

Merci enfin à Virginie pour sa patience et son soutien au quotidien durant toutes ces années.

Contents

List of Figures

List of Tables

Abstract

Using high-throughput technologies in Mass Spectrometry (MS) and Next-Generation Sequencing (NGS) is changing scientific practices and landscape in microbiology. On the one hand, mass spectrometry is already used in clinical microbiology laboratories through systems identifying unknown microorganisms from spectral data. On the other hand, the dramatic progresses during the last 10 years in sequencing technologies allow cheap and fast characterizations of microbial diversity in complex clinical samples, an approach known as "metagenomics". Consequently, the two technologies will play an increasing role in future diagnostic solutions not only to detect pathogens in clinical samples but also to identify virulence and antibiotic resistance.

This thesis focuses on the computational aspects of this revolution and aims to contribute to the development of new *in vitro* diagnostics (IVD) systems based on high-throughput technologies, like mass spectrometry or next generation sequencing, and their applications in microbiology. To deal with the volume and complexity of data generated by these new technologies, we develop innovative and versatile statistical learning methods for applications in IVD and microbiology. The field of statistical learning is indeed well-suited to solve tasks relying on high-dimensional raw data that can hardly be manipulated by medical experts, like identifying an organism from an MS spectrum or affecting millions of sequencing reads to the right organism.

Our main methodological contribution is to develop and evaluate statistical learning methods that incorporate prior knowledge about the structure of the data or of the problem to be solved. For instance, we convert a sequencing read (raw data) into a vector in a nucleotide composition space and use it as a *structured input* for machine learning approaches. We also add prior information related to the hierarchical structure that organizes the reachable microorganisms (*structured output*).

Résumé

L'utilisation des technologies haut débit de spectrométrie de masse et de séquençage nouvelle génération est en train de changer aussi bien les pratiques que le paysage scientifique en microbiologie. D'une part la spectrométrie de masse a d'ores et déjà fait son entrée avec succès dans les laboratoires de microbiologie clinique au travers de systèmes permettant d'identifier un microorganisme à partir de son spectre de masse. D'autre part, l'avancée spectaculaire des technologies de séquençage au cours des dix dernières années permet désormais à moindre coût et dans un temps raisonnable de caractériser à la fois qualitativement et quantitativement la diversité microbienne au sein d'échantillons cliniques complexes (approche désormais communément dénommée métagenomique). Aussi ces deux technologies sont pressenties comme les piliers de futures solutions de diagnostic permettant de caractériser simultanément et rapidement non seulement les pathogènes présents dans un échantillon mais également leurs facteurs de résistance aux antibiotiques ainsi que de virulence.

Cette thèse vise donc à contribuer au développement de nouveaux systèmes de diagnostic *in vitro* basés sur les technologies haut débit de spectrométrie de masse et de séquençage nouvelle génération pour des applications en microbiologie.

L'objectif de cette thèse est de développer des méthodes d'apprentissage statistique innovantes et versatiles pour exploiter les données fournies par ces technologies haut-débit dans le domaine du diagnostic *in vitro* en microbiologie. Le domaine de l'apprentissage statistique fait partie intégrante des problématiques mentionnées ci-dessus, au travers notamment des questions de classification d'un spectre de masse ou d'un "read" de séquençage haut-débit dans une taxonomie bactérienne.

Sur le plan méthodologique, ces données nécessitent des développements spécifiques afin de tirer au mieux avantage de leur structuration inhérente: une structuration en "entrée" lorsque l'on réalise une prédiction à partir d'un "read" de séquençage caractérisé par sa composition en nucléotides, et un structuration en "sortie" lorsque l'on veut associer un spectre de masse ou d'un "read" de séquençage à une structure hiérarchique de taxonomie bactérienne.

Chapter 1

Introduction

In this chapter, we provide an overall background and technical notations related to the main concepts studied in this thesis, namely *microbiology* and *supervised learning*.

Microbiology is the study of all microorganisms and so includes disciplines like bacteriology, mycology or virology. In medical microbiology, the diagnosis of infectious diseases relies on the study of pathogens characteristics. The identification of the infectious agent may involve more than a standard physical examination. For instance, sophisticated techniques (such as polymerase chain reaction [173] or mass-spectrometry [144]) are used to detect abnormalities induced by the presence of a pathogen agent, at a molecular level.

Supervised learning allows to infer a rule between features/measurements and an outcome of interest. This rule is inferred using noisy input observations, called *training examples*, for which we also know the corresponding output *response* variable. Once the rule is inferred, it can be used to predict the output of any new input data. For instance, automatic microbial identification based on high-throughput data aims at linking large amount of microorganisms characteristics to their identity, and can be cast as a learning problem. Detailed introduction to supervised learning are given in, e.g, [70, 170].

This chapter is organized as follows. Section 1.1 is related to microbiology and *in vitro* diagnostics applications. Section 1.2 provides an overview on supervised learning and more precisely on *classification* in Section 1.3. In Section 1.4, we detail how we correctly compare and evaluate the different methods. Finally, Section 1.5 provides a presentation of the contributions of this thesis.

1.1 Microbiology and *in vitro* diagnostics

In vitro diagnostics (IVD) tests are comprised of reagents, instruments and systems used to analyze the content of biological samples of interest in the process of a medical diagnosis. For instance, IVD tests are commonly used in a clinical context for measuring base compounds in the body, indicating the presence of biological markers (HIV,

tumor) or detecting disease-causing agents. The same tests are also used for industrial purposes, like food safety or sterility testing, in agri-food, cosmetic, pharmaceutic industries. However, the focus of our research work is restricted to clinical applications. More than 70% of the US medical decisions draw upon the results of an IVD test [66], yet only 2% of the US$2 trillion spent annually on healthcare goes to diagnostics [65]. According to [182], there exists more than 4,000 different tests available for clinical use and about 7 billions of IVD tests are performed each year. They are critical for medical decision-making, allowing to identify infections and diseases, and have a huge impact in terms of lives saved and reduced health care budget thanks to low-cost devices for the costly diseases, like cardiovascular diseases, cancer or infectious diseases (HIV, tuberculosis, influenza, etc.)

During my thesis, I spent most of my time working for bioMérieux, the world leader in IVD tests for microbiology. IVD companies, like bioMérieux, are massively investing to develop the future diagnostics solutions based on high-throughput technologies.

1.1.1 Diagnostics for infectious diseases

Infectious diseases are caused by pathogenic microorganisms, such as bacteria, viruses, fungi or parasites. Microbial identification, a problematic in IVD, aims at identifying the microorganism causing the disease from clinical samples such as blood, urine or saliva. Identifying a pathogen is a crucial step in the diagnostics workflow and there is still a need for faster and more reliable tests to help clinicians prescribe an appropriate treatment. The other crucial step in diagnostics for infectious diseases is the antibiotic susceptibility testing (AST) [76]. The main goal of AST is to determine which antibiotic treatment will be most successful *in vivo*. The combination of a correct identification of the pathogen agent and its relevant antibiotic sensitivity represents the ideal clinical therapy. Here, the focus of this thesis is on the development of innovative strategies for microbial identification.

Most existing microbial identification technologies require a culture step. Most bacteria will grow overnight, whereas some mycobacteria require as many as 6 to 8 weeks [13]. Microorganisms present in the sample are isolated on a culture medium that recreates favorable growth conditions. After a few hours, colonies will appear on the medium; each colony only contains replicates of an initially isolated microbe. Thus, this culture step acts like a signal amplification step, multiplying microbiological material in the sample in order to collect enough material from each colony to perform the identification step.

A great variety of identification systems have been proposed, which are sometimes categorized into phenotypic, genotypic and proteotypic methods [160]. Phenotypic methods like, for instance, active pharmaceutical ingredient (API) [4], typically base their identification on the results of several chemical reactions revealing metabolic characteristics of the microorganism.

Genotypic identification relies on the genetic material (DNA, RNA) of the microorganism and involves sequencing a specific genetic marker, like the 16S rRNA which is often considered as the gold-standard for bacterial identification [46]. These technologies have been commonly used in research laboratories for decades, yet they only started to change the IVD market in the 1990s due to the need for validations [23] by agencies like the US Food and Drug Administration (FDA). The adoption of those complex tests also requires specific training for clinicians and healthcare providers, in particular to interpret the results. More recently, so-called proteotypic methods have been introduced as well. These methods base their identification on measurements of the cell content. They include for instance RAMAN spectroscopy [36], fluorescence spectroscopy [21] and Matrix-Assisted Laser Desorption Ionization Time of Flight mass-spectrometry (MALDI-TOF MS) [47].

1.1.2 A new paradigm in microbial identification: high-throughput technologies

Mass-spectrometry for microbial identification

Until recently, clinical microbiology has mainly relied on conventional phenotypic and biochemical techniques [167]. After a traditional culture step, standardized test systems such as the Phoenix® system (Becton, Dickinson Diagnostics, Sparks, MD), Microscan Walkaway® (Dade-Behring MicroScan, Sacramento, CA), API® and VITEK® 2 (bioMérieux, Marcy l'Etoile, France), have so far been used to speed up microbial identification: the average time needed for a reliable identification ranges from 6h to 18h [60]. In the last few years, PCR methods have complemented the biochemical approaches, decreasing time-to-results with no mandatory culture step and even becoming in some cases the reference method [175]. However, PCR methods rely on the design of new oligonucleotide primers requiring the analysis of the genome of each clinically relevant microorganism. The efficiency of such approaches can also be reduced by unexpected mutations or unknown variants. Additionally, PCR sensitivity can be too high for some applications, detecting a microbe that is present at non-pathogenic levels [106].

Even if it requires a culture step, MALDI-TOF MS can identify the genus and species of a microorganism after a rapid (few minutes) and simple MS experiment. It is now broadly accepted by the clinical microbiology community as a routine testing tool for microbial identification at the level of species [53, 96, 168].

Generally, MS technologies rely on an instrument that takes a sample, ionizes it, for example by bombarding with a laser energy, and converts electric signals to intensity peaks [155]. MALDI-TOF is often referred to as a soft-ionization technology, because of the small amount of energy used, and low risk of bound ruptures. Indeed, fragile biomolecules such as proteins are protected by matrix crystallized molecules from a direct ionization source. Charged particles created by ionization pass through a vacuum

tube playing the role of analyzer. Ions are simply discriminated according to their mass-to-charge ratio (m/z), with a shorter time-of-flight for the smaller ions. At the end of the analyzer, a detector measures, at each impact, the intensity and the mass of the charged particles, leading to an intensity peak profile, also called a mass-spectrum. The whole process is illustrated in Figure 1.1.

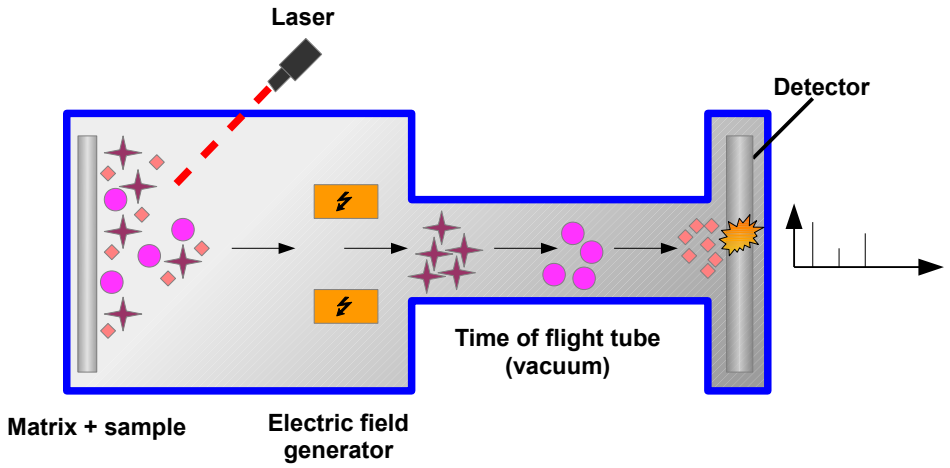

Figure 1.1: **MALDI-TOF mass-spectrometry.** The sample is mixed with a matrix that protects fragile biomolecules. A ionization source, like a laser, is used to pulverized the mixture. Then, ions pass through a vaccum tube where small ions are faster than heavy ones. At the end of the time of flight tube, a detector measures impacts intensity and returns a mass-spectrum profile.

The output of an MS experiment can be represented by a large number of parameters, like m/z peak positions and their associated intensities. In diagnostic systems based on mass spectra in microbiology, like the Biotyper (Bruker Daltonics, Germany), or VITEK®-MS, the interpretation of raw data is not performed by clinicians, but relies on mathematical algorithms. The major commercially available mathematical systems are the Bruker Main Spectrum analysis (MSP) and the bioMérieux SuperSpectrum and Advanced Spectra Classifier (ASC). For those algorithms, the extraction of the intensity peaks and the comparisons of spectra are entirely automated [43]. Each measured spectrum is compared to the spectra in a reference database, and the system converts a similarity score into one or more microbial species names. Indeed, proteins contained in a microorganism are peculiar to their biological species and can be used to identify a microbe observed in a biological sample.

Despite differences in the reference databases, both systems address the large majority of clinically relevant species found in routine clinical practice, including the 20 bacteria that represent $> 80\%$ of isolates recovered from human clinical samples. Compared to conventional biochemical tests, MALDI-TOF achieves comparable identifica-

tion results for a vast majority ($\sim$ 95%) of the isolates and in case of discordance between the two approaches, 16S rRNA sequencing confirmed MALDI-TOF in 63% of discordant cases [17].

Next-generation sequencing and metagenomics

The study of metagenomes, also called *metagenomics*, consists in analyzing genetic material recovered from environmental samples. While in traditional microbiology, genome sequencing and genomics rely upon cultivated cultures, metagenomics does not require pure clonal cultures of individual organisms. From an environmental sample, one can estimate its microbial diversity using conserved and universally present markers such as ribosomal RNA [181] and clones specific genes (mostly the bacterial 16S rRNA gene, but 25-30 highly conserved genes are listed in [45]). Such *targeted* approaches revealed that the majority of microbial biodiversity had been missed by cultivation-based methods [78]. It is estimated that more than 99% of microorganisms observable in nature typically are not cultivable by using standard culture techniques [6, 126].

Recent advances in genome sequencing technologies and metagenome analysis provide a broader understanding of microbes and highlight differences between healthy and disease states. Metagenome studies have recently increased in number and scope due to the rapid advancements of high-throughput Next Generation Sequencing (NGS) technologies such as GS FLX system from 454 Life Sciences, a subsidiary of Roche [112], the IonTorrent's Personal Genome Machine (PGM) [142], the Illumna MiSeq and HiSeq [19], and the Pacific Biosciences RS (PacBio) [59]. These modern sequencing technologies give us access to a new way of analyzing clinical samples, because they are culture-independent and randomly sequence all microorganisms present in an environment.

1.1.3 Hierarchical organization of microorganisms

Microorganisms are diverse, but share biological properties and evolutionary history. It is therefore useful to group them in a hierarchical structure, called a *taxonomy*, to organize this diversity. A taxonomic tree is a rooted structure that links groups (taxa) from top and bottom according to general properties (top) to specific properties (bottom): two children taxa sharing the same parent taxon have common features contained in the parent taxon while each sibling taxon has specific characteristics, unshared with its siblings. The taxonomy concept is not proper to the biology field (e.g., semantic Web), but plays an important role in life sciences knowledge organization. Numerous phylogenetic taxonomies are available online, such as NCBI taxonomy [178], UniProt knowledge base [107]. In the *in vitro* diagnostics context, it is possible to design a polyphasic taxonomic tree as a combination of phylogenetic and phenotypic levels representing successive decision rules used in medical and clinical analysis, as defined for instance in Bergey's Manual of Bacteriology [74] and shown in Figure 1.2. According

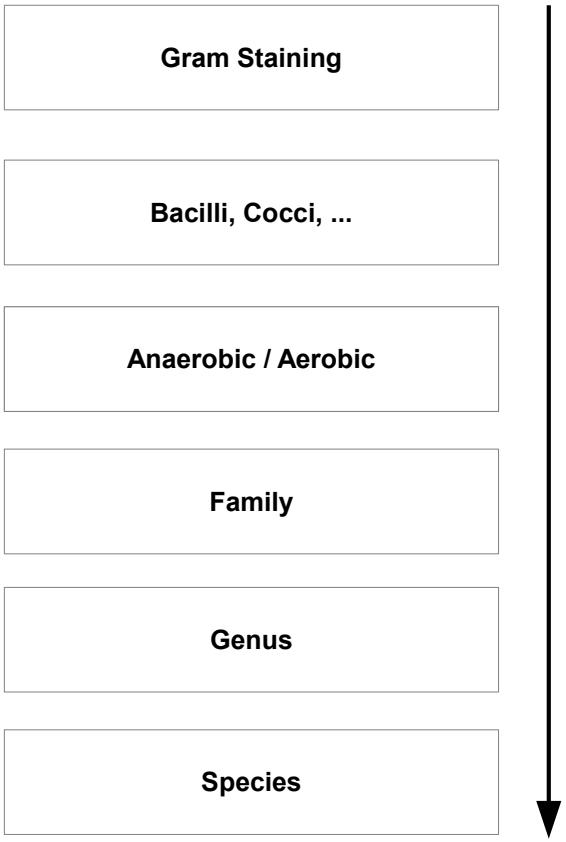

Figure 1.2: **Example of a polyphasic taxonomy.** Top levels (e.g., Gram staining) correspond to phenotypic classification tests and low levels support phylogenetic information.

to the classification proposed by Carl von Linné [104], the finest and lowest level of the taxonomic tree is the *species* level. The higher ranks are called, from the lowest to the most generic: Genus, Family, Order, Class, Phylum, Kingdom, Domain, as summarized in Figure 1.3. To underline the general character of the higher taxonomic levels, note that there are only three known (and accepted) domains that are Archea, Bacteria and Eukaryota. The last one regroups all organisms made of cells with a genetic material enclosed by a nuclear envelope, including all animals, plants and fungi.

In microbiology, one often considers a level below the species level, called the *strain* level. A microbial strain is a particular member of a species that differs from the other members by a minor but significant variation [179]. These genetic variations may have a large impact on the expressed characteristics, or *phenotypes* of the microorganism. For instance, the microbial species *Escherichia coli* is the most abundant commensal bacteria in the human gastrointestinal track [89] and it coexists with the host with mutual benefit, such as the use of gluconate permease in the colon [161]. However,

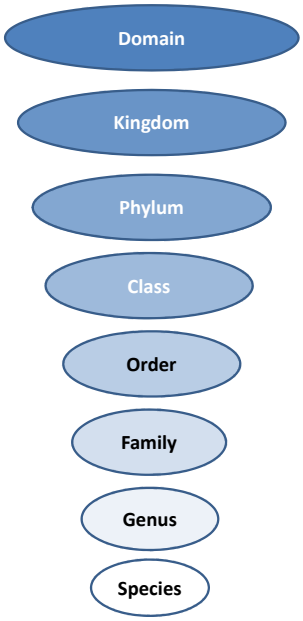

Figure 1.3: **Taxonomic structure of the Tree of Life.** Top levels are the most generic and the bottom levels are the finest.

several *E.coli* strains have acquired virulence mechanisms allowing a rapid colonization of new environments and causing deadly syndromes, like bloody diarrhea in the recent 2011 *E.coli* O104:H4 outbreak in Europe [120]. Hence, strains from the same species can have different biological properties related to virulence or drug resistance.

The concept of bacterial species is slightly different from the numerous eukaryota definitions [141], like interbreeding population [115] or other sexual characterizations. The current gold-standard criterion proposed in [122] measures a cross-hybridization proportion between two DNA strands coming from different organisms. The DNA-DNA hybridization (DDH) threshold for considering that two organisms belong to the same species is at least 70%. Because cross-hybridization experiments are not easily applicable to all the bacterial environments, alternative approach based on a conserved gene marker, 16S rRNA, has been proposed in [157]. Results in [93] suggest that a 97% 16S rRNA gene sequence identity is easier to measure with DNA sequencing technologies and is equivalent to the previous 70% DDH threshold. The classification of species has been affected by the gold-standard changes and the technological revolutions: from the precipitation assays for blood plasma in 1950's, to the DDH [150] in the 1970's, to the recent DNA sequencing. Reorganizations in the phylogenetic taxonomy have been induced at all the classification levels and this tree is still being modified by taxonomists, based on recent findings.

Another issue with the taxonomic organization of the species is the problem of *taxa in disguise* [87]. They are taxonomic units that have evolved from another unit of similar rank making the parent unit paraphyletic. It means that phylogenetically, all descendants of the parent unit are identical from an evolution point of view but not taxonomically. In general, this paraphyly can be solved by moving the taxon in disguise

under the parent unit. However, in microbiology, reorganizing and renaming taxonomic units may induce confusion over the identity of microorganisms with a medical impact, like pathogens. For instance, the *Shigella* genus is an "*E.coli* in disguise" [97] that develops a characteristic form of pathogenesis residing on the *pInv* plasmid [67]. *Shigella* members are the cause of a severe infectious disease killing hundreds of thousands people each year: the bacillary dysentery or shigellosis [159]. Because *E.coli* can also cause similar symptoms, the current taxonomic classification will not change to avoid confusion in a medical context. Another example is the species belonging to *Bacillus cereus* group. They present 16S rRNA sequence similarities around 99-100% [15], higher than the previously described 97% threshold and should be regrouped in a single taxonomic entity. For medical reasons including the pathogenicity of some members, like *Bacillus anthracis*, responsible for anthrax disease, they will not be merged in the taxonomic tree.

1.2 Supervised learning

Generally speaking, the field of machine learning can be defined as the construction of powerful informatics systems that can learn from observations and measures, instead of following a list of instructions.

In this thesis, we focus on methods for *supervised learning*. Supervised learning consists in the estimation of a rule between some input and output data. This rule is inferred using noisy observations, called *training examples* for which we also know the corresponding *response* variable. The objective is to infer the unknown relation from training data and then, use this rule to correctly *predict* the output of any new input data. Depending on the nature of the response, one can distinguish two main classes of problem. If the output variable is discrete and represents categories, the problem is called *classification*, while it is called *regression* when the response is a continuous real number.

This section first provides general background and notations for supervised learning. Then, we discuss an important concept in learning, the trade-off between approximation and estimation. In the last part of this section, we introduce the *regularization* concept for learning model under constraints.

1.2.1 Supervised learning: notations

In supervised learning, our goal is to make a model to predict an output Y in an output space $\mathcal{Y}$ given an input X in an input space $\mathcal{X}$. The output variable is also often called the *response* variable. For the output space, we typically take $\mathcal{Y} = \mathbb{R}$ for regression tasks, or $\mathcal{Y} = \{-1; +1\}$ for binary classification problems. Regarding the input, we will restrict ourselves to data represented by p numerical descriptors or *features*, hence consider an input space $\mathcal{X} \subseteq \mathbb{R}^p$. The model is learned from the observation of a set of

n input-output pairs, $(x_i, y_i)_{i=1,\dots,n} \in (\mathcal{X} \times \mathcal{Y})^n$ called the *training set*. For clarity, it is convenient to merge the observed inputs into a n-by-p matrix $\mathbf{X} = (x_{i,j})_{i=1,\dots n; j=1,\dots,p}$, where each row is an input and each column corresponds to a feature. Similarly, we merge the outputs into a n-dimensional vector $\mathbf{Y} = (y_i)_{i=1,\dots,n} \in \mathcal{Y}^n$.

To model the input-output relationship, it is standard in statistical learning to assume that the training examples $(x_i, y_i)_{i=1,\dots,n}$ are realizations of random variables $(X_i, Y_i)_{i=1,\dots,n}$ independent and identically distributed (i.i.d.) according to an unknown joint distribution $\mathbb{P}(X, Y)$ on $\mathcal{X} \times \mathcal{Y}$, and that future observation will also be realizations of independent random variables distributed according to $\mathbb{P}$. Note that this unknown distribution can be written as the product of the marginal distribution $\mathbb{P}(X)$, which describes how the inputs are distributed, and of the conditional distribution $\mathbb{P}(Y|X)$, which describes how an output is related to an input.

Given a function $f : \mathcal{X} \to \mathcal{Y}$ that deterministically predicts an output $f(x) \in \mathcal{Y}$ for any input $x \in \mathcal{X}$, we would like to measure its quality by how "well" it predicts the response variable on unseen examples. For that purpose, it is useful to introduce a *loss function* $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ to measure the disagreement $l(\hat{y}, y)$ between a predicted response $\hat{y}$ and a true response y, small loss values corresponding to good predictions. We will discuss in more details standard loss function in Section 1.2.3. Given a loss function, the *risk* of a predictor $f : \mathcal{X} \to \mathcal{Y}$ can now be defined as the expected loss it will incur on unseen examples, namely

$$R(f) = \int l(f(X), Y) d\mathbb{P}(X, Y)\,. \tag{1.1}$$

The goal of statistical learning can then be summarized as the task of using the training set to estimate a predictor $\hat{f} : \mathcal{X} \times \mathcal{Y}$ with the smallest possible risk $R(\hat{f})$.

1.2.2 Empirical risk minimization, approximation and estimation errors

Ideally, the goal of statistical learning is therefore to find the predictor f^* that minimizes the risk $R(f)$ over all possible measurable functions $f : \mathcal{X} \to \mathcal{Y}$, by solving the risk functional minimization problem [169]

$$f^* = \arg\min_f R(f). \tag{1.2}$$

Unfortunately, since the joint probability $\mathbb{P}(X, Y)$ is unknown, the risk $R(f)$ is not computable and f^* is not reachable. Instead of $R(f)$, what we can compute from the training data is the *empirical risk*:

$$R_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^{n} l(f(x_i), y_i)\,, \tag{1.3}$$

which for each f is an unbiased estimate of $R(f)$. To estimate a predictor $\hat{f}$ from the training data, the *empirical risk minimization (ERM)* estimator is the predictor that minimizes the empirical risk over a pre-defined set of candidate predictors $\mathcal{F}$:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} R_{\text{emp}}(f). \tag{1.4}$$

The choice of the set of candidates $\mathcal{F}$ is of uttermost importance for learning. Roughly speaking, if $\mathcal{F}$ is too large, for example if we consider all possible measurable functions, then we may find complicated functions (in fact any function passing through all the points in the training set) with minimal empirical risk, which may however make terrible predictions on unseen examples. This phenomenon is called *overfitting*, and can be controlled by reducing the set of candidates $\mathcal{F}$. On the other hand, if $\mathcal{F}$ is too small, then it may be the case that no predictor in $\mathcal{F}$ is a good model for the input-output relationship, leading to poor predictors too. This situation is called *underfitting*. To characterize the role played by $\mathcal{F}$ in controlling overfitting, it is useful to decompose the excess risk of $\hat{f}$ compared to the optimal risk of f^* as follows:

$$\begin{aligned} R(\hat{f}) - R(f^*) &= [R(\hat{f}) - \min_{f \in \mathcal{F}} R(f)] + [\min_{f \in \mathcal{F}} R(f) - R(f^*)] && (1.5) \\ &= \text{estimation error} + \text{approximation error}. && (1.6) \end{aligned}$$

Here, the *estimation error* is due to the difficulty of approximating the true risk by the empirical risk with a limited amount of training data, while the *approximation error* is induced from approximating f^* with a restricted model space $\mathcal{F}$ that does not necessarily contain f^*. Intuitively, this error decomposition is similar to the classical *bias-variance* trade-off with the estimation error playing the role of the variance and the approximation error playing the role of bias. In this setting, a model selected in a restricted set $\mathcal{F}$ does not fit the data well and is biased. On the other hand, a model selected on a complex and large set of functions does not generalize its predictions if small changes to the data distribution occur: this is a high-variance solution. Figure 1.4 illustrates the evolution of training error (red) and generalization error on new data (blue), as functions of the estimated model complexity. The error on the training data can always be decreased by using complex models that overfit the dataset, inducing poor generalization performances. We represent with a black dot, the optimal model in the sense that it minimizes the test error which is our goal. With high-dimensional data, the estimation error can easily dominate the approximation error if $\mathcal{F}$ is not drastically controlled. This explains to some extent why simple models such as linear predictors are popular and successful in many applications of machine learning, and are the standard models in many algorithms such as linear regression, logistic regression, or support vector machines [50].

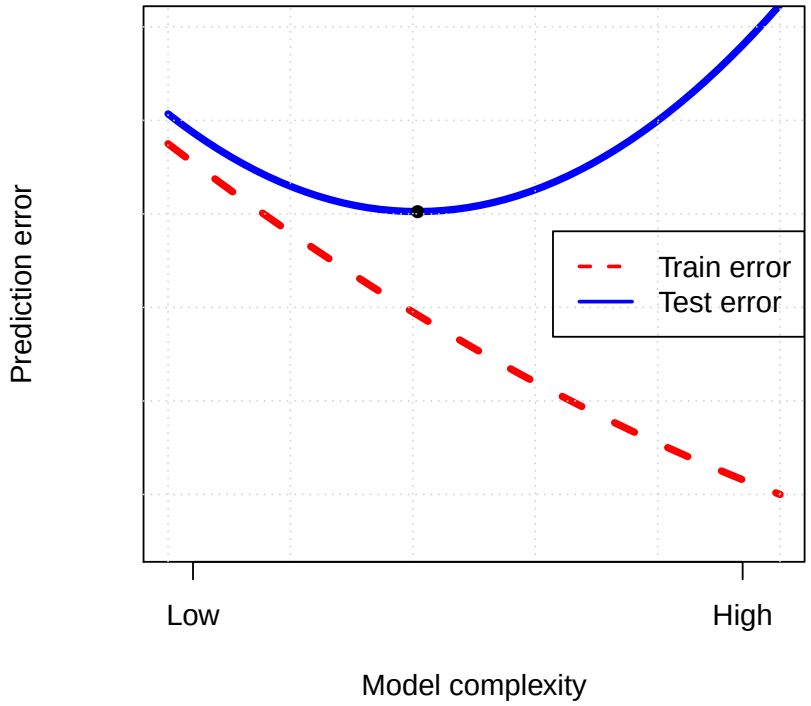

Figure 1.4: **Bias-Variance trade-off.** The prediction error on the training data (dotted red line) monotonically decreases with more complex models. For those models, the error made on new data (blue) illustrates the problem of overfitting with high error values for "too complex" models. An optimal model in terms of a trade-off between bias and variance could be the one with the minimal generalization error (black dot).

1.2.3 The choice of a loss function

The definitions of the risk (1.1) and of the empirical risk (1.3) depend on the choice of a loss function l, which we now discuss. Perhaps the most intuitive notion of risk, particularly for classification problems, is to count the number of mistakes made by the model when predicting outputs for new data. The corresponding loss function is called "gold-standard" or "0-1" loss and can be formulated as follows

$$l_{0-1}(y, f(x)) = \begin{cases} 0 & \text{if } f(x) = y, \\ 1 & \text{otherwise.} \end{cases}$$

Although this loss function is intuitively appealing, it is rarely used because it leads to computationally difficult optimization problems when we want to solve the empirical minimization problem (1.4), due to its non-convexity. Instead, it is common to consider convex surrogate loss functions, which lead to empirical risks which are convex functionals of f and can efficiently minimized. Remember that a function $h : \mathcal{Z} \to \mathbb{R}$

over a convex set $\mathcal{Z}$ is convex if [137]

$$\forall z_1, z_2 \in \mathcal{Z}, \forall t \in [0,1] : h(tz_1 + (1-t)z_2)) \leq th(z_1) + (1-t)h(z_2). \tag{1.7}$$

In the case of binary classification ($\mathcal{Y} = \{-1, +1\}$), the *hinge* loss is defined as

$$l_{\text{hinge}}(y, f(x)) = \max(0, 1 - yf(x)). \tag{1.8}$$

It is a convex loss used in particular in the support vector machine algorithm [50]. More details are given in Section 1.3.1.

Another convex loss function, commonly used in classification, is called *logistic*, because its empirical risk is linked to the log-likelihood of the logistic regression. It is defined as

$$l_{\log}(y, f(x)) = \ln(1 + \exp(-yf(x))). \tag{1.9}$$

The hinge and logistic loss functions are convex surrogates for the l_{0-1} function in binary classification. For regression ($\mathcal{Y} = \mathbb{R}$), let us mention the popular *squared error* loss function, because of its importance in linear regression problems, such as least squares regression. It is defined as

$$l_{\text{squared}}(y, f(x)) = (y - f(x))^2. \tag{1.10}$$

Note that the squared error loss can also be used in classification settings, for example, by rounding an estimated output to its closest integer.

Figure 1.5 shows the four loss functions described, as a function of the *margin* $m = yf(x)$. Apart from the squared loss which is clearly different, the hinge and logistic loss functions behave as convex versions of the 0-1 loss. We also note that the logistic loss is softer than the hinge loss which is non-differentiable for $yf(x) = 1$.

1.2.4 Regularized methods and model interpretability

In Section 1.2.2, we explained the need to restrain the set of candidate functions $\mathcal{F}$ in order to avoid overfitting and balance the estimation and approximation errors. We also mentioned that linear predictors are frequently used in machine learning problems, for their simplicity and performance. For that reason, we consider these models in the following sections. Formally, a function $f(x)$ is a linear function of $x = (x_1, \ldots, x_p) \in \mathbb{R}^p$ if it can be written as

$$f_w(x) = w^\top x = \sum_{i=1}^{p} w_i x_i$$

for a weight vector $w \in \mathbb{R}^p$. In this setting, the empirical risk minimization problem (1.4) is equivalent to

$$\hat{w} = \arg\min_{w \in \mathcal{W}} R_{\text{emp}}(f_w)\,, \tag{1.11}$$

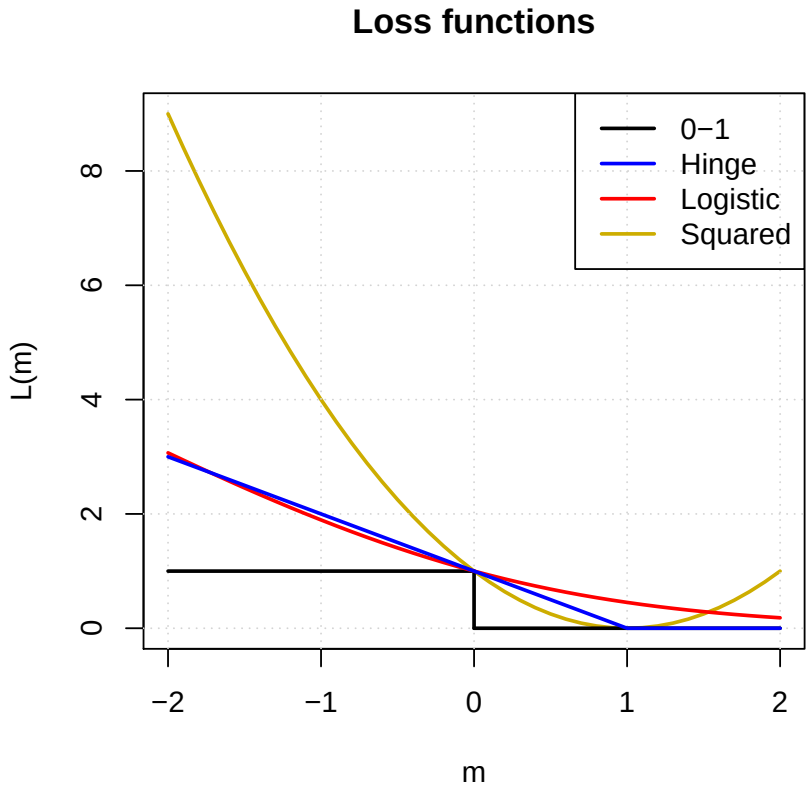

Figure 1.5: **Loss functions.** The squared (gold), logistic(red), hinge (blue) and 0-1 (black) losses are depicted in this figure.

where the set $\mathcal{W}$ is a subset of $\mathbb{R}^p$. A standard way to define $\mathcal{W}$ is through a *penalty/regularizer* function $\Omega : \mathbb{R}^p \to \mathbb{R}$, as follows:

$$\begin{aligned} \hat{w} &= \arg\min_{w \in \mathbb{R}^p} R_{\text{emp}}(w) \\ &\text{s.t. } \Omega(w) \leq \mu \,, \end{aligned} \tag{1.12}$$

where $\mu \in \mathbb{R}_+$. Interestingly, under weak assumptions on the convexity of the loss function l and of the penalty Ω, the Lagrange multiplier theory [25, Section 4.3] tells us that if $\hat{w}$ is the solution of (1.12) for a certain $\mu > 0$, there exists $\lambda \geq 0$ such that $\hat{w}$ is also a solution of the regularized problem:

$$\min_{w \in \mathbb{R}^p} R_{\text{emp}}(w) + \lambda \Omega(w). \tag{1.13}$$

This result allows some flexibility in the way to present the problem (1.11): the constrained and the regularized formulations. Even if there is no direct mapping between the two constants μ and λ, they play an inverse role in the regularization of w. When $\mu = +\infty$ (resp. $\lambda = 0$), there is not constraint on w and $\mathcal{W} = \mathbb{R}^p$. On the contrary, if $\mu = 0$ (resp. $\lambda = +\infty$), the only feasible solution is the zero vector. In the following paragraphs, we detail some regularization approaches inducing suitable properties on

w, such as smoothness, sparsity, or more complex structured constraints.

Rigde regression

We recall that the least squares estimator which minimizes

$$\min_{w\in\mathbb{R}^p} \frac{1}{2}\sum_{i=1}^{n} \left(f_w(x_i) - y_i\right)^2 = \min_{w\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{Y} - \mathbf{X}w\|_2^2$$

is given by $(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$, and that the problem is ill-posed when $p > n$ in the sense that it has multiple solutions. Ridge regression was proposed by [72] to solve the least squares problem when $p > n$, by adding a ℓ_2-norm penalty to the standard least squares problem:

$$\hat{w} = \arg\min_{w\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{Y} - \mathbf{X}w\|_2^2 + \frac{\lambda}{2}\sum_{i=1}^{p} w_i^2. \tag{1.14}$$

The solution of the problem (1.14) is indexed by the regularization parameter λ and is now seen to be

$$\hat{w}^{\text{ridge}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^\top\mathbf{Y}, \tag{1.15}$$

where $\mathbf{I}_p$ denotes the identity matrix in $\mathbb{R}^{p\times p}$. Although the main motivation for ridge regression was historically to reduce numerical issues when inverting $\mathbf{X}^\top\mathbf{X}$ by adding a positive ridge on the diagonal of the matrix, it is also beneficial for statistical reasons by controlling the estimation/approximation error balance with the penalty function $\Omega(w) = \|w\|_2^2$. This regularization has also been applied to classification learning tasks with other loss function, like support vector machines in the case of the hinge loss [50].

Sparsity-inducing penalties

The trade-off between model complexity and its generalization to new data has also been studied through feature selection and sparse models, that is, by estimating predictors that only take into account a subset of the features. Sparse models are popular because the selection of a smaller feature set can make the model more interpretable, but also because constraining a model to use only a limited number of features is a way to fight overfitting by controlling the complexity of the class of candidate models. A popular formulation to infer sparse linear models in a computationally efficient framework is the Least Absolute Shrinkage and Selection Operator (Lasso) method [165], which is similar to the ridge regression (1.14) but regularizes the squared error by an ℓ_1-norm regularization instead of an ℓ_2-norm:

$$\hat{w}^{\text{Lasso}} = \arg\min_{w\in\mathbb{R}^p} \|\mathbf{Y} - \mathbf{X}w\|_2^2 + \lambda\sum_{i=1}^{p} |w_i|, \tag{1.16}$$

where $\lambda \geq 0$ and $|.|$ denote the absolute value function. Like for ridge regression, the Lasso solution converges to the least squares solution when λ goes to zero. Figure 1.6

shows the Ridge (left) and Lasso (right) geometrical interpretations in dimension 2. In each panel, the grey region corresponds to the constraint set $\{\|w\| \leq 1\}$ and the elliptical contours are the least squares error for different solutions. The optimal solution of the constrained problem is represented with a red dot, as the first contour meeting the constraint set. For the Lasso regression, the diamond shape of the constraint set leads to a solution at a corner, where the first coordinate w_1 is equal to zero, while it can not be the case for the ridge regression, due to its circular shape.

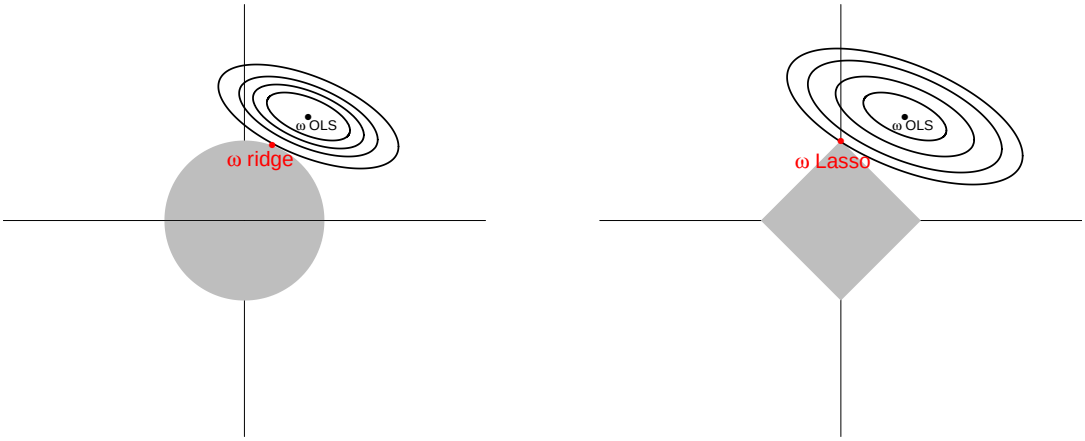

Figure 1.6: **Geometry of Ridge and Lasso regressions.** Left: the Ridge constraint $w_1^2 + w_2^2 \leq 1$. Right: the Lasso constraint $|w_1| + |w_2| \leq 1$. The elliptical contours represent some residual sums of squares. the minimization of the residual sum of squares according to the constraint corresponds to the contour tangent to the grey shape. In the Lasso case, the solution (red) occurs sometimes at a corner and corresponds to a zero coefficient in w (here, the first coordinate).

Although Lasso is performing optimally in high-dimensional settings [22], it is known to have stability problems in the case of strong correlations between the features [186]. For instance, the Lasso will randomly select one variable among a group of highly correlated variables.

Recently, many methods have been proposed to incorporate more information about the underlying structure linking the variables. To name a few, the elastic net [186] combines the ℓ_1 and ℓ_2 norms to ensure the joint selection of the correlated features, but does not explicitly take into account the actual correlation structure if it is known. The group Lasso [185] and its overlapping version [81] consider pre-defined subgroups of features and regularize the sum of the ℓ_2 norms of these groups. These approaches require a prior knowledge on the groups composition. If the correlation structure is suspected but unknown, the k-support norm was proposed by [9] as an extension of the elastic net, that consider all the possible overlapping groups of size k.

In Chapter 3, we introduce a new regularized approach useful when looking for

orthogonal or disjoint groups of features through multiple classification tasks.

1.2.5 Solving the empirical risk problem

The empirical risk minimization problem (1.4) aims to find a model w that minimizes the average loss on the training examples. In this section we discuss practical algorithms to solve this problem, when the loss function is convex. While any convex optimization problem may in theory be solved by general-purpose techniques like interior point methods, such techniques are computationally heavy in high dimensions and first-order methods such as gradient or stochastic gradient techniques are often preferred in machine learning. For simplicity, we consider unregularized problems where $\Omega \equiv 0$ in this section, the extension to regularized problem being relatively straightforward [147].

Gradient descent (GD)

In order to solve this optimization problem, several papers, like [143], proposed a *gradient descent* minimization. Considering that the gradient of the empirical risk (1.3) is available for each training point, each iteration updates the weight vector w from the previous step:

$$w_{t+1} := w_t - \frac{\gamma_t}{n} \sum_{i=1}^{n} \nabla_w l(w_t^\top x_i, y_i), \tag{1.17}$$

where γ_t is a constant or decreasing parameter. Results in [55] demonstrate that, under conditions on the starting point w_0 and the choice of γ_t, this algorithm can achieve linear convergence rates to the optimum (i.e., the distance between w_t and w^* decreases like $\exp(-t)$). Although the convergence is slower than second-order methods like Newton-Raphson in terms of number of iteration, first-order methods that only require a gradient estimation at each iteration are faster in practice.

Stochastic gradient descent (SGD)

Stochastic Gradient Descent can be thought of as a simplification of classical gradient descent, particularly useful when the number of training examples n is very large [26]. Indeed, in the gradient descent scheme, each iteration (1.17) relies on the computation of an average value over all the examples taking a time proportional to n, which can be prohibitive when n is very large. At each iteration, SGD estimates the gradient using a single and randomly picked example instead of computing a gradient on the whole training set:

$$w_{t+1} := w_t - \gamma_t \nabla_w l(w_t^\top x_t, y_t), \tag{1.18}$$

where (x_t, y_t) is the training point randomly picked at the step t. Very often, the training set is randomly permuted and the training examples are picked cyclically.

Those cycles are also called *passes* or epochs.

In terms of convergence speed, this approach is slower than the classic gradient descent with a convergence speed in $\frac{1}{t}$ at best [101]. Intuitively, SGD is able to converge as fast as gradient descent to an optimal neighborhood, but the gradient estimation on a single training point induces some variations around the optimum. However, each SGD iteration is very fast and efficient to implement, since it only requires looking at one training point at a time. In addition, the large amount of data available in many domains (e.g., health care, public sector administration, personal location data,...[111]) keeps increasing and some recent works (e.g., [131]) even consider that the training set is virtually infinite. In this setting, *on-line* learning algorithms consider the training set as a data streaming and perform a single pass over the available examples. With a infinite number of examples and an allowed training time t_{max}, one can run SGD on as many training points as possible before the imparted time, or one can select a subset of the training set that can be processed by a standard gradient descent, in memory and time. Theoretical results proposed in [28] indicate that the most efficient option is to use the maximal number of different examples with an on-line algorithm. Indeed, even if the convergence of the GD algorithm is better than SGD, the total number of examples considered by SGD will be higher than GD. Interestingly, considering a infinite training set is equivalent to drawing the examples according to the unknown probability distribution $\mathbb{P}(X, Y)$. So, instead of minimizing an empirical risk R_{emp} with a finite training set, the on-line learning algorithm will directly minimize the expected risk R. We refer the interested reader to a detailed study of the GD and SGD properties [27].

1.3 Classification

In this section, we continue our introduction to machine learning with a particular focus on classification problems, and review in particular various techniques that implement or not the regularized empirical risk minimization principle.

1.3.1 Binary classification

We first consider the simple case of discriminating only two classes, meaning that the output space $\mathcal{Y}$ is restrained to $\{-1, +1\}$. The goal of *binary* classification is to find a *decision rule* which may be used to separate the inputs X_i belonging to the different class labels Y_i. This can be interpreted as computing the posterior probabilities $p(y|x)$ for $y \in \{-1, +1\}$ and choosing the maximal value.

Fisher Linear Discriminant

The Fisher Linear Discriminant described in [124] is one of the simplest classification algorithms. The idea is to find the best direction w which maximizes the interclass

variability and minimizes the intraclass variability. The variability between the classes S_B is defined as $(\mu_{-1} - \mu_{+1})(\mu_{-1} - \mu_{+1})^\top$, where μ_{-1} (resp. μ_{+1}) is the mean value for the class "-1" (resp. class "+1"). The variability within the classes S_W is defined as

$$S_W = \sum_{\mathcal{C} \in -1,+1} \sum_{i \in \mathcal{C}} (x_i - \mu_{\mathcal{C}})(x_i - \mu_{\mathcal{C}})^\top. \tag{1.19}$$

Finding the optimal direction w is equivalent to solve the following problem

$$\hat{w} = \arg\min_{w \in \mathbb{R}^p} \frac{w^\top S_B w}{w^\top S_W w}, \tag{1.20}$$

which can be computed by using a simple procedure based on the Lagrangian formulation. To predict a new data point with Fisher Linear Discriminant, one calculates the distance from the point to the means of the projections of the training classes on the direction $\hat{w}$ and returns the closest class. There is also a weighting scheme that minimizes the bias induced by unbalanced training classes.

k Nearest Neighbours

The k Nearest Neighbours (k-NN)[51] relies on a more local classification than the Fisher Linear Discriminant. A new data point is classified according to the predominantly represented label in a neighbourhood of size k. The k closest neighbours depend of the choice of a suitable distance, which by default it often the Euclidean distance.

Nearest Prototype

In the presence of a large training dataset, computing all the distances between a new point and the training examples can be computationally expensive for the k-NN algorithm. An alternative proposed in [42] and called Nearest Prototypes consists in summarizing each class label by a small subset of training points: the prototypes. It is also possible to define a class centroid as the average prototype (Nearest Centroid approach). Here, the classification of a new data point only requires the computation of a distance value per class.

Decision tree

The Decision Tree method [33] constructs a tree structure by recursively separating the training points in subsets. In each non-terminal node, the algorithm determines, on the subset of training data affected to this node, a decision rule of the form $X_i < c_i$, where X_i corresponds to a particular variable describing the input data and c_i is a constant threshold value. There also exists some extensions that consider decision rule with linear combinations of the input variables, instead of a single variable. Based on this rule, each training point can be affected to one of the two children nodes, until it comes to a leaf node, where the prediction is made. The optimization of each decision

rule generally relies on a criterion which maximizes the homogeneity within each child node and the heterogeneity between the two children nodes. Let us introduce some more notations to described such criteria. We denote by N_m the number of training observations affected to the node m, by $R_m \subset \mathbb{R}^p$ the subspace described by the decision rule at the node m, and by p_{mk}, the proportion of training points belonging to the class k and affected to the node m:

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k). \tag{1.21}$$

The predominantly represented class in the node m is denoted by $k(m) = \arg\max_k p_{mk}$. The main criteria used to optimize the decision rules at each node include:

- Misclassification error: $\frac{1}{N_m} \sum_{x_i \in R_m} I(y_i \neq k(m)) = 1 - p_{mk(m)}$
- Gini index [63]: $\sum_{k \neq k'} p_{mk} p_{mk'} = \sum_{k=1}^{K} p_{mk}(1 - p_{mk})$
- Cross-entropy/deviance:$-\sum_{k=1}^{K} p_{mk} \log p_{mk}$.

Random Forests

Random forests [32] is a learning method that combines multiple decision trees. For each tree construction, a subset of training examples is randomly sampled, as it is done in Bagging [31], and considered as the new training set for this tree. In addition, the cut at each node is usually optimized over a random subset of the features. For the prediction of a new data point, the data is passed through all decision trees, each voting once, and the output is the most popular class.

Naive Bayes Classifier

A Naive Bayes (NB) classifier is based on applying Bayes' theorem assuming that all features in the input space are independent of each other. To label a new data point x, the posterior probability of class $C_i \in \{-1, +1\}$ given x is $P(C_i|x)$. The decision rule of the Bayes classifier is to choose the class $\hat{C}$, with the largest posterior probability.

$$\hat{C} = \arg\max_i P(C_i|x). \tag{1.22}$$

Applying the Bayes rule, the posterior probabilities $P(C_i|x)$ can be calculated by:

$$P(C_i|x) = \frac{P(x|C_i) \times P(C_i)}{P(x)}, \tag{1.23}$$

where $P(x|C_i)$ is the probability of observing x in the class C_i, $P(C_i)$ is the prior probability of observing class C_i and $P(x)$ is the unconditional probability of observing

x. Because $P(x)$ is the same for each C_i, the problem (1.22) is equivalent:

$$\hat{C} = \arg\max_i P(x|C_i) \times P(C_i). \tag{1.24}$$

Assuming conditional independence between each feature, the class-conditional probability is the product of p individual probabilities:

$$P(x|C_i) = \prod_{j=1}^{p} P(x_j|C_i). \tag{1.25}$$

In the case of discrete features (like the ones in document classification), those individual probabilities $P(x_j|C_i)$ correspond to the maximum-likelihood solution of a multinomial model [132] and are typically estimated by counting the overall proportion of each feature x_j in the C_i class members:

$$P(x_j|C_i) = \frac{\#\{x_j \in C_i\}}{\#\{x \in C_i\}}. \tag{1.26}$$

Generally, one estimates $P(C_i)$ as the proportion of examples belonging to the class C_i in the training set. Under the assumption that all $(P(C_i))_i$ are equal, the scoring function (1.24) can be simplified :

$$\hat{C} = \arg\max_i \prod_{j=1}^{p} P(x_j|C_i). \tag{1.27}$$

Support vector machine (SVM)

SVM have met significant success in numerous real-world learning tasks, including text classification [50, 85]. In its original form, the SVM algorithm is a binary classification algorithm. It aims at building a classification rule allowing to classifying instances from a space $\mathcal{X}$ as positive or negative. In other words, the SVM algorithm seeks to build a hyperplane separating the space $\mathcal{X}$ in two half-spaces. In the following we will consider the usual case where $\mathcal{X}$ is a standard Euclidean vector space, but we note that SVM can be generalized to non-vector spaces (e.g., sequences or graphs) using kernels [3]. To learn the function f, the SVM algorithm seeks to correctly classify the training data while maximizing the margin of the hyperplane, which is inversely proportional to the norm of the vector w. These two criteria are hard to fulfill simultaneously, and in practice the SVM algorithm achieves a trade-off between these two objectives. This trade-off is controlled by a parameter usually denoted as C, and the SVM solution is

obtained by solving the following optimization problem:

$$\min_{w,b,\xi} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \tag{1.28}$$

$$\text{such that :} \tag{1.29}$$

$$\xi_i \geq 0, \quad \forall i \tag{1.30}$$

$$y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \quad \forall i, \tag{1.31}$$

where $(\xi_i)_i$ are called slack variables and take values greater than 1 only for misclassified points. The standard SVM formulation is a regularized problem (1.2.4), where the penalty is the ℓ_2-norm and the loss function is the hinge loss. The C parameter plays the same role as $1/\lambda$.

In Chapter 2, we describe more complex and structured SVM formulations embedding a regularization based on a hierarchical tree distance between the different classes.

1.3.2 Multiclass extension

One may consider a more complex case, where the possible affectations for a input x belong to an extended set of labels $\mathcal{Y} = \{1, ..., K\}$ Interestingly, all the previously described approaches can be extended to the multiclass case [5]. In some cases, like for example for k-NN or decision tree classifiers, this extension is natural and simply replacing the set of labels $\{-1, +1\}$ by $1, ..., K$ is sufficient. In other cases, some specific strategy must be implemented, as summarized in the rest of this section.

Multiclass SVM

For the SVM approach, reformulations of the binary problem (1.28) have been proposed to handle the multiclass case [176, 30, 52, 166]. However, the formulations in [176, 30] result in a single constrained problem that can be unfeasible for a large K, while those in [52, 166] are more efficient for a large number of classes. These approaches generally learn simultaneously a set of class specific weight vectors $w_k \in \mathbb{R}^p$, for $k = 1, ..., K$. To do so, the idea is to learn the weight vectors from the training dataset such that the highest scores are given by the scoring functions of the appropriate class. Formally, we want to achieve the following criterion:

$$\langle w_{y_i}, x_i\rangle \geq \langle w_k, x_i\rangle \quad \text{for } k \in \mathcal{Y} \setminus y_i, \text{ and } i = 1, ..., N,$$

where "\" is the set exclusion operator. To efficiently solve this problem, we adopt a SVM-like formulation where we enforce a margin in the above constraints

$$\langle w_{y_i}, x_i\rangle \geq \langle w_k, x_i\rangle + 1 \quad \text{for } k \in \mathcal{Y} \setminus y_i,$$

but tolerate margin violations

$$\langle w_{y_i}, x_i \rangle \geq \langle w_k, x_i \rangle + 1 - \xi_i \text{ with } \xi_i \geq 0, \text{ for } k \in \mathcal{Y} \setminus y_i.$$

Altogether, this gives the following optimization problem [52]:

$$\min_{\{w_k\}_{k=1..K}, \xi} \frac{1}{2} \sum_{k=1}^{K} \|w_k\|^2 + C \sum_{i=1}^{n} \xi_i \tag{1.32}$$

$$\text{such that :} \tag{1.33}$$

$$\xi_i \geq 0, \ \forall i \tag{1.34}$$

$$\langle w_{y_i}, x_i \rangle \geq \langle w_k, x_i \rangle + 1 - \xi_i, \ \ \forall i, \ \ \forall k \in \mathcal{Y} \setminus y_i. \tag{1.35}$$

Note that the prediction step uses the decision rule $\hat{C}(x) = \arg\max_k \langle w_k, x \rangle$.

This extension of the binary SVM to a multiclass setting requires to change the original formulation and in some cases, there is no evidence that a single mathematical function correctly separates all the represented classes [2]. Furthermore, the most standard and easiest way to address multiclass classification problems with SVMs is to combine binary classifiers into a multiclass classification rule, as explained below.

One-versus-all (OVA)

The *one-versus-all* scheme [135] consists in learning a set of K binary SVMs trained to separate each of the K classes from the $K-1$ other ones, leading to a set of hyperplanes $\left\{w_k\right\}_{k=1,\ldots,K}$, and the class predicted for the instance x is the one obtaining the highest score, as in the multiclass formulation. Compared to the multiclass scheme, we end up with K problems instead of a single optimization problem, with however a lesser number of constraints than the unique multiclass problem. The benefit that can be expected by the multiclass formulation with respect to the OVA scheme is to obtain better classification performances due to a better calibration of the K scoring functions used to make the prediction. Indeed, in the one-versus-all scheme, no mechanism explicitly enforces the scoring functions of a given class to be higher to those of other classes (and in particular to similar ones). However, [136] provides performances comparable to multiclass approaches.

One-versus-one (OVO)

The *one-versus-one* scheme [69] is also called pairwise classification. A set of $\frac{K(K-1)}{2}$ binary SVMs is trained to distinguish between every pair of classes, and the class predicted for an instance x is the one obtaining the highest number of votes (a number between 0 and $K-1$) according to these classifiers. Results in [75] suggest that this approach can perform better than the OVA formulation.

Error-Correcting Output-Coding (ECOC)

This approach uses the concept of error-correcting codes detailed in [56]. It works by training a fixed N number of binary classifier that is greater than K. Each class is then represented by a different binary code of length N. The class codes can be summarized in a binary matrix in $\{0,1\}^{K\times N}$, where each row is a class code. For each column of this matrix, a classifier is learned using the zero-labeled classes as negative examples and the other ones as positive examples. The label prediction of a new data point is done by putting the N predictions into a binary code and by returning the closest class in terms of Hamming distance [68] between the class codes and the predicted code. Results reported in [56] show a better generalization ability of ECOC over the OVA and OVO formulations.

Error-Correcting Tournaments (ECT)

The multiclass formulations presented above have a running time which is $\mathcal{O}(K)$ [135] and does not scale very well with large K classification problem. An alternative to these strategies is described in [20] and is called *Error Correcting Tournament* (ECT). This approach operates in two phases, described in the Figure 1.7. The first step consists in m single-elimination tournaments over the K labels. For each tournament, labels are paired at the first round and the winners of each round play a second round, and so on. At the end of a tournament, there is a single winner: the predicted label for this tournament. Then, given the m predicted labels, there is an "All-Star" tournament in order to decide which winner label is the final prediction returned by the algorithm.

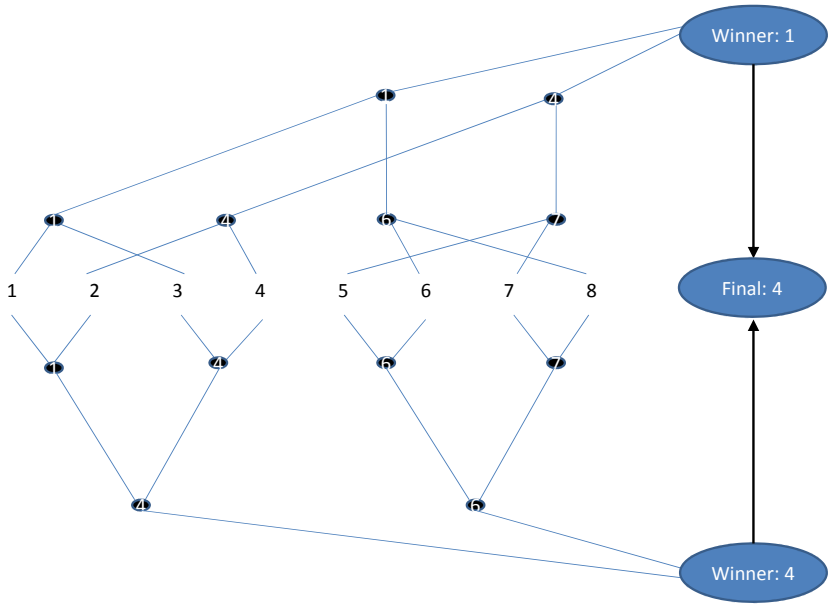

Figure 1.7: **Error-Correcting Tournaments.** This is an example of $m = 2$ elimination tournaments with $K = 8$ classes. Each tournement has its own tree structure, leading to different winners for the same data point ot classify. A final round between the different tournaments winners allows to select the predicted label.

Interestingly, this approach provides a complexity for training and test steps in $\mathcal{O}(\log(K))$ which is well suited to classification problems with large number of classes. As a side effect, the gain in computation speed for ECT is counterbalanced by decreasing accuracy performances [44].

However, there is no clear evidence that, in general, a formulation is better than the others. This suggests that the best multiclass strategy is problem dependent.

1.4 Model evaluation

In order to efficiently compare the different machine learning approaches, one needs standard evaluation rules. In this section, we describe some of them that are used in the next three chapters. As stressed in Section 1.2.2, a good predictive model should demonstrate high generalization on a new data set.

1.4.1 Accuracy measures

Let us assume that we have a so-called *test set* of n input-output pairs that was not used to train a predictor $\hat{f}$. Here we discuss how it can be used to estimate the performance of $\hat{f}$ on future data.

The performance indicators we consider obviously depend on the learning task. In a regression context, like in Section 3.6.1, we can compute the average l_2 loss error, also called Mean Squared Error (MSE)

$$\text{MSE}(Y, \hat{Y}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y} - y)^2, \tag{1.36}$$

where $Y = (y_1, ..., y_n)$ is the vector of actual responses and $\hat{Y} = (\hat{y}_1, ..., \hat{y}_n)$ is the vector of values estimated by the predictive model.

For the classification tasks evaluation, we may consider several indicators. First, a common measure is the correct classification rate, also called *micro accuracy* [146, 110]. It is defined as the proportion of correctly labeled examples for a given dataset and directly involves the "0-1" loss

$$\frac{1}{n}\sum_{i=1}^{n}\text{I}(\hat{y}_i, y_i), \tag{1.37}$$

where I is the indicator function equal to 1 if the compared terms are equal and 0 otherwise. In the multiclass classification context, unbalanced class sizes may induce a bias in the micro-accuracy score: large classes dominate small classes in the overall correct classification rate. To put similar weights on small classes, [110] proposed a *macro accuracy* score

$$\frac{1}{K}\sum_{c\in\mathcal{C}}\frac{1}{N_c}\sum_{i\in c}\text{I}(\hat{y}_i, y_i), \tag{1.38}$$

where K is the number of classes in $\mathcal{C}$ and N_c is the number of data points belonging to the class c. In Chapter 4 experiments, we also use an alternative indicator by using a median value which is less sensitive to the outliers.

$$\text{median}_{c \in \mathcal{C}} \left(\frac{1}{N_c} \sum_{i \in c} \mathrm{I}(\hat{y}_i, y_i) \right). \tag{1.39}$$

Other specific performance indicators are considered in the following chapters, where they embed problem structure, like a tree distance in Section 2.4 or sparsity patterns in Section 3.6.2. Details are given in the corresponding sections.

1.4.2 Model selection and model assessment

As described in Section 1.3, there exists a plethora of methods to build predictive models from a training set of input-output pairs. Finding the best predictive model among all the possible models and the corresponding hyper-parameters is an important step called *model selection*. In general, if the amount of available data is large enough, one should split the dataset in independent parts [70, Chap. 7] to avoid overfitting described in Section 1.2.2. The *holdout* method considers three disjoint subsets: the *training* set, the *validation* set and the *test* set. The training set is used to learn different models corresponding to different methods or parameters. An estimation of the prediction error is made on the validation set and an optimal model is selected. During this evaluation, the test set is kept aside and is used at the end to estimate the generalization error of the selected model.

This approach has two main drawbacks, as discussed in [91]. Under the assumption that the more data an algorithm processes, the higher the accuracy is, the holdout method is a pessimistic estimator because only a fraction of the available data, commonly 2/3, is used to train the models. In addition, the holdout method relies on a single random split and the estimation of the error rate can be misleading given a fortunate or unfortunate split.

1.4.3 Cross-validation procedures

Other approaches have been proposed to better use the available data and estimate the prediction error of models. Probably the most widely used method is *cross-validation*, where the dataset is split in N balanced and exclusive folds, and we operate a rotation over these subsets with $N-1$ folds used for learning and one fold for evaluating, as illustrated in Figure 1.8. This procedure returns N estimated accuracy values that are averaged to obtain the final prediction error. However, this value is estimated on a single split of N folds. The complete cross-validation estimation [91] is the average over all possible N folds splits and is too expensive. For practical reasons, one commonly consider multiple random splits. A particular case where $N = n$ is known as *leave-one-out* (LOO) and presents a low bias with respect to the actual prediction error but

a possible high variance. Indeed, the N training sets only differ of one single instance and are very similar. In addition, learning n different models over the whole dataset can be a computational burden. Good compromise values, like $N = 5$ or $N = 10$ are proposed in [34, 91].

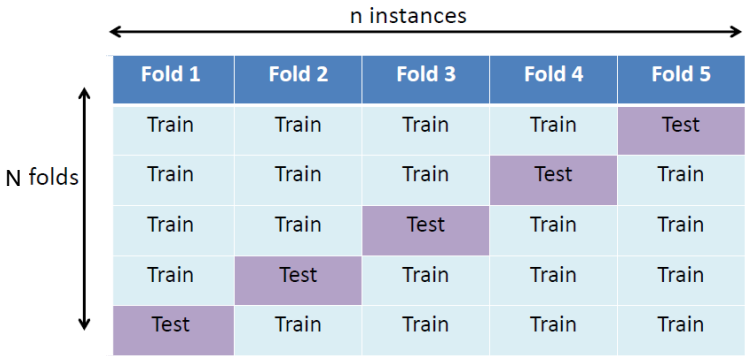

Figure 1.8: **Cross-validation for model selection.** This is an example of $N = 5$ folds cross-validation strategy. The dataset is splitted in N subsets and by rotation over these subsets, N different models are learned using $N - 1$ folds and an estimation of the prediction error is computed on the remaining fold.

In the classification context on unbalanced datasets, unfortunate random splits can lead to remove all the members of a given class from some training folds. To overcome this issue, one may consider a *stratified* N-fold cross-validation, where the different folds contain approximately the same proportions of class members as the whole dataset.

1.5 Contribution of this thesis

1.5.1 Microbial identification based on mass-spectrometry data

Identification of microorganisms using proteomics fingerprints obtained with mass-spectrometry experiments can be viewed as a multiclass classification problem. The goal of a trained model is to correctly discriminate hundreds of different microbial species present in a reference database made of labeled mass-spectrometry data. When using a VITEK-MS instrument, each raw mass-spectrum is pre-processed and represented as a vector of intensity peaks belonging to a high-dimensional space of more than thousand variables. These data cannot be analyzed by human experts due to the large amount of parameters. Previous works [54] have already proved that building a smart microbial identification system by using machine learning algorithm is a rel-

evant approach. In this first study, we evaluate the potential of structured machine learning approaches by embedding the taxonomic organization of the microbial species and compare them to standard learning methods. On a challenging internal real-world dataset, we investigate the gain in accuracy when adding *a priori* information.

1.5.2 Jointly learning tasks with orthogonal features or disjoint supports

Common multi-task learning problems involve learning several related predictive models, enforcing some form of similarities between the different models.

In this second study, we investigate the opposite problem where one constrains the different models to be orthogonal. This way, we jointly learn unrelated tasks. Previous works demonstrate the relevance of such approaches for emotion detection on human faces [139].

We investigate the regularization developed in [184] and propose more general formulations involving non-convex penalties. We provide an evaluation on synthetic and real-world datasets. A natural extension of the previous approach consists in combining orthogonality and sparsity constraints. It leads to models with disjoint supports, meaning that if a feature belongs to a model task, it is not used by the others models.

1.5.3 Taxonomic assignation of sequencing reads from metagenomics samples

Metagenomics characterizes the microbial diversity by sequencing of DNA directly from an environmental sample. Due to the large volume of metagenomics datasets, one of the main challenging steps is the taxonomic sequence assignment, also called binning. In this work, we investigate the potential of modern, large-scale machine learning implementations for taxonomic affectation of next-generation sequencing reads. The resulting models are competitive with well-established alignment tools for problems involving a small to moderate number of candidate species, and for reasonable amount of sequencing errors. Our results suggest, however, that compositional approaches are still limited in their ability to deal with problems involving a greater number of species. We finally confirm that compositional approach achieve faster prediction times.

Chapter 2

Benchmark of structured machine learning methods for microbial identification from mass-spectrometry data

This chapter has been submitted under a slightly different form as [172], a joint work with Pierre Mahé, Jean-Baptiste Veyrieras and Jean-Philippe Vert.

Abstract

Microbial identification is a central issue in microbiology, in particular in the fields of infectious diseases diagnosis and industrial quality control. The concept of species is tightly linked to the concept of biological and clinical classification where the proximity between species is generally measured in terms of evolutionary distances and/or clinical phenotypes. Surprisingly, the information provided by this well-known hierarchical structure is rarely used by machine learning-based automatic microbial identification systems. Structured machine learning methods were recently proposed for taking into account the structure embedded in a hierarchy and using it as additional *a priori* information, and could therefore allow to improve microbial identification systems.

We test and compare several state-of-the-art machine learning methods for microbial identification on a new Matrix-Assisted Laser Desorption/Ionization Time-of-Flight mass spectrometry (MALDI-TOF MS) dataset. We include in the benchmark standard and structured methods, that leverage the knowledge of the underlying hierarchical structure in the learning process. Our results show that although some methods perform better than others, structured methods do not consistently perform better than their "flat" counterparts. We postulate that this is partly due to the fact that standard methods already reach a high level

of accuracy in this context, and that they mainly confuse species close to each other in the tree, a case where using the known hierarchy is not helpful.

Résumé

L'identification microbienne est une étape clé en microbiologie, et tout particulièrement dans les domaines du diagnostic de maladies infectieuses et du contrôle qualité en industrie. Le concept d'espèces microbiennes est étroitement lié à la notion de classification biologique et clinique, où la proximité entre les espèces se mesure généralement en termes de distances évolutives et/ou de phénotypes cliniques. Étonnamment, l'information fournie par de telles structures hiérarchiques est rarement utilisée lors de la construction de systèmes automatiques d'identification microbienne. Récemment, des approches structurées d'apprentissage statistique ont été proposées pour prendre en compte la structure hiérarchique qui organise les classes et l'utiliser comme une information *a priori* afin d'améliorer les systèmes d'identification microbienne. Dans cette étude, nous avons évalué plusieurs approches d'apprentissage faisant références, sur un jeu de données contenant des spectres de masse issus de la technologie MALDI-TOF (Matrix-Assisted Laser Desorption/Ionization Time-of-Flight). Nous donnons les détails sur la manière de comparer au mieux des approches standards et structurées, qui incorporent la connaissance d'une organization taxinomique dans le processus d'apprentissage. Bien que nos résultats montrent que certaines méthodes ont de meilleures perfomances, utiliser des méthodes structurées n'améliore pas les résultats sur le jeu de données considéré. Nous supposons que cela provient du fait que les méthodes standards atteignent un très bon niveau de performance dans ce contexte, où les confusions dans la classification concernent généralement des couples d'espèces taxinomiquement proches. Ce type d'erreur n'est pas corrigé par les méthodes structurées considérées dans cette étude.

2.1 Introduction

Microbial identification is the task of determining to which species a microorganism isolated from a clinical or industrial sample belongs. It plays a central role in the diagnosis of infectious diseases and industrial quality control. In the clinical setting, identification is often the first step towards a finer characterization of the microorganism, aiming in general to establish its virulence and/or antibiotic resistance profiles, which is ultimately used by the clinician to prescribe a therapy.

Since the proof of concept of bacterial identification with MALDI-TOF MS [8], , this high-throughput technology has been improved up to a genuine paradigm breaking technology in microbiology, allowing to quickly, cheaply and efficiently characterize a microorganism [24, 43, 62, 162]. Starting from an isolated colony of the targeted microorganism, MALDI-TOF MS provides a snapshot of its proteomic content. Such a proteomic fingerprint is highly species specific, and can be used to identify a microorganism by matching it with a reference database of annotated fingerprints [167].

At the basis of MALDI-TOF MS identification system is therefore a software component in charge of finding the closest match between the fingerprint of the unknown

microorganisms and the reference fingerprints of the database. From the data analysis perspective, this can be formalized as a multiclass classification task. This learning task presents several challenging issues. First, MALDI-TOF mass spectra are measured on several tens of thousands of mass to charge channels, and although they are generally pre-processed in order to extract their predominant peaks [49], the resulting peak lists are still high-dimensional vectors. Moreover, current commercial systems like the Biotyper (Bruker Daltonics, Germany), LT2 (Andromas, France), or VITEK-MS (bioMérieux, France) address several hundreds of species [113], which constitutes a relatively massive multiclass problem. Finally, the number of observations per class, that is, of representative strains per species, is often limited, which leads to strongly unbalanced datasets. On the other hand, the classes of the problem correspond to microbial species which can be organized into well known hierarchical structures, generally defined in terms of evolutionary distances and/or phenotypic differences. Such tree structures provide a rich source of information that could be added as prior knowledge within the training of automatic microbial identification systems. Several "structured" machine learning methods were indeed recently proposed for taking into account the structure embedded in a hierarchy and using it as additional *a priori* information [73, 166, 158, 58], and could potentially be used to train microbial identification systems. Surprisingly, however, this possibility has not been investigated to our knowledge, and current systems implement "flat" multiclass classification algorithms that do not take into account the known tree structure. In this paper, we evaluate the relevance of structured machine-learning methods in the context of microbial identification from MALDI-TOF mass spectra. For that purpose, we use the MicroMass dataset [108] to benchmark several "flat" and "structured" machine learning techniques.

2.2 Benchmark dataset

The dataset considered in this benchmark is described in Table 2.1. It involves 20 Gram positive and negative bacterial species covering nine genera. Pathogen strains of *Bacillus cereus* cause foodborne illness, but this bacterium is also use as probiotics for animals. *Bacillus thuringiensis* is a non-human pathogen and produces toxins which are very useful in insecticides design. Both *Citrobacter braakii* and *Citrobacter freundii* are opportunistic pathogens that represent around 30% of nosocomial infections [177]. The bacterium *Clostridium difficile* is known to survive in clinical environments because of a resistance to alcohol and most hospital disinfectants [57], and thus responsible of around 30% of nosocomial diarrhea [99]. Some *Enterobacter cloacae* strains are known pathogens in respiratory track infections, in particular for ventilator-associated pneumonia. *Escherichia coli* is a very common bacteria covering 80% of the human intestinal aerobic flora. However, some strains are virulent and cause urinary track infections, kidney failures and hemorrhagic diarrhea. Although they are very close to *Escherichia coli*, all *Shigella* genus members are human-only pathogens and causative

agents of shigellosis (dysentery). In developed countries, *Shigella sonnei* is responsible of 70% of shigellosis cases [148]. *Haemophilus influenzae* is an opportunistic pathogen causing otitis and meningitis, while *Haemophilus parainfluenzae* is a commensal microbe of the human mouth, rarely pathogen. Among the 10 *Listeria* genus members, *Listeria monocytogenes* and *Listeria ivanovii* are the two bacterial species known to be pathogen. While the first one can cause listeriosis in human, the second one is mostly found in ruminants. The two *Streptococcus* genus members are known to be genetically close, belonging to the *Streptococcus mitis* group [90], even if *Streptococcus mitis* is a commensal microbe in the human mouth and *Steptococcus oralis* is an opportunistic pathogen. Finally, *Yersinia enterocolitica* is the most frequent agent causing yersiniosis and *Yersinia frederiksenii* is a rarely pathogen bacteria that may cause gastrointestinal infections [37].

Table 2.1: **MicroMass dataset**. This table describes the MicroMass dataset content, in terms of used bacterial genera and species. It also provides information on the number of bacterial strains and mass-spectra for each species.

Species name	Species ID	Number of strains	Number of spectra
Bacillus cereus	BAC.CEU	10	26
Bacillus thuringiensis	BAC.THU	8	11
Citrobacter braakii	CIT.BRA	9	26
Citrobacter freundii	CIT.FRE	10	28
Clostridium difficile	CLO.DIF	7	14
Clostridium glycolicum	CLO.GLY	9	16
Enterobacter asburiae	ENT.ASB	10	29
Enterobacter cloacae	ENT.CLC	16	52
Escherichia coli	ESH.COL	20	60
Haemophilus influenzae	HAE.INF	18	50
Haemophilus parainfluenzae	HAE.PAR	9	21
Listeria ivanovii	LIS.ISI	9	29
Listeria monocytogenes	LIS.MNC	10	31
Shigella boydii	SHG.BOY	9	18
Shigella flexneri	SHG.FLX	10	32
Shigella sonnei	SHG.SON	10	31
Streptococcus mitis	STR.MIT	10	26
Streptococcus oralis	STR.ORA	9	24
Yersinia enterocolitica	YER.ETC	10	27
Yersinia frederiksenii	YER.FRD	10	20

This dataset was extracted from the reference database embedded in the commercial VITEK-MS system and made public through the UCI machine learning repository[1] . Each species is represented by 11 to 60 mass spectra obtained from 7 to 20 bacterial strains, leading altogether to a dataset of 213 strains and 571 spectra. These spectra were obtained according to the standard workflow used in clinical routine in which the microorganism was first grown on an agar plate from 24 to 48 hours, before some

[1] http://archive.ics.uci.edu/ml/datasets/MicroMass

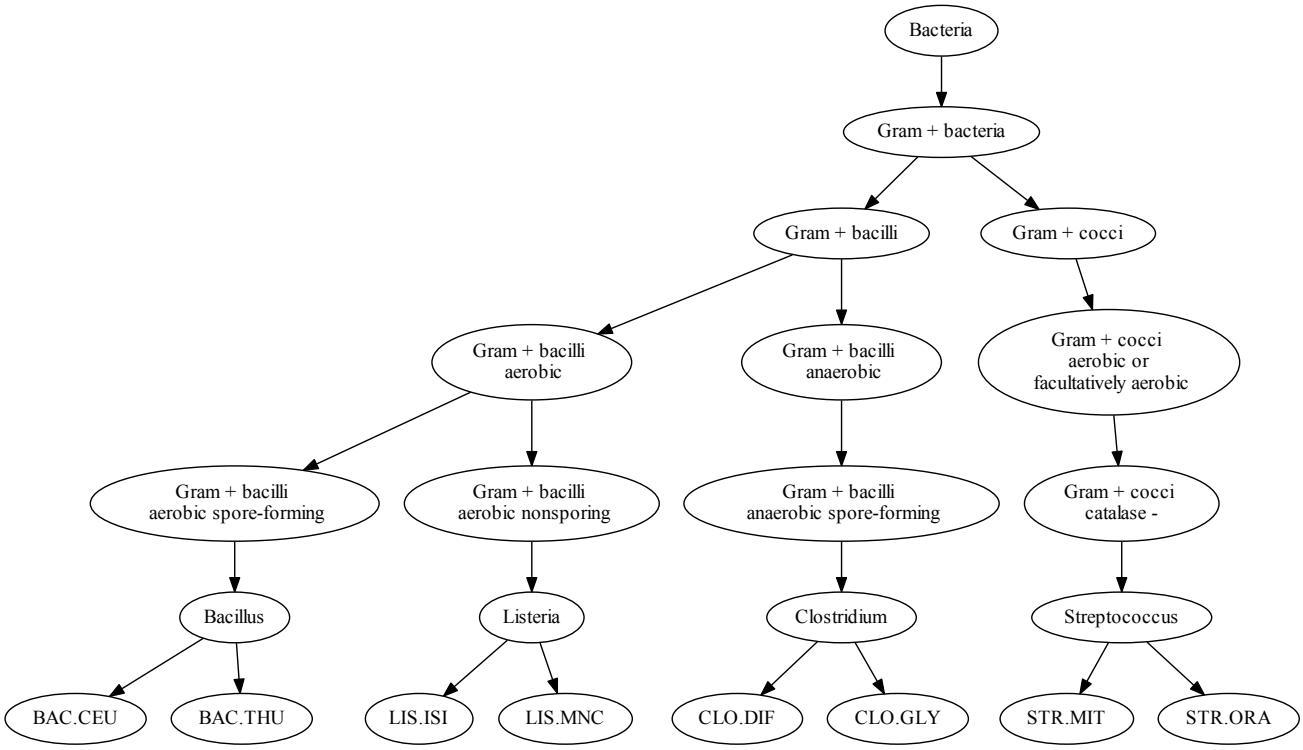

Figure 2.1: **MicroMass hierarchical tree structure (Gram + bacteria).** This tree shows the hierarchical organization of the bacterial panel considered in this benchmark, that belong to the Gram + bacteria. The leaves of the tree correspond to the 8 species and their parent to the 4 genera. Internal nodes correspond to either phenotypic (*e.g.* aerobic and anaerobic at the top of the tree) or taxonomic attributes.

colonies were picked, spotted on a MALDI slide and a mass spectrum was acquired.

The 20 bacterial species involved in this study and the underlying hierarchical tree are shown in Figures 2.1 and 2.2.

We note that the tree considered in this study involves both phenotypic and evolutionary traits, its uppermost level separating species into Gram positive and Gram negative, and its two lowest levels corresponding to the species and genus taxonomic ranks. Such a hybrid hierarchical definition is common in the context of clinical microbiology, where manual identification involves a succession of tests to establish several phenotypic and metabolic properties of the microorganism to identify (e.g., Gram +/- or aerobe/anaerobe). These properties correspond to the upper levels of the tree, while the lower ones correspond to standard phylogenetic ranks (e.g., family, genus and species). We also note that this dataset contains several pairs of groups of species known to be hard to discriminate in general. This is for instance the case of the *Bacillus cereus* and *Bacillus thuringiensis* species, which are known to belong to the *Bacillus cereus group* [71], as well as the *Escherichia coli* species and the species of the *Shigella* genus, which are often considered to belong to the same species [97]. Accordingly, *Escherichia coli* and the three *Shigella* species involved in this dataset were gathered

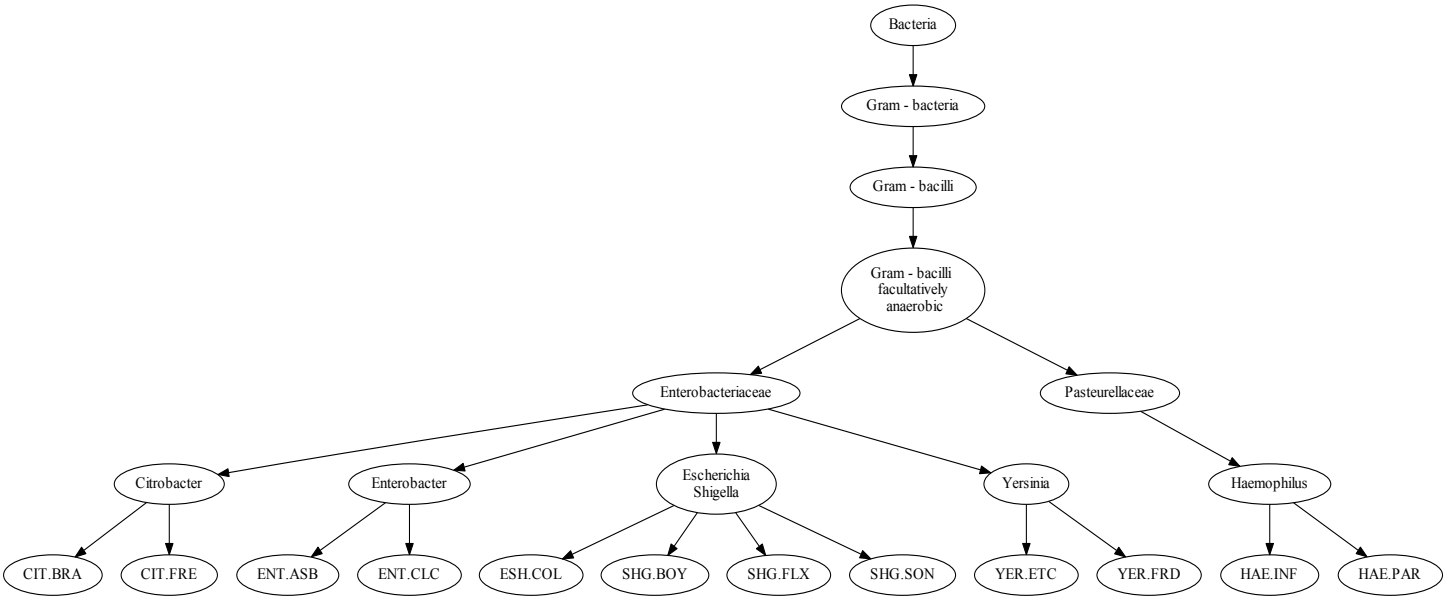

Figure 2.2: **MicroMass hierarchical tree structure (Gram - bacteria).** This tree shows the hierarchical organization of the bacterial panel considered in this benchmark, that belong to the Gram - bacteria. The leaves of the tree correspond to the 12 species and their parent to the 5 genera. Internal nodes correspond to either phenotypic (*e.g.* aerobic and anaerobic at the top of the tree) or taxonomic attributes.

in a common genus.

We note finally that we have considered in this study a peak-list representation in which a mass spectrum is represented by a vector $x \in \mathbb{R}^p$, where p is the numbers of bins considered to discretize the mass to charge range, and each entry of x is derived from the intensity of the peak(s) found in the corresponding bin. While several schemes have been proposed to define such a peak-list representation [49], we have relied here on the approach embedded in the VITEK-MS system, which provides a peak-list representation of dimension $p = 1300$, with typically between 50 and 150 peaks per spectrum. Further details about this dataset are available in [108].

Figure 2.3 represents a clustered version of the MicroMass dataset, where the rows correspond to 571 mass-spectra ordered according to their genus label and the columns are the 1300 intensity peaks grouped by an unsupervised clustering step. Interestingly, we remark block structures suggesting that some features uniquely belong to one genus class.

2.3 Structured classification methods

In this section we provide a brief description of the various classification strategies considered in this study. The multiclass formulation detailed in Section 1.3.2 opens way to the development of cost-sensitive multiclass classifiers and of so-called structured

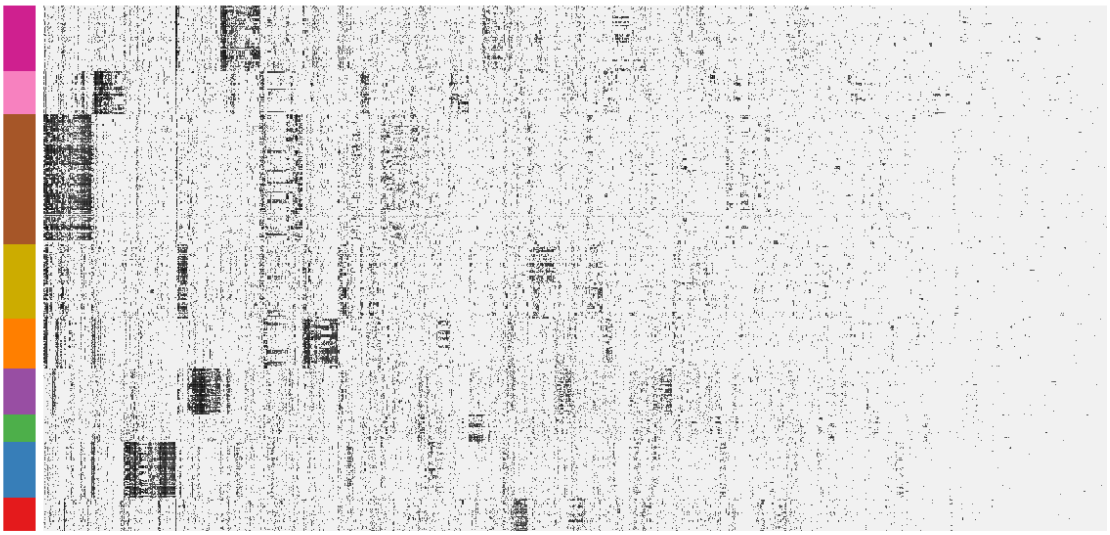

Figure 2.3: **MicroMass dataset visualization**. The heatmap shows the dataset organisation of the mass-spectra considered in our benchmark. The rows correspond to 571 spectra ordered according to the 9 different genera/colors. The columns are the 1300 intensity peak variables grouped by an unsupervised clustering step. Each black dot is a non-zero value in the dataset, while the light grey zones have a null signal.

classifiers, that we introduce in the two following sections.

2.3.1 Cost-sensitive multiclass SVMs

For practical applications, different errors can have different impact: it can be less severe to mistake class A for class B than class A for class Z for instance. This is notably the case for microbial identification which can orient therapy before antibiotic susceptibility results are available. Cost-sensitive classifiers distinguish between the various types of classification errors and penalize them differently in the learning process. The multiclass formulation can be easily modified to accommodate such a cost-sensitive mechanism [166]. Indeed, assume that a loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is available[2] such that $\Delta(y, y')$ quantifies the loss, or severity, of predicting class y' if the true class is y, $\Delta(y, y') > 0$ for $y \neq y'$ and $\Delta(y, y) = 0$. Such a loss function can be leveraged in the training process through a redefinition of the constraints involved in the underlying optimization problem, according to one of the following re-definitions of the constraints $\langle w_{y_i}, x_i \rangle \geq \langle w_k, x_i \rangle + 1 - \xi_i$ in Equation (1.32) as:

- $\langle w_{y_i}, x_i \rangle \geq \langle w_k, x_i \rangle + \Delta(y_i, k) - \xi_i$ in the so-called "margin-rescaling" formulation,
- $\langle w_{y_i}, x_i \rangle \geq \langle w_k, x_i \rangle + 1 - \frac{\xi_i}{\Delta(y_i,k)}$, in the so-called "slack-rescaling" formulation.

Both redefinitions have the effect of adjusting the strength of the constraints according to the loss function. In the margin-rescaling formulation, the value of the margin becomes proportional to the loss while in the slack-rescaling formulation, we keep a unity

[2]Note that in practice this loss function can be summarized as a $K \times K$ matrix.

margin but penalize more strongly margin violations associated to a high loss. Note that the standard formulation corresponds to using a binary loss function: $\Delta(y, y') = 1$ for $y \neq y'$ and $\Delta(y, y) = 0$. For practical applications, this cost-sensitive formulation allows to leverage the training process prior information about the relationship between the classes and/or requirements about the classification performances expected. In this study we call this approach `TreeLoss` and use $\Delta(y, y')$ as the length of the shortest path connecting the two species in the considered tree.

2.3.2 Hierarchy structured SVMs

The structured SVM formulation of [166] enables to make a further use of the hierarchical structure underlying the microbial identification multiclass problem. Indeed, it does not only allow to leverage a loss function in the learning process to penalize misclassifications involving hierarchically distant species, but it also introduces new variables that can be further exploited by the algorithm. For the sake of clarity and to highlight the differences between the approaches, we start by casting the multiclass SVM model (Section 1.3.2) in the structured SVM framework.

For that purpose, we concatenate the weight vectors $w_k \in \mathbb{R}^p$ for $k = 1, ..., K$ into a single vector $W = [w_1 w_2 ... w_K] \in \mathbb{R}^{p \times K}$ and introduce a mapping function $\Lambda : \{1, ..., K\} \to \mathbb{R}^K$ defined as:

$$\Lambda(k) = \Big(\mathbf{1}(k = j)\Big)_{j=1,...,K},$$

where the function $\mathbf{1}(.)$ is equal to one if its argument is true and zero otherwise. This function therefore maps a label $k \in \mathcal{Y} = \{1, .., K\}$ to a binary vector of length K with all but the k-th entry, which is set to 1, set to 0. Based on this function Λ, a *joint input-output representation* $\Psi(x, y)$ of the (feature vector,class label) pair (x, y) can be defined as:

$$\Psi(x, y) = x \otimes \Lambda(y),$$

where $\otimes$ denotes the tensor-product operator which is defined as:

$$\otimes : \mathbb{R}^p \times \mathbb{R}^K \to \mathbb{R}^{p \times K}, \text{ such that} (a \otimes b)_{i+(j-1)p} = a_i b_j.$$

The above definition of Λ induces a block structure in the vector $\Psi(x, y)$, which consists of K repetitions of the feature vector x, with all but the y-th block set to zero, as represented in Figure 2.4.

It is then relatively easy to see that $\langle w_k, x \rangle = \langle W, \Psi(x, k) \rangle$, hence that the original

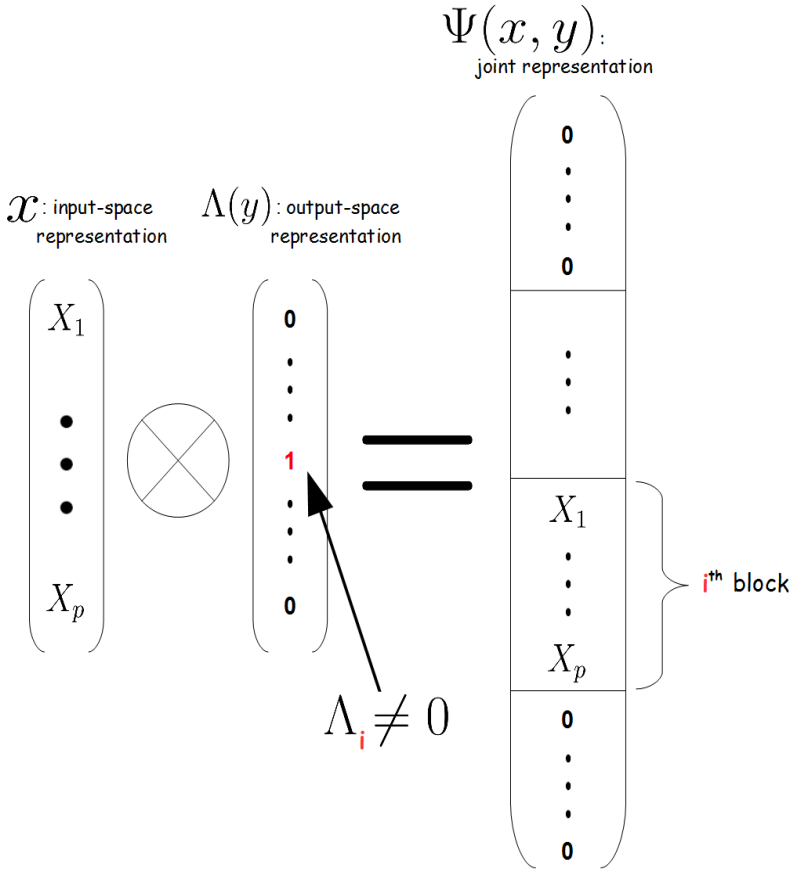

Figure 2.4: **Joint mapping or Multiclass SVM.** Given a example $x \in \mathbb{R}^p$ and the associated label y, we derive a joint representation $\Psi(x, y)$ by using the tensor product between the input representation and the output representation. This new representation has a sparse block structure with copy of x only for the block corresponding to label y (red). It allows to cast the multiclass SVM formulation into the structured SVMs framework.

multiclass optimization problem ((1.32)) can equivalently be written as:

$$\min_{W=[w_1 w_2 ... w_K], \xi} \quad \frac{1}{2}||W||^2 + C\sum_{i=1}^{N} \xi_i \tag{2.1}$$

such that:

$$\xi_i \geq 0, \ \forall i$$

$$\langle W, \Psi(x_i, y_i)\rangle \geq \langle W, \Psi(x_i, k)\rangle + \Delta(y_i, k) - \xi_i, \quad \forall i, \quad \forall k \in \mathcal{Y} \setminus y_i,$$

with $\Delta(y, y') = 1$ for $y \neq y'$ and $\Delta(y, y) = 0$.

Then the classification rule becomes:

$$G(x) = \arg\max_k \langle W, \Psi(x, k)\rangle, \quad k \in \mathcal{Y} = \{1, ..., K\}\,. \tag{2.2}$$

Interestingly, this approach can be generalized to build predictive models consider-

ing an output space $\mathcal{Y}$ presenting an arbitrary structure, thereby dramatically increasing the scope of the problems that can be addressed by this approach. Owing to the structured nature of the output space, the term structured SVMs was coined in [166].

Generally speaking, to carry out such a structured SVM approach, one has to define:

- the joint feature representation $\Psi(x, y)$,
- the loss function $\Delta(y, y')$ quantifying the severity of mistaking pairs of instances of the output space,
- the algorithm in charge of computing $\text{argmax}_y \langle W, \Psi(x, y) \rangle$, which is needed both at the prediction step and during the training step. Indeed, the cutting-plane algorithm that is typically used to solve the optimization problem underlying structured SVMs relies on this operation to navigate in the space of constraints that the model needs to satisfy.

In practice, the main difficulty resides in this latter step, which can be challenging if the cardinality of the output space is large.

We now introduce a hierarchy structured-SVM formulation (`Structured`) that has been applied to classifications of text documents [73, 166] or next-generation sequencing reads [128], and that can be used for our purpose to leverage a hierarchical structure reflecting the proximity of the bacterial species considered. We still assume that our identification problem involves K distinct species, the classes of the above multiclass problems, which correspond to the leaves of a taxonomy made of $T > K$ taxa. The output space $\mathcal{Y}$ considered corresponds to the whole set of taxa: $\mathcal{Y} = \{1, ..., T\}$ and its feature representation $\Lambda : \mathcal{Y} \to \mathbb{R}^T$ is defined as:

$$\Lambda(y) = \Big(\mathbf{1}(j \in \mathcal{A}(y)) \Big)_{j=1,...,T},$$

where the function $\mathbf{1}(.)$ is equal to one if its argument is true and zero otherwise, and $\mathcal{A}(y)$ denotes the set of ancestors of the taxon y: it contains the indexes of the taxa found on the path connecting the taxon y to the root of the taxonomy. By convention, we also include y in $\mathcal{A}(y)$: a taxon belongs to its set of ancestors.

In other words, the feature representation $\Lambda(y)$ of the taxon y is a binary vector of length T, in which the entries corresponding to the taxa belonging to the path from the root to y are set to one, the other entries being set to zero. With this representation, the output space $\mathcal{Y}$ can equivalently be seen as the set of *paths* of the taxonomy. As shown in Figure 2.5, the joint input-output representation $\Psi(x, y) = x \otimes \Lambda(y)$ therefore produces vectors of size $p \times T$, and, as in the multiclass case because of the definition of $\Lambda(y)$ and the use of the tensor-product operator, the vector $\Psi(x, y)$ has a block structure. It consists of T "blocks" obtained by T repetitions of the input-space feature representation x, where only the blocks corresponding to the entries of $\Lambda(y)$ which are equal to one are not null.

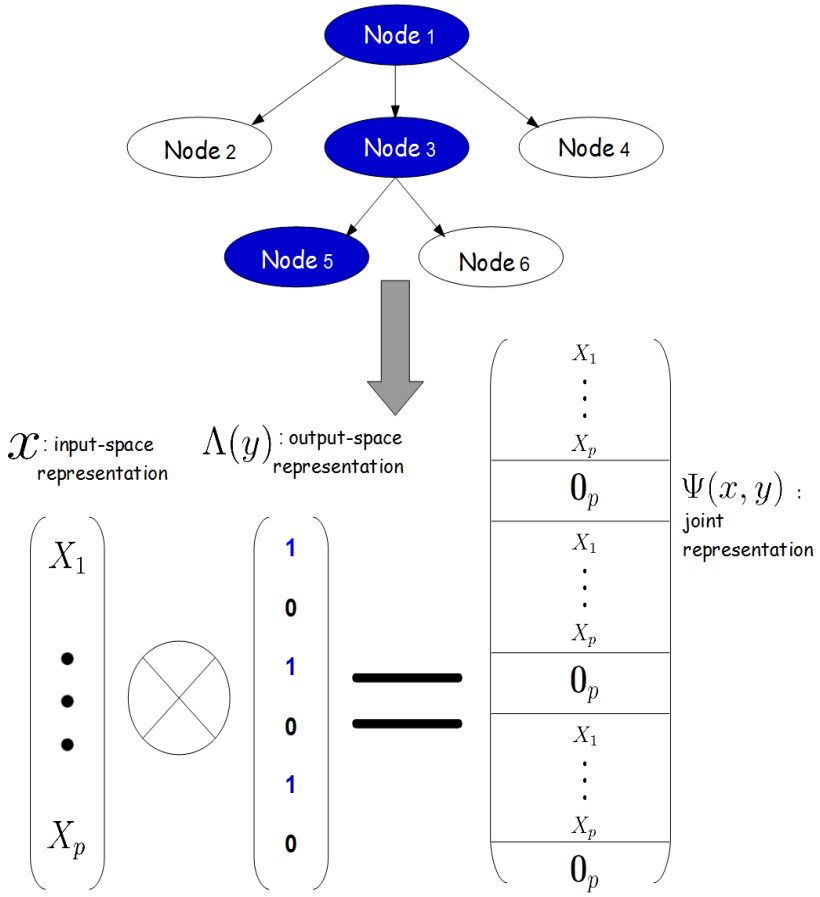

Figure 2.5: **Joint mapping for Structured SVM.** Given a taxonomic tree structure and a couple (x, y), we derive an output-space representation with a binary vector $\Lambda(y)$ where non-zero weights are the taxa presented in the y path (blue). Combined with the input x, we obtain the joint representation $\Psi(x, y)$ by using the tensor product operator. This new represenation has a block structure with copies of x in the blocks corresponding to taxa in y path, and null vectors otherwise.

Coming back to Figure 2.1, we note that nodes with a single child do not bring any additional information than that brought by their child. In practice, they need not to be considered in this joint feature representation and one can resort to a pruned taxonomy in which they are discarded to reduce the computational cost. We emphasize however than the original taxonomy should be used to define the loss function Δ. Last but not least, we need an algorithm to compute:

$$\arg\max_{y \in \mathcal{Y}} \langle W, \Psi(x, y) \rangle \,.$$

Fortunately, in our case this is relatively straightforward. Indeed the cardinality of $\mathcal{Y}$ is finite and relatively small: a large microbial taxonomy may involve at most a few hundred nodes which would remain manageable in a brute-force approach to compute the scores. Moreover, because of the block structure of $\Psi(x, y)$, the scores can be computed recursively in a transverse depth-first-search of the taxonomy. Indeed,

the score of the node $v \in \{1, ..., T\}$ is equal to the score obtained at its parent node plus its own contribution, which is given by $\langle x, w_v \rangle$, if $W = [w_1 w_2 ... w_T]$. As a result, recursively computing the T scores has a complexity barely above $T \times p$, the cost needed to compute the contribution to the overall scores of each individual node.

This approach, which we refer to as `Structured` below, and the cost-sensitive multiclass formulation introduced above have in common to leverage a loss function to take into account the severity of the classification errors. This property can be expected to increase the quality of the predictions made by these algorithms, which will be trained to avoid "severe", that is, high loss, classification errors. As in Section 2.3.1, we define $\Delta(y, y')$ as the length of the shortest path connecting nodes y and y' in the taxonomy, hence directly define the notion of severity as the taxonomic distance.

The hierarchical extension provides however two important differences with respect to the multiclass formulation. First, it works with an enriched joint input-output feature representation in which additional feature variables are shared by output classes (i.e., nodes of the taxonomy) whenever they are on the same path, and the longer they remain on the same path, the more variables they share. This property allows to share information between classes, which we can expect to be beneficial to the training algorithm. Moreover, this extension offers the possibility to modify the nature of the output space itself by considering the whole taxonomy and not the species level only. This property should allow to algorithm to carry out prediction at various taxonomic ranks, and hopefully classify an unknown microorganism at an upper rank than the species one, instead of making an hazardous prediction at the species level. We note however that this latter property is not mandatory and that one can consider leaf nodes only to carry out prediction at the species level.

2.3.3 Cascade approach

The last SVM-based strategy we consider is a divide and conquer approach where a SVM classifier is learned at each internal node of the tree to assign a spectrum to one of its children. A top-down approach is then used to classify a spectrum to a leaf node by this cascade of classifiers [158, 58]. Although any type of classifier can be considered at each node, we chose to rely on SVM in this study. Formally, we focus on classifiers $f(x)$ that are built on SVM-OVA classifiers $w_1, \ldots, w_m$. Training set for a given classifier w_i is made of all examples belonging to a species descending from taxon i. For the predictions of new data, we use the recursive procedure described in [184]:

$$f(x) = \left\{ \begin{array}{c} \textbf{initialize } i := 0 \\ \textbf{while } \text{node } i \text{ has children} \\ i := \arg\max_{j \in \text{children}(i)} w_j^T x \\ \textbf{return } i \end{array} \right\} \tag{2.3}$$

With the definition (2.3), the procedure always leads to predictions at leaves level that corresponds, in our case, to species. Finally, following [18],we consider a variant of this approach in which the tree used to define the cascade is obtained in a preliminary step of unsupervised clustering carried out from species-specific prototypes. We refer to this approach as *Dendrogram-SVMs* (`DSVM`) as opposed to *Cascade-of-Classifiers* (`CoC`) when the original hierarchy is used.

2.3.4 Other benchmarked methods

Finally, we consider three methods not based on SVMs in this benchmark. Random forest (RF) and similarity-based approaches have indeed already been successfully used in the context of MS data classification [54, 163]. We therefore include in the benchmark the RF method described in [32], referred to as `RF`, which consists in learning many decision trees and predicting with a majority vote strategy. We also evaluate two similarity-based approaches: a *1-nearest-neighbour* (`1-NN`) and a *1-nearest-centroid* (`1-Centroid`) approach. In the 1-NN method, a new spectrum is classified in the same class as its closest spectrum in the training set. The same classification rule is applied in the nearest-centroid approach, the centroid of a given species being defined as its median spectrum. These three approaches were described in Section 1.3.

2.4 Experimental setting

We evaluate the classification performance of the various methods by cross-validation, described in Section 1.4.3. To define the cross-validation folds we take into account the strain information. Indeed, the dataset consists of 571 spectra obtained from 213 strains, with in average less than 3 and up to 6 spectra per strain. The variability observed within the replicate spectra of a given strain is purely technical, and is therefore lower than the level of variability that is expected in clinical routine, where an additional level of biological variability is expected due to the fact that the microorganisms to identify differ from that used to learn the classification model. To mimic this setting, hence to avoid optimistic evaluation of classification performance, we therefore affect spectra of a given strain to the same cross-validation fold. In this study, we actually resort to a *leave one strain out* cross-validation strategy in which a single strain is kept aside at each step, thus leading to a 213-fold cross-validation set up.

To assess the classification performance, we primarily consider an accuracy criterion. However, since each species of the dataset is represented by a varying number of strains and each strain by a varying number of spectra, we adopt a nested definition of accuracy criterion, instead of classical proportion of correct classifications. We first define a *strain-level accuracy* as the proportion of spectra that are correctly classified for each strain, and a *species-level accuracy* as the average strain-level accuracy for each species. The overall accuracy indicator is then defined as the average species-level accuracy. In

order to compare the benchmarked approaches, we rely on the two-sample Kolgomorov-Smirnov test [92, 151] applied on vectors of 20 accuracies at the species level.

As can be read from Figure 2.1, this loss can vary in this study from 2 to 12, when a species is respectively mistaken for a species of the same genus or of the other Gram. Because these types of errors are easier to interpret than summary statistics of the tree loss distribution, we report the proportion of errors that fall in the following categories: "within-genus error" ($\Delta = 2$), "outside genus but same Gram error" ($2 < \Delta < 12$), "distinct-Gram error" ($\Delta = 12$).

The regularization (C) parameter of the various SVM formulations considered in this study was optimized within each fold of the leave one strain out cross-validation process by an inner 10 fold cross-validation. As before, spectra of the same strain are systematically affected to the same fold. The grid of candidate values was set to $\{10^{-6}, 10^{-2}, ..., 10^{2}, 10^{6}\}$ and the value was chosen to maximize the nested accuracy indicator defined above. The standard and cascade SVM approaches (`SVM-OVA`, `SVM-OVO`, `CoC` and `DSVM`) were implemented using the `R` package `LiblineaR`[3]. For the two cascade approaches (`CoC` and `DSVM`), one-versus-all classifiers were trained at each internal node of the hierarchy. The tree involved in the `DSVM` method was generated by the `hclust` function of the `R` package `stats`, with a complete linkage clustering method. The `Multiclass` SVM implementation relies on the `C` library `SVM-light` [84]. The cost-sensitive (`TreeLoss`) and `Structured` SVM formulations were implemented based on the `C` library `SVM-struct` [86]. We have relied on the slack-rescaling approach to integrate the loss function Δ in the learning process. We have moreover considered a precision of $\epsilon = 0.1$ on the solution and used the 1-slack algorithm operating in the dual (option `w=3`).

The hyperparameters of the alternative strategies were set from preliminary experiments. We relied on the `R` package `randomForest` to build `RF` models. The number of trees (ntree) and variables per tree (mtry) of the random forest were respectively set to the default value of 500 and to 36, according to the standard heuristics mtry $= \sqrt{p}$. Preliminary experiments revealed that these parameters had little influence on the results as long as they were sufficiently high, especially ntree. Regarding similarity-based methods, the number of neighbours to consider in the nearest neighbours was set to 1 and the Euclidean distance was used. The choice of the distance criterion had little influence on the results but performance decreased when the number of neighbours increased. The Euclidean distance was used for the nearest centroid approach as well.

We note finally that the feature vectors were systematically scaled to unit Euclidean norm.

method	accuracy	# correct	# within-genus	# within-Gram	# distinct-Gram
1-NN	76.8	442	119	6	4
1-Centroid	78.8	445	104	7	15
RF	84.0	494	63	12	2
SVM-OVO	86.6	506	52	13	0
SVM-OVA	88.9	514	50	4	3
Multiclass	88.9	516	47	4	4
Treeloss	89.3	517	47	3	4
Structured	89.4	517	47	4	3
CoC	88.6	505	55	11	0
DSVM	87.1	507	56	2	6

Table 2.2: **Cross-validation results on MicroMass dataset**. This table summarizes the cross-validation results obtained for each benchmarked method. The accuracy measure corresponds to the nested accuracy definition. The four following figures explicitly give the numbers of correct prediction, of *within-genus* errors (for which a species was mistaken for a species of the same genus), of *within-Gram* errors (for which a species was mistaken for a species of another genus of the same Gram) and of *distinct-Gram* errors (for which a species was mistaken for another species of the other Gram). Method names are specified in the main text of section 2.3.

2.5 Results and discussion

The results of the benchmark experiment described in the previous sections are summarized in Table 2.2. Considering the overall accuracy obtained by the various methods, we first note that SVM classifiers, with an accuracy ranging from 86.6 to 89.4%, outperform random forests (accuracy of 84%) and similarity-based approaches (accuracy of 76.8% and 78.8% for the nearest neighbour and nearest centroid approaches respectively). In both cases, these differences are significant (P-value < 0.05). Among the different SVM formulations, we see that the best structured SVM (`Structured`), with an accuracy of 89.4%, outperforms the best "flat" SVMs (`SVM-OVA` and `Multiclass`), which reach an accuracy of 88.9%. This difference, however, is not significant (P-value > 0.05), suggesting that the more elaborate structured SVMs are not particularly useful for this application.

This being said, a closer look at the nature of the misclassifications, given in Table 2.2, reveals some slight differences between the various SVM strategies. We note indeed that while `SVM-OVA`, `Multiclass`, `TreeLoss` and `Structured` make fewer errors than `SVM-OVO` and `CoC` (54 to 57 *versus* 65 to 66), some of these errors involve mistaking a species for a species of the other Gram, which never occur with `SVM-OVO` and `CoC`. This however comes at the price of an increased proportion of errors involving mistaking a species for another one of the same Gram but of another genus, and therefore suggests that a trade-off between the number and the severity of the classification errors can be

[3] http://cran.r-project.org/web/packages/LiblineaR/

achieved. In a similar spirit, we observe some discrepancies between the results provided by the two cascade approaches (`CoC` and `DSVM`): while the two approaches lead to a similar number of classification errors, `DSVM` leads to a higher rate of uncorrect Gram errors for a lower rate of distinct genus but same Gram errors. These two methods only differ in the tree considered, which therefore suggests that its structure has indeed an important role in the learning process and that it could be optimized [152].

Finally, a striking observation that can be made from Table 2.2 is that the great majority of errors involve predicting a species for a species of the same genus, for any considered method. While this makes sense from a biological point of view, this raises at least two hypotheses to explain why the structured methods considered in this benchmark, and in particular those derived from the structured SVM formalism (`TreeLoss` and `Structured`), did not bring any improvement over their "flat" counterparts. First, we note that with a loss function $\Delta(y, y')$ defined as the length of the shortest path between species y and y' in the tree, this type of error is the less penalized one. While this is indeed a natural and relevant definition, it can mainly be expected to limit the number of errors involving remote pairs of species, and hardly to improve over a "flat" strategy that does a limited number of errors of this kind, as this is the case for `SVM-OVA` for instance. On the other hand, it may also be the case that the tree considered in this study is not informative below the genus level. As mentioned previously, the dataset considered in this study involves several pairs or groups of species that are known to be hard to discriminate in general, and by MALDI-TOF MS in particular.

Figure 2.6 shows the counts of the most common types of misclassifications obtained across all the methods considered. It reveals that five pairs or groups of species proved to be particularly challenging: *Bacillus cereus* / *B. thuringiensis*, *Streptococcus mitis* / *S. oralis*, *Enterobacter asburiae* / *E. cloacae*, *Citrobacter braakii* / *C. freundii*, and the group defined by *E. coli* and the three *Shigella* species. It also shows that *E. coli* and *Enterobacter cloacae*, that do not belong to the same genus but both to the *Enterobacteriaceae* family, are relatively often mistaken. The biological proximity within some of these pairs or groups of species may in fact be beyond what can be captured by the MALDI-TOF technology. The *B. cereus* and *B. thuringiensis* species are for instance known to belong to the *Bacillus cereus* group, which is sometimes considered to define a single species [71] and other studies indeed suggest that they cannot be discriminated by MALDI-TOF [100]. *Streptococcus mitis* and *Streptococcus oralis* are also part of similar group comprising more than 99% 16S rRNA similarity [90], and MALDI-TOF mass-spectrometry is known to be hardly able to distinguish them properly [180].

Figure 2.7 illustrates the fact that mass spectra obtained in this study from *B. cereus* and *B. thuringiensis* are hardly distinguishable, at least when they have undergone the process of peak extraction, as opposed to the spectra obtained from *Clostridium difficile* and *C. glycolicum*, that are almost never mistaken one for the other.

To confirm the results obtained at the species level, we evaluate SVM-based methods

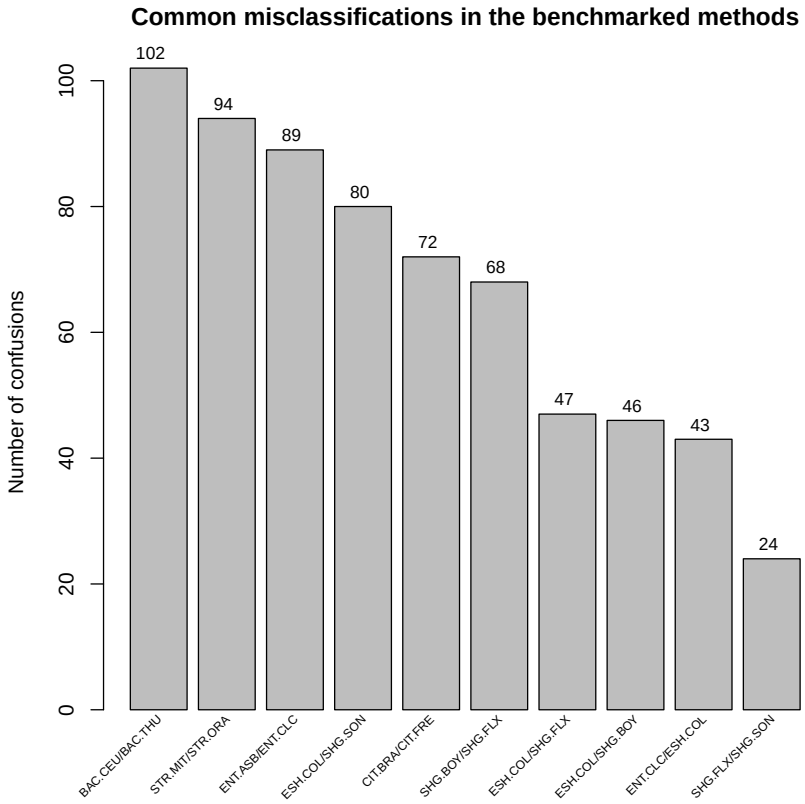

Figure 2.6: **MicroMass dataset: Common classification errors.** Each bar represents one of the most frequent confusions observed across all evaluated classification approaches.

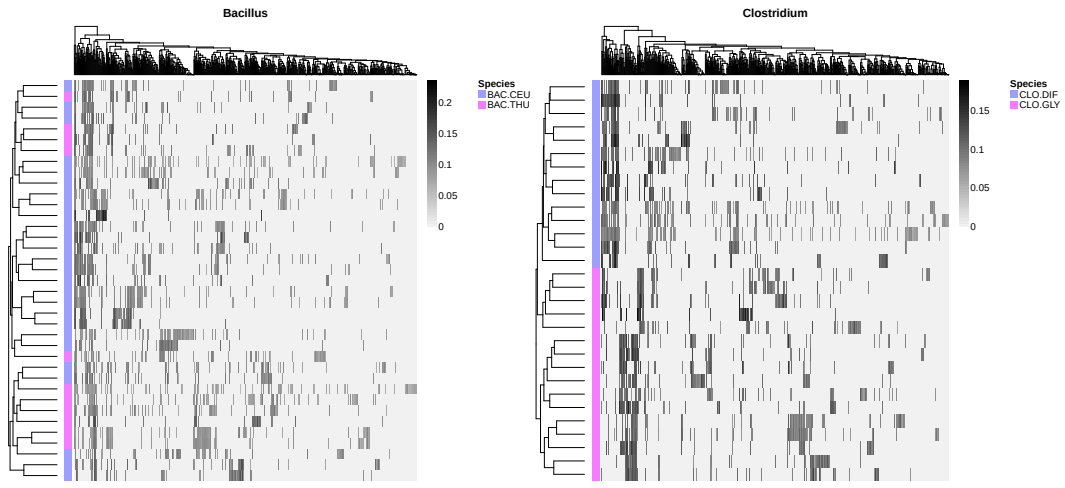

Figure 2.7: **MicroMass dataset: Mass-spectra clustering at the genus level.** Left: *Bacillus* genus. Right: *Clostridium* genus. Mass-spectra (rows) belonging to a given genus are clustered according to their peak lists (columns). For clarity purpose, we removed features equal to zero among all the genus mass-spectra.

Methods	Accuracy(%)	Methods	Accuracy(%)
SVM-OVO	99.1	Treeloss	98.6
SVM-OVA	99.2	CoC	98.9
Multiclass	99.3	DSVM	98.3

Table 2.3: **Performances of benchmarked methods at genus levels**. This table gives strain-level accuracy scors obtained for several benchmarked approaches. Methods are divided in two columns: classical flat methods (left) versus hierarchy-based approaches (right). All method name abbreviations are given in the section 2.3.

at the genus level and put accuracy values in the following Table 2.3. As expected, the prediction performances are near perfect and we better understand what happens a rank above the species level.

2.6 Conclusion

We evaluated several structured methods in the microbial identification context, using mass-spectrometry data. Our results suggest that methods exploiting the underlying bacterial hierarchical structure perform as well as standard "flat" methods. We noted in particular that the majority of classification errors obtained by all the methods considered in this benchmark are within-genus misidentifications. We postulate that the structured methods considered in this benchmark are not tailored to improve flat methods for this type of errors. Unfortunately, a larger panel of strains with a careful definition of the reference identification would be required to validate this hypothesis. [184] recently proposed a structured regularization method specifically designed to cope with this issue, and it would therefore be interesting to evaluate its relevance in this context.

Chapter 3

On learning matrices with orthogonal columns or disjoint supports

This chapter has been published in a slightly different form in [171], as joint work with Pierre Mahé, Alexandre d'Aspremont, Jean-Baptiste Veyrieras, and Jean-Philippe Vert.

Abstract

We investigate new matrix penalties to jointly learn linear models with orthogonality constraints, generalizing the work of Xiao et al. [184] who proposed a strictly convex matrix norm for orthogonal transfer. We show that this norm converges to a particular atomic norm when its convexity parameter decreases, leading to new algorithmic solutions to minimize it. We also investigate concave formulations of this norm, corresponding to more aggressive strategies to induce orthogonality, and show how these penalties can also be used to learn sparse models with disjoint supports. We evaluate these approaches on synthetic and real datasets.

Résumé

Nous nous intéressons à de nouvelles pénalités matricielles pour apprendre conjointement des modèles linéaires avec des contraintes d'orthogonalité, généralisant ainsi les travaux de Xiao et al. [184] qui proposent une norme strictment convexe pour induire du transfert orthogonal. Nous montrons que cette fonction converge vers une norme atomique particulière lorsque son paramètre de convexité décroît. Cette équivalence avec une norme atomique donne accès à de nouvelles solutions algorithmiques pour la minimiser. En continuant à décroître la convexité de la norme, nous avons également évalué des formulations concaves

qui correspondent à des stratégies plus agressives pour obtenir de l'orthogonalité. Nous proposons une extension naturelle de cette approche au cas où nous souhaitons apprendre des modèles parcimonieux avec des supports disjoints. Ces approches sont évaluées sur des données simulées, ainsi que des jeux de données réelles.

3.1 Introduction

Learning several models simultaneously instead of separately, a framework often referred to as multitask or transfer learning, is a powerful setting to leverage information across related but different problems [40, 164, 16, 11, 61]. In particular it has been empirically shown that when different tasks share some similarity, such as learning binding models for similar proteins [82], predicting exams score for students of different schools [11, 61] or learning models for semantically related concepts in a hierarchy [116, 38], jointly learning the models with a multitask strategy leads to better performance. In all aforementioned examples (and many others), the underlying assumption is that the tasks share some similarity, and the multitask strategies exploit this assumption by, e.g., imposing shared parameters estimated jointly across the tasks, or penalizing differences between the models learned in the tasks.

Alternatively, in some situations we would like to solve different tasks under the opposite assumption, namely, that the models are *different*, e.g., that they use different features or should be orthogonal to each other. This is the case for example when we want to learn unrelated tasks, such as recognizing the identity and the emotion of a person on a picture, where we know from literature that these two recognition problems depend on different and uncorrelated features of the same image [39, 139]. In structured learning such as classification in a hierarchical taxonomy, it has been proposed to learn local models at each node of the hierarchy and to encourage the classifier at each node to be different from the classifiers at its ancestors, in order to better reflect the natural coarse-to-fine nature of the classifiers at different levels of the hierarchy [184, 64]. Several approaches have been proposed recently to learn such different models. [184] proposed to penalize a weighted ℓ_1 norm of the off-diagonal entries of the covariance matrix between the tasks, in order to promote sparsity of inner products hence orthogonality between tasks; however some extra ridge term must be added in order to make the penalty convex and amenable to efficient optimization, leading to potentially unwanted over-regularization. [139] proposed also a convex penalty to learn two groups of tasks based on orthogonal subspaces; again, due to the non-convex nature of the norm applied to inner products between vectors, an extra ridge term is needed to make the penalty convex. Finally, [64] proposed a method to learn a tree of metrics, enforcing disjoint sparsity between the different metrics. The convex penalty of [64], though, only promotes sparsity for non-negative vectors, such as the diagonals of metric matrices, and can not easily be extended to enforce disjoint sparsity on general vectors.

In this work, we extend the work of [184] in two directions. First, we investigate generalization of the penalty proposed by [184] when we decrease its convexity, in order to make it more "aggressive" in promoting orthogonality. Our main findings can be visualized in Figure 3.1, which shows the level sets of penalties we consider. Starting from the strictly convex penalty of [184], corresponding to a strictly convex unit ball with singularities at matrices with orthogonal columns (left), we show that by reducing its convexity it converges to a convex atomic norm [41], whose unit ball is the convex hull of the singularities of the first ball. This shows that for particular choices of parameters the penalty of [184] is "optimal" to learn matrices with pairwise orthogonal columns, in the sense that it is the tightest convex function which is equal to the Frobenius norm on the subset of matrices that we are interested in. This observation has also algorithmic consequences: while [184] propose an optimization scheme that only works when the penalty is strictly convex, we show that the dual norm in the limit case of the atomic norm can be estimated efficiently by solving a small semidefinite program (SDP), leading to new algorithmic solutions to use this norm as regularizer in a learning problem. We also propose and investigate empirically more concave extensions of this norm in order to increase the propensity to learn matrices with orthogonal columns (right). Our second extension is to show how these penalties can be modified to learn sparse models with disjoint supports, a particular case of orthogonal models which is relevant when different tasks are know to involve different features.

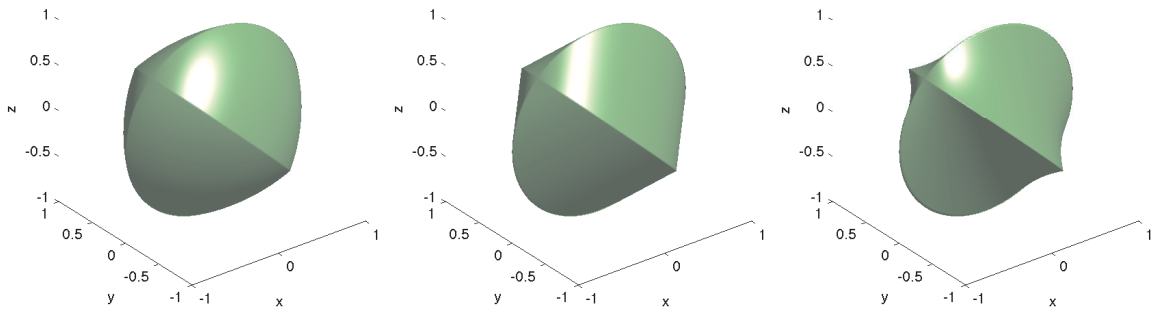

Figure 3.1: Level sets of the penalty Ω_K defined in (3.2) for 2-by-2 symmetric matrices parametrized as $\begin{pmatrix} x & y \\ y & z \end{pmatrix}$, when $K = \begin{pmatrix} \gamma & 1 \\ 1 & \gamma \end{pmatrix}$ and we vary γ from $\gamma = 2$ (left), which corresponds to a strictly convex penalty proposed by [184], to $\gamma = 1$ (center), which is a limit case where the penalty is convex but not strictly convex and turns out to be an atomic norm (Theorem 3.2.1), and to $\gamma = 1/2$ (right), which corresponds to a non convex penalty.

3.2 An atomic norm to learn matrices with orthogonal columns

We consider the problem of learning a $d \times T$ matrix $W = (w_1, \dots, w_T)$, where each column w_i is a d-dimensional vector corresponding to a task such as a linear classification model at a node of a taxonomy. We call such a matrix *scaled orthogonal* if $W^\top W$ is diagonal, i.e., if all columns of W are orthogonal to each other, and denote by $\mathcal{O}$ the set of $d \times T$ scaled orthogonal matrices. Note that this should not be confused with the stronger concept of orthogonal matrix often used in mathematics, which means that W is square and $W^\top W$ is the identity, i.e., that the columns form an orthonormal basis.

A general approach to estimate W from observations is to formulate the inference as an optimization problem:

$$\min_W f(W) + \frac{\lambda}{2} \Omega(W)^2 \,, \tag{3.1}$$

where $f(W)$ is an empirical risk which measures the fit to data like those described in Section 1.2.1, $\Omega(W)$ is a penalty that enforces some constraints on the solution such as sparseness or low-rankness, and $\lambda > 0$ is a parameter adjusting the tradeoff between these two objectives. When $f(W)$ and $\Omega(W)$ are convex functions, then (3.1) is a convex optimization problem that can often be solved efficiently and lead to a unique solution. Classical examples of penalties $\Omega(W)$ include the ℓ_1 norm to promote sparsity in W [165], the nuclear norm to learn low-rank matrices [156], and the ℓ_1/ℓ_2 norm to perform joint feature selection across tasks [125].

Suppose we know that some or all of the columns of W should be orthogonal to each other. [184] proposed an orthogonal regularizer of the form $\sum_{i,j} K_{ij} |w_i^\top w_j|$, where K_{ij} is a non-negative weight to enforce more or less the orthogonality between w_i and w_j. This is however not a convex function of W, and [184] propose to define a convex penalty by adding ridge terms to this regularizer, namely:

$$\Omega_K(W)^2 = \sum_{i=1}^{T} K_{ii} ||w_i||^2 + \sum_{i \neq j} K_{ij} |w_i^\top w_j| \,, \tag{3.2}$$

where K is an hyperparameter matrix representing structure among different models. [184] give a sufficient condition on K to ensure that (3.2) is convex, but there remains a lot of freedom in the choice of K.

Let us consider the case where we choose $K_{ii} = 1$ and $K_{ij} > 0$ in (3.2). Then we see that for scaled orthogonal matrices $W \in \mathcal{O}$ the penalty (3.2) boils down to the Frobenius norm:

$$\forall W \in \mathcal{O} \,, \quad \Omega_K(W)^2 = \sum_{i=1}^{T} ||w_i||^2 = ||W||_F^2 \,.$$

The extra terms $K_{ij} |w_i^\top w_j|$ in (3.2) ensure that, in addition, the penalty is not differentiable at scaled orthogonal matrices, allowing under some conditions the recovery of

such matrices when (3.2) is plugged into (3.1) [10, 41].

There are however many penalties, including (3.2), that are convex, singular on $\mathcal{O}$ and which equal the Frobenius norm in $\mathcal{O}$. Among them, we propose to consider the *tightest* one, namely, the atomic norm in the sense of [41] induced by the set of atoms $\mathcal{A} = \{W \in \mathcal{O} : ||W||_F = 1\}$. This norm, which we denote below by $\Omega_{\mathcal{O}}(X)$ for any $d \times T$ matrix X, can be expressed as

$$\Omega_{\mathcal{O}}(X) = \inf \left\{ \sum_{Y \in \mathcal{A}} \lambda_Y : X = \sum_{Y \in \mathcal{A}} \lambda_Y Y,\ \lambda_Y \geq 0 \right\} . \tag{3.3}$$

In other words, this last expression writes $\Omega_{\mathcal{O}}(X)$ as the ℓ_1 norm of the vector of coefficients λ in a decomposition of X into atoms, namely, scaled orthogonal matrices of unit Frobenius norms. Plugging (3.3) into (3.1) provides a convex problem to infer an atom, or a sparse combination of atoms. Note that, contrary to Ω_K (3.2), $\Omega_{\mathcal{O}}$ is always convex without technical conditions. In addition, since both norms are equal on the atoms $\mathcal{A}$, the tangent cone of $\Omega_{\mathcal{O}}$ at any scaled orthogonal matrix $W \in \mathcal{O}$ is contained in the tangent cone of Ω_K at the same point, suggesting that the recovery and inference of a scaled orthogonal matrix through the convex procedure (3.1) is easier with $\Omega_{\mathcal{O}}$ than with Ω_K [41].

The following result shows that, surprisingly, the norms Ω_K with adequate weights and $\Omega_{\mathcal{O}}$ coincide on matrices with two columns. This theorem is illustrated in Figure 3.1, where we show the unit ball of Ω_K when we change K. The ball at the center corresponds to a limit situation where Ω_K is still convex, but not strictly convex anymore. We see in this picture that the ball can equivalently be defined as the convex hull of two circles, which correspond precisely to the set of matrices with orthogonal columns and unit Frobenius norm; i.e., that Ω_K in this case is precisely the atomic norm induced by these atoms.

Theorem 3.2.1. *For any $d \geq 1$ and any $d \times 2$ matrix $W = (w_1, w_2)$, it holds that:*

$$\Omega_{\mathcal{O}}(W) = \Omega_K(W) , \tag{3.4}$$

with

$$K = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} . \tag{3.5}$$

Proof. Since K in (3.5) is entry-wise non-negative, and since the companion matrix

$$\bar{K} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

is positive semidefinite, we know from [184, Theorem 1] that Ω_K^2 is convex in this case. Since (3.4) obviously holds for $W \in \mathcal{O}$, and since $\Omega_{\mathcal{O}}$ is the tightest convex function such that (3.4) holds on $\mathcal{O}$, we directly get that $\Omega_{\mathcal{O}}(W) \leq \Omega_K(W)$ for any

$W \in \mathbb{R}^{d\times 2}$. To prove the converse inequality, it suffices to find, for any $W \in \mathbb{R}^{d\times 2}$, a decomposition of the form $W = \lambda U + (1-\lambda)V$, with $U, V \in \mathcal{O}$, $\lambda \in [0,1]$, such that $\Omega_K(U) = \Omega_K(V) = \Omega_K(W)$. Geometrically, this would mean that any point on the unit ball of Ω_K lies on a straight segment that connects two atoms on this ball, meaning that the unit ball of Ω_K is precisely the convex hull of the unit ball restricted to the atoms. The following lemma, which can be proved by direct calculation, shows that this is indeed possible by explicitly providing such a decomposition.

Lemma 3.2.2. *For any $W = (w_1, w_2) \in \mathbb{R}^{d\times 2}$, let:*

- *if $w_1^\top w_2 \geq 0$, $U = (w_1 + w_2, 0)$ and $V = \left(w_1 - \frac{w_1^\top w_2}{\| w_2 \|^2} w_2, \left(1 + \frac{w_1^\top w_2}{\| w_2 \|^2}\right) w_2\right)$,*
- *if $w_1^\top w_2 < 0$, $U = (w_1 - w_2, 0)$ and $V = \left(w_1 - \frac{w_1^\top w_2}{\| w_2 \|^2} w_2, \left(1 - \frac{w_1^\top w_2}{\| w_2 \|^2}\right) w_2\right)$,*

and let $\lambda = \frac{\left| w_1^\top w_2 \right|}{\left| w_1^\top w_2 \right| + \| w_2 \|^2}$. Then it holds that:

- *$U, V \in \mathcal{O}$,*
- *$\lambda \in [0,1]$ and $W = \lambda U + (1-\lambda)V$,*
- *$\Omega_K(W) = \Omega_K(U) = \Omega_K(V)$.* ∎

Theorem 3.2.1 can be easily generalized (with a different set of atoms) when K is any 2-by-2 symmetric, positive semidefinite matrix with non-negative entries and with 0 as eigenvalue, corresponding to the limit case where Ω_K is convex but not strictly convex: it is then always an atomic norm. The extension of Theorem 3.2.1 to more than 2 columns, however, is not true. Atoms of $\Omega_\mathcal{O}$ are matrices with *all* columns orthogonal to each other, so using $\Omega_\mathcal{O}$ as a penalty on matrices with $T > 2$ columns may either lead to such an atom, or to a sparse linear combination of atoms, which would in general have no pair of column orthogonal to each other. The following theorem, which is a simple consequence of Theorem 3.2.1, shows that for some choices of K in the $T > 2$ case, the penalty Ω_K can be written as a sum of $\Omega_\mathcal{O}$ that penalizes pairs of columns.

Theorem 3.2.3. *For any $T \geq 2$, let K be a symmetric T-by-T matrix with non-negative entries and such that, for any $i = 1, \ldots, T$,*

$$\forall i = 1, \ldots, T \quad K_{ii} = \sum_{j \neq i} K_{ij}\,.$$

Then, for any $d \geq 1$ and any $d \times T$ matrix $W = (w_1, \ldots, w_T)$, it holds that:

$$\Omega_K(W) = \sum_{i<j} K_{ij} \Omega_\mathcal{O}((w_i, w_j))\,,$$

where $(w_i, w_j) \in \mathbb{R}^{d\times 2}$ is the matrix with columns w_i and w_j.

Proof. Let $A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$. By Theorem 3.2.1, we know that $\Omega_A((w_i, w_j)) = \Omega_{\mathcal{O}}((w_i, w_j))$ for al $i \neq j$, therefore:

$$\begin{aligned}
\sum_{i<j} K_{ij}\Omega_{\mathcal{O}}((w_i, w_j)) &= \sum_{i<j} K_{ij}\Omega_A((w_i, w_j)) \\
&= \sum_{i<j} K_{ij}\left(||w_i||^2 + \| w_j \|^2 + 2|w_i^\top w_j|\right) \\
&= \sum_{i=1}^{T}\left(\sum_{j\neq i} K_{ij}\right)\| w_i \|^2 + \sum_{i\neq j}|w_i^\top w_j| \\
&= \Omega_K(W)\,.
\end{aligned}$$

■

3.3 The dual of the atomic norm

In this section we consider the atomic norm $\Omega_{\mathcal{O}}$ for matrices with 2 columns, and show that we can efficiently compute its dual and a subgradient of its dual by solving a 6-dimensional SDP. This can be useful to provide simple duality gaps and stopping criteria to learn with convex but not strictly convex penalties Ω_K, which are in particular not amenable to optimization with the method of [184].

Remember that the dual of a norm $\Omega(X)$ is

$$\Omega^*(X) = \sup_{Y\ :\ \Omega(Y)\leq 1} \mathbf{Tr}(X^\top Y)\,.$$

Since $\Omega_{\mathcal{O}}$ is an atomic norm induced by the atom set $\mathcal{A}$, its dual satisfies [41]:

$$\Omega^*_{\mathcal{O}}(X) = \sup_{Y\in\mathcal{A}} \mathbf{Tr}(X^\top Y)\,, \tag{3.6}$$

and in addition any atom $Y \in \mathcal{A}$ which achieves the maximum in (3.6) is a subgradient of $\Omega^*_{\mathcal{O}}$ at X. We now show that computing $\Omega^*_{\mathcal{O}}(X)$ and a subgradient can be done efficiently:

Theorem 3.3.1. *For any $d \geq 1$ and $X \in \mathbb{R}^{d\times 2}$, a solution to*

$$\Omega^*_{\mathcal{O}}(X) = \sup_{Y\in\mathcal{A}} \mathbf{Tr}(X^\top Y) \tag{3.7}$$

can be obtained from the solution of a SDP over matrices of size 6×6.

Proof. From the definition of $\mathcal{A}$ we can reformulate (3.7) as:

$$\begin{aligned}
\Omega^*_{\mathcal{O}}(X) = \quad & \text{maximize} && \mathbf{Tr}(Y^\top X) \\
& \text{subject to} && Y^\top Y \text{ diagonal} \\
& && \|Y\|_F = 1,
\end{aligned}$$

in the variable $Y \in \mathbb{R}^{d\times 2}$. Because $-Y$ is a feasible point whenever Y is, this problem

is equivalent to

$$\begin{array}{rll} \Omega^*_{\mathcal{O}}(X)^2 = & \text{maximize} & \mathbf{Tr}(Y^\top X)^2 \\ & \text{subject to} & Y^\top Y \text{ diagonal} \\ & & \|Y\|_F = 1, \end{array} \tag{3.8}$$

which is a *non-convex* quadratic program in Y. We first reformulate this problem in vector terms and write $z = \mathbf{vec}(Y) \in \mathbb{R}^{2d}$, so that $z^\top = (z_1^\top, z_2^\top)$ with $z_1 = Y_1$ and $z_2 = Y_2$. Problem (3.8) becomes

$$\begin{array}{ll} \text{maximize} & (X_1^\top z_1 + X_2^\top z_2)^2 \\ \text{subject to} & z_1^\top z_2 = 0 \\ & \|z_1\|_2^2 + \|z_2\|_2^2 = 1, \end{array}$$

which is again

$$\begin{array}{ll} \text{maximize} & (\mathbf{vec}(X)^\top z)^2 \\ \text{subject to} & z^\top \begin{pmatrix} 0 & \mathbf{I} \\ \mathbf{I} & 0 \end{pmatrix} z = 0 \\ & z^\top z = 1\,. \end{array}$$

Following the classical lifting technique derived by [149, 105], we can produce a semidefinite relaxation of this last problem by changing variables, setting $Z = zz^\top$, and dropping the implicit rank constraint on Z, to get

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}\left(\mathbf{vec}(X)\,\mathbf{vec}(X)^\top Z\right) \\ \text{subject to} & \mathbf{Tr}\left(\begin{pmatrix} 0 & \mathbf{I} \\ \mathbf{I} & 0 \end{pmatrix} Z\right) = 0 \\ & \mathbf{Tr}(Z) = 1,\ Z \succeq 0, \end{array} \tag{3.9}$$

which is a SDP in the matrix variable $Z \in \mathbf{S}_{2d}$. The quadratic convexity results of [35] (see also [14], §II.14), also known as the $\mathcal{S}$-procedure or Brickman's theorem, tells us that the optimal value of the semidefinite program (3.9) is equal to the optimal value of the non-convex quadratic problem (QP) in (3.8), and a solution Y to (3.8) can be constructed from an optimal solution Z of (3.9) (see, e.g., [29] App. B.3 for an explicit recursive procedure).

Problem (3.9) is an SDP over $2d \times 2d$ matrices, which can be prohibitive in practice as soon as d gets large. Let us now show that a simple decomposition allows to reformulate the problem as a SDP of fixed dimension 6. We can compute the QR decomposition of X written $X = QR_2$ where $Q \in \mathbb{R}^{d\times d}$ is an orthogonal matrix and $R_2 \in \mathbb{R}^{d\times 2}$ with $R_2 = (R^\top, 0)^\top$ where $R \in \mathbb{R}^{2\times 2}$ is an upper triangular matrix. This means that without loss of generality, the original problem of computing $\Omega^*(X)$ can be rewritten

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(Y^T Q R_2) \\ \text{subject to} & Y^T Q Q^T Y \text{ diagonal} \\ & \|Q^T Y\|_F = 1\,, \end{array} \tag{3.10}$$

which is equivalent to

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(Y^T R_2) \\ \text{subject to} & Y^T Y \text{ diagonal} \\ & \|Y\|_F = 1\,, \end{array}$$

in the variable $Y \in \mathbb{R}^{d\times 2}$. This means that we can always assume that X is block upper diagonal with lower block equal to zero. This program can be rewritten

$$\begin{array}{ll} \text{maximize} & (\mathbf{vec}(R_2)^T z)^2 \\ \text{subject to} & z^T \begin{pmatrix} 0 & \mathbf{I}_d \\ \mathbf{I}_d & 0 \end{pmatrix} z = 0 \\ & z^T z = 1\,, \end{array}$$

in the variable $z = \mathbf{vec}(Y) \in \mathbb{R}^{2d}$. Now notice that

$$\begin{pmatrix} 0 & \mathbf{I}_d \\ \mathbf{I}_d & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \mathbf{I}_d = (P^T \,\mathbf{diag}(-1,1) P) \otimes \mathbf{I}_d\,,$$

where $P = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}$ is an orthogonal matrix. Let us write $S = P \otimes \mathbf{I}_d$ (also an orthogonal matrix), $w = Sz$ and $b = S\,\mathbf{vec}(R_2)$, we can rewrite the QP above as

$$\begin{array}{ll} \text{maximize} & (\mathbf{vec}(R_2)^T S^T w)^2 \\ \text{subject to} & w^T \begin{pmatrix} -\mathbf{I}_d & 0 \\ 0 & \mathbf{I}_d \end{pmatrix} w = 0 \\ & w^T w = 1\,, \end{array}$$

in the variable $w \in \mathbb{R}^{2d}$. Now $b = S\,\mathbf{vec}(R_2)$ means

$$b = (P \otimes \mathbf{I}_d)\,\mathbf{vec}(R_2) = \mathbf{vec}(R_2 P)\,,$$

so if $R_2 = (R^T, 0)^T$ where $R \in \mathbb{R}^{T\times T}$ as above, then $b = \mathbf{vec}((P^T R^T, 0)^T)$ hence the b has only four nonzero coefficients at indices $J = \{1, 2, d+1, d+2\}$. This means that the QP can be reformatted as

$$\begin{array}{ll} \text{maximize} & w_J^T (b_J b_J^T) w_J \\ \text{subject to} & w_J^T \begin{pmatrix} -\mathbf{I}_2 & 0 \\ 0 & \mathbf{I}_2 \end{pmatrix} w_J = y_1^T y_1 - y_2^T y_2 \\ & w_J^T w_J + y_1^T y_1 + y_2^T y_2 = 1\,, \end{array}$$

in the variables $w_J \in \mathbb{R}^4$ and $y_1, y_2 \in \mathbb{R}^{d-2}$, where we have defined $z_1^T = (w_3, \ldots, w_d)$ and $z_1^T = (w_{d+3}, \ldots, w_{2d})$. By symmetry, we can assume, without loss of generality, that the coefficients of the vectors y_1 and y_2 are uniformly equal to scalars $y_1, y_2 \in \mathbb{R}$,

so the last problem is equivalent to

$$\begin{aligned} \text{maximize} \quad & w_J^T (b_J b_J^T) w_J \\ \text{subject to} \quad & w_J^T \begin{pmatrix} -\mathbf{I}_2 & 0 \\ 0 & \mathbf{I}_2 \end{pmatrix} w_J = (d-2)y_1^2 - (d-2)y_2^2 \\ & w_J^T w_J + (d-2)y_1^2 + (d-2)y_2^2 = 1\,, \end{aligned}$$

which is now a QP of dimension 6 in the variables $w_J \in \mathbb{R}^4$ and $y_1, y_2 \in \mathbb{R}$. This last problem can then be lifted as above, to become

$$\begin{aligned} \text{maximize} \quad & \mathbf{Tr}\, W \begin{pmatrix} b_J b_J^T & 0 \\ 0 & 0 \end{pmatrix} \\ \text{subject to} \quad & \mathbf{Tr}\, W \begin{pmatrix} -\mathbf{I}_2 & 0 & 0 \\ 0 & \mathbf{I}_2 & 0 \\ 0 & 0 & \mathbf{diag}(-(d-2),(d-2)) \end{pmatrix} = 0 \\ & \mathbf{Tr}\, W \begin{pmatrix} \mathbf{I}_4 & 0 \\ 0 & (d-2)\mathbf{I}_2 \end{pmatrix} = 1,\ W \succeq 0, \end{aligned} \tag{3.11}$$

which is a semidefinite program in the variable $W \in \mathbf{S}_6$. The optimal values of programs (3.10) and (3.11) are equal and a solution to (3.10) can be constructed from an optimal solution to (3.11). Because (3.11) is a semidefinite program of fixed dimension 6, it can be solved efficiently *independently of the dimension d.* All we need is the QR decomposition of X which can be formed with cost $O(d)$ when $X \in \mathbb{R}^{d\times 2}$. ∎

3.4 Algorithms

In order to learn with the penalty Ω_K we need to solve problems of the form

$$\min_W f(W) + \frac{\lambda}{2}\,\Omega_K(W)^2\,. \tag{3.12}$$

When Ω_K is strictly convex, [184] propose a regularized dual averaging (RDA) method based on subgradient descent, and show that a subgradient of $\Omega_K(W)$ in that case is given by $G = (g_1, \ldots, g_t)$ where

$$g_i = K_{ii} w_i + \sum_{j\neq i} \operatorname{sign}\left(w_i^\top w_j\right) K_{ij} w_j\,, \tag{3.13}$$

with the convention $\operatorname{sign}(0) = 0$. When Ω_K is not strictly convex, e.g., when it is a sum of atomic norms as in Theorem 3.2.3 or when it is not even convex (as on the right-hand plot of Figure 3.1), the RDA methods can not be used anymore. In that case, we propose to use a classical subgradient descent scheme, using the subgradient (3.13), and a step size decreasing with $t^{-1/2}$ where t is the iteration. Note that [184] only prove that (3.13) is a valid subgradient when Ω_K is convex; we keep the same formula in the general case since Ω_K is differentiable almost everywhere. In the non-

convex case, subgradient descent will converge to a stationary point, so one may run it several times with random initializations before taking the best solution. In the experiments below, we always run subgradient descent starting from the null matrix, and observed empirically that it often leads to a good solution compared to multiple random initializations.

Let us now discuss another possible optimization scheme when K satisfies the conditions of Theorem 3.2.3, i.e., when the penalty is a linear combination of nuclear norms over pairs of columns. In that case, by Theorem 3.2.3 the optimization problem has the form:

$$\min_W f(W) + \frac{\lambda}{2} \sum_{i<j} K_{ij} \Omega_{\mathcal{O}}((w_i, w_j))^2 \,. \tag{3.14}$$

We can then write an equivalent dual problem amenable to optimization. Let us first consider the simple case of $T = 2$ columns, in which case (3.14) boils down to

$$\min_W \left\{ f(W) + \frac{\lambda}{2} \Omega^2_{\mathcal{O}}(W) \right\} \tag{3.15}$$

in the variable $W \in \mathbb{R}^{d \times 2}$. Remember that for any norm, if $h(x) = \|x\|^2/2$ then the Fenchel dual of h is $h^*(y) = \|y\|_*^2/2$ [29, §3.3.1]). Then [25, Th. 3.3.5] shows that the dual of (3.15) is written

$$\sup_Z \left\{ -f^*(Z) - \frac{1}{2\lambda} \left(\Omega^*_{\mathcal{O}}(Z)\right)^2 \right\} \tag{3.16}$$

in the variable $Z \in \mathbb{R}^{d \times 2}$. Under mild technical conditions, the optimal values of both problems are equal. Back to the general case (3.14), note that the conjugate of the function $\Omega_{\mathcal{O}}((w_i, w_j))$, which we write $\tilde{\Omega}^*_{ij}(W)$, is given by

$$\tilde{\Omega}^*_{ij}(W) = \begin{cases} \Omega^*_{\mathcal{O}}((W_i, W_j)) & \text{if } W_l = 0 \text{ for } l \neq i, j \\ +\infty & \text{otherwise.} \end{cases}$$

Then, using the following inf-convolution result [138, Th. 16.4]:

$$(f_1 + \ldots + f_s)^*(y) = \inf_{y_1, \ldots, y_s} \{ f_1^*(y_1) + \ldots + f_s^*(y_s) : y_1 + \ldots + y_s = y \},$$

we obtain that the Fenchel dual of problem (3.14) is written

$$\sup_Z \left\{ -f\left(\sum_{i<j} Z_{ij} \right) - \sum_{i<j} \frac{1}{2\lambda K_{ij}} \tilde{\Omega}^*_{ij}(Z_k)^2 \right\} \tag{3.17}$$

in the variables $(Z_{ij})_{i<j} \in \mathbb{R}^{d \times T}$. Note that the definitions of $\tilde{\Omega}^*_{ij}$ mean that each Z_{ij} only has two nonzero columns at positions i and j. Now, note that by Theorem 3.3.1, the function to be optimized in (3.17) can be efficiently estimated and a subgradient can be computed. Any value of (3.17) provides a lower bound to (3.14), thus giving

a duality gap that can be used to monitor convergence of the subgradient descent method.

3.5 Learning disjoint supports

An interesting particular case of learning orthogonal vectors is the situation where we seek sparse vectors with disjoint supports. In this section we briefly discuss how Ω_K can help in this situation, too. For simplicity we only discuss the case of $T = 2$ vectors, an extension to the general case being straightforward. The matrix $W \in \mathbb{R}^{d\times 2}$ has columns with complementary supports if, for $i = 1, \ldots, d$,

$$W_{1,i} \neq 0 \implies W_{2,i} = 0 \text{ and } W_{2,i} \neq 0 \implies W_{1,i} = 0\,,$$

or in other words $W_1 \circ W_2 = 0$ where $\circ$ denotes the Hadamard (entrywise) product of matrices. If we denote by $|W|$ the matrix whose entries are the absolute values of the entries of W, then we further observe that $|W_1 \circ W_2| = |W_1| \circ |W_2|$, so $W_1 \circ W_2 = 0$ if and only if $|W_1| \circ |W_2| = 0$. Interestingly, if $V \in \mathbb{R}^{d\times 2}$ is a matrix with non-negative entries, then $V_1 \circ V_2 = 0$ is equivalent to $V_1^T V_2 = 0$; this shows that W has columns with complementary supports if and only if $|W_1|$ and $|W_2|$ are orthogonal.

This suggest a general way to learn a matrix with disjoint supports, by solving a problem of the form:

$$\min_W f(W) + \frac{\lambda}{2}\,\Omega_K(|W|)^2\,, \tag{3.18}$$

where Ω_K is a penalty that induces orthogonality among columns. To solve (3.18), we introduce a non-negative matrix V such that $-V \leq W \leq V$ (where $\leq$ refers to element-wise comparisons), and solve the following problem:

$$\min_{-V\leq W\leq V} f(W) + \frac{\lambda}{2}\,\Omega_K(V)^2\,. \tag{3.19}$$

At the optimum of (3.19), we have $V = |W|$ which shows that (3.19) is indeed equivalent to (3.18). Since a subgradient of (3.19) in (V, W) can easily be computed, we propose to solve (3.19) by a projected subgradient scheme, where at each iteration we update V and W along a subgradient, and then project the new point to the constraint set $-V \leq W \leq V$ and $V \geq 0$.

3.6 Experiments

In this section, we present numerical experiments on two simulated datasets. We benchmark the following methods:

- Xiao: this is the method described in [184] where we solve (3.1) with the penalty (3.2). We consider both convex and non-convex versions, by changing the matrix K

in (3.2).

- Disjoint Supports: this is the approach where we solve (3.18), with non-convex and convex versions.
- Ridge Regression: this standard method corresponds to learning the tasks independently by ridge regression and is described in Section 1.2.4.
- LASSO: this is the classical approach inducing sparsity over all tasks, without sharing information across the tasks and is described in Section 1.2.4.

In all experiments involving Ω_K, we consider a symmetric matrix K parametrized by its diagonal value γ,

$$K = \begin{pmatrix} \gamma & & 1 \\ & \ddots & \\ 1 & & \gamma \end{pmatrix} . \tag{3.20}$$

Based on the conditions for the convexity of Ω_K studied by [184], we control the convexity of Ω_K used in the Xiao and Disjoint Supports approaches with the following rule on γ:

- $\gamma > T - 1$ leads to a strictly convex Ω_K function as described in [184],
- $\gamma = T - 1$ is the the limit case where Ω_K satisfies the conditions of Theorem 3.2.3, i.e., where it is a sum of atomic norms over pairs of columns:

$$\Omega_K(W) = \sum_{i<j} \Omega_{\mathcal{O}}((w_i, w_j)) , \tag{3.21}$$

- $\gamma < T - 1$ corresponds to the case where Ω_K is not convex.

We test the different methods on regression problem where, given a matrix of covariates $X \in \mathbb{R}^{n \times d}$ and a matrix of T response variables $Y \in \mathbb{R}^{n \times T}$, we seek to minimize the squared error $f(W) = \|Y - XW\|^2$.

3.6.1 The effect of convexity

We use simulated data to test whether theoretical differences between $\Omega_K, \Omega_{\mathcal{O}}$ and concave formulations have an impact on analytical performances. In particular, by playing with γ in (3.20), we investigate to what extent the convexity constraint imposed by [184] is restrictive in terms of performance.

For that purpose, we randomly generate models W consisting of $T = 10$ tasks in $d = 10$ dimensions, such that all tasks are orthogonal to each other. The training set X_{train} is composed of $n = 50$ instances, each element of X_{train} being sampled from a normal distribution $\mathcal{N}(0, 1)$. We simulate the response variable $Y_{train} \in \mathbb{R}^{n \times T}$ according to $Y_{train} = X_{train}W + \epsilon$, where ϵ is a noise matrix of i.i.d. centered Gaussian variables

with variance σ^2. We estimate the performance of each model on a test set of 1000 samples generated similarly. We also measure how orthogonal the models are, by the mean absolute difference between the angle between two columns of W and $\pi/2$. For each value of γ we estimate the Xiao model with different regularization parameters λ over a grid of 21 values in $\{10^{-4}, ..., 10^{1}\}$ and regularly spaced after log transform; the grid was set to ensure that it covered good parameters for all methods. For each γ, we report the performance of the best λ in terms of test MSE. We repeat the full procedure 100 times and report the average results over the 100 repeats.

Figure 3.2 shows the performance of the methods in terms of test error (top), and in terms of how far the models learned are from orthogonal models (bottom). On each plot, the horizontal axis is the γ parameter on the diagonal of K defined in (3.20), and the vertical dotted line corresponds to the atomic norm (3.21) and is the transition from convex (to its right) to non-convex (to its left). From left to right, we show results corresponding to increasing noise in the response variable, with the variance of ϵ set respectively to $1, 2.5$ and 4. We see that in the small noise regime (left), non-convex formulations perform better while with high noise (right), the convex formulations are more adapted. Inbetween (middle), the best performance is reached for slightly non-convex penalties. In all cases, the models learned are similar in terms of how non-orthogonal they are; we see that non-convex formulations lead to significantly more orthogonal models than convex formulations. Overall, these results suggest that restricting ourselves to strictly convex penalties may be restrictive and sub-optimal in some cases; they show that non-convex penalties can allow to learn more orthogonal models with better performance.

3.6.2 Regression with disjoint supports

As a second proof of concept, we check the relevance of the formulation presented in Section 3.5 to jointly learn linear models with disjoint supports. For that purpose, we simulate data as in Section 3.6.1, with the additional constraint that the columns of W are orthogonal and have disjoint supports. Since $d = T = 10$, this means that W is simply diagonal. We fix the noise level at $\sigma^2 = 1$, and simulate training sets of increasing size between 10 and 50 samples, repeating the full procedure 100 times. We compare four methods: (i) the Xiao model with varying parameter γ according to (3.20), leading to orthogonal but non-sparse vectors, (ii) our new method (3.18) again with convex and non-convex formulations by varying γ in (3.20), (iii) a baseline ridge regression model and (iv) a LASSO regression model leading to sparse but not necessarily orthogonal vectors. For each model, a 5-fold cross-validation is performed on the training set to select an optimal regularization parameter λ, which is then used to train the model on the full training set before doing a prediction on an independent test set. We assess the performance of each method on the test set in terms of accuracy (measured by the MSE), and in terms of disjoint support recovery, measured as the

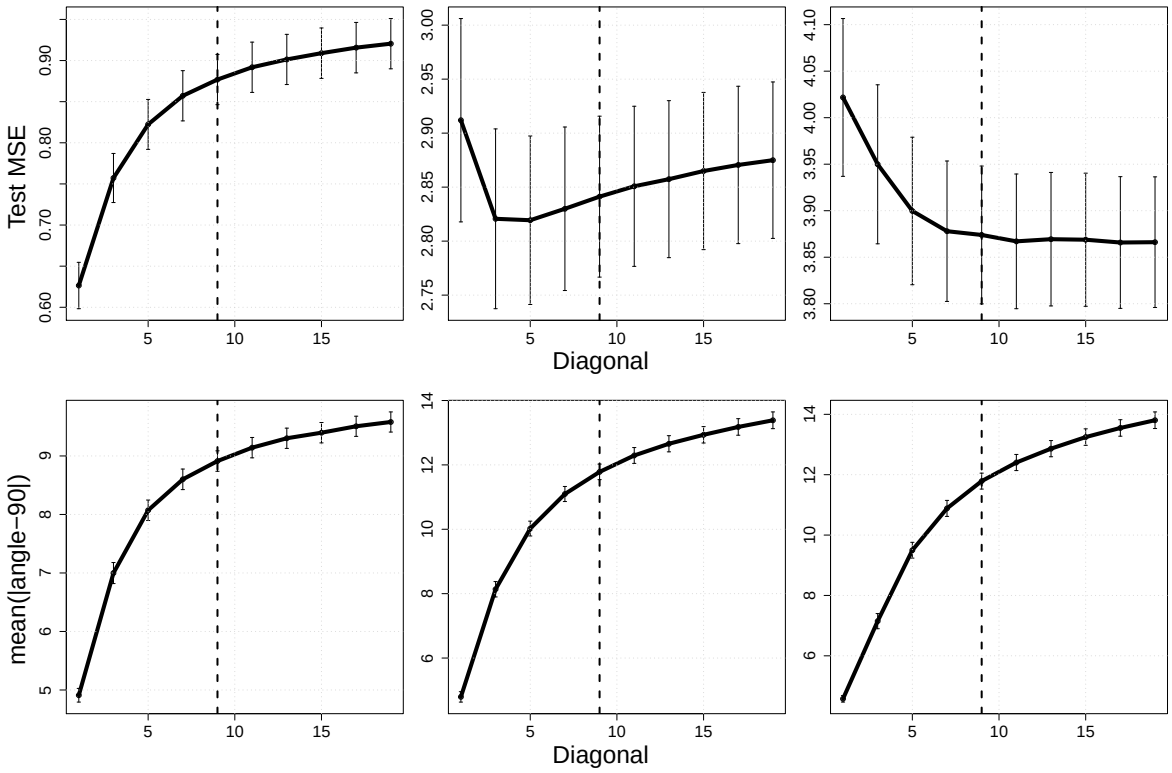

Figure 3.2: **The effect of convexity.** Test MSE (top) and deviation from pairwise orthogonality (bottom) as a function of the convexity parameter γ, from low to high noise regimes (from left to right: $\sigma^2 \in \{1, 2.5, 4\}$). On each plot, the horizontal axis is the γ parameter on the diagonal of K defined in (3.20). The vertical dotted line corresponds to the atomic norm (3.21).

proportion of features which are correctly selected in a single column of W.

The results are shown in Figure 3.3, where for sake of clarity we only report the results of Xiao and Disjoint Supports for the optimal diagonal value γ, which in both cases is equal to 0.1, corresponding to a very non-convex penalty. In terms of performance, we see that Xiao is a bit better than Ridge regression for $n = 50$ training points, which is coherent with the observation made in Section 3.6.1 in the small-noise regime, although for less than 30 samples, Ridge regression is better. Both methods are outperformed by LASSO, which in this case benefits from the very sparse structure of W. Interestingly, the new Disjoint Support model significantly outperforms all other methods for all training set sizes (P-value $< 10^{-3}$). As for the ability of different methods to correctly recover the disjoint supports, we see that Disjoint Supports shows increasing support recovery score for large training set size, and outperforms LASSO which induces global sparsity but is not able to affect features to an unique column. Ridge Regression and Xiao are not shown because they do not achieve any sparsity in the model they learn. In summary, this simulation shows that the Disjoint Supports model has the potential to outperform other methods when the model to learn is sparse with disjoint supports.

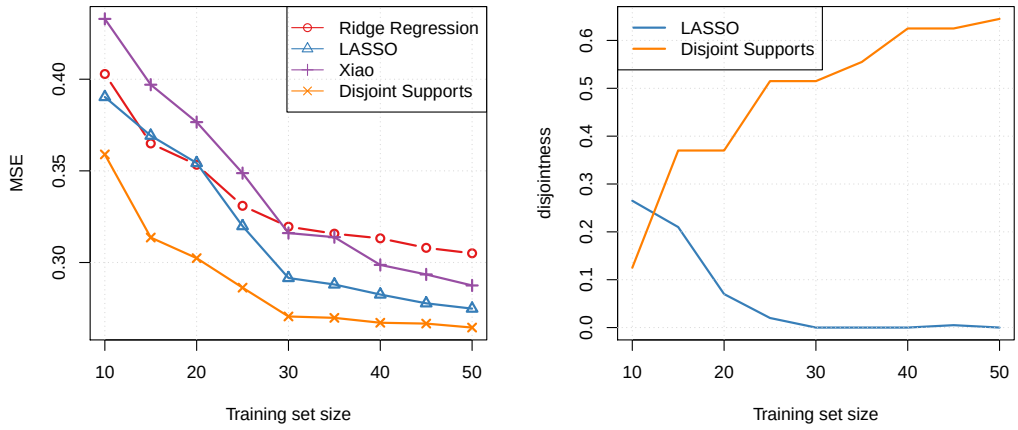

Figure 3.3: **Sparse regression with disjoint supports.** Test MSE for training set of increasing size (left), and proportion of correctly affected features (right). Ridge regression and Xiao are not shown on the right plot because they are not sparse.

3.6.3 Learning two groups of unrelated tasks

We use the **J**apanese **F**emale **F**acial **E**xpression (JAFFE) database ([88]). It is composed of 213 images of $T_2 = 10$ subjects displaying a range of $T_1 = 7$ mutually exclusive expressions. We seek to learn a model that predicts the emotion expressed in an unlabelled picture. For this purpose, each image was preprocessed with the method described in [139], leading to a $d = 203$ dimensional representation. To avoid some bias due to some subjects, we consider that features which are useful for identifying subjects should not be used in expression recognition tasks. We can formulate this problem in the disjoint learning framework: there is a group of 10 tasks related to subjects (W_2) that is different to the group of 7 emotions tasks (W_1). In this configuration, the constraint matrix K contains orthogonality constraints only between columns of different groups and is given by (3.22):

$$K = \begin{pmatrix} K_{11} & & 0 & 1 & \dots & 1 \\ & \ddots & & & \ddots & \\ 0 & & K_{77} & 1 & \dots & 1 \\ 1 & \dots & 1 & K_{88} & & 0 \\ & \ddots & & & \ddots & \\ 1 & \dots & 1 & 0 & & K_{1717} \end{pmatrix}. \tag{3.22}$$

The method OrthoMTL described in [139] aims to affect features to group of tasks with an orthogonality assumption. For this reason, we were interested in evaluating our method on JAFFE. We compare Xiao's approach to Ridge regression, OrthoMTL and OrthoMTL-EN using the same design: we select randomly m instances in training

set and the remaining ones in test set. Then we select, by 5-folds cross-validation, the best regularization parameter $\lambda = 10^k$, where $k \in \{-2, ..., 2\}$ and obtain a misclassification rate on test set. Note that for Xiao's approach, we also optimize the diagonal values of $(K_{ii})_i$ on the $\{0.1, 0.5, 1, 8.37, 17\}$ grid, where we only consider the case where $K_{11} = ... = K_{1717} = \gamma$. The value 8.37 ($\sim \sqrt{T_1 \times T_2}$) corresponds approximately to the minimal γ value leading to a positive semidefinite K matrix and to the atomic norm in (3.21). For OrthoMTL approaches, we also optimize, by cross-validation, the parameters described in [139], $\rho \in \{10^{-2}, ..., 10^2\}$ for convexity, $\lambda \in \{10^4, ..., 10^7\}$ for orthogonality, and $\gamma \in \{10^{-4}, ..., 10^2\}$ for sparsity. We have repeated the described experiment 50 times. The different learning curves are averaged over repeats and represented in Figure 3.4. The two approaches proposed in [139] have better performances on this dataset that contains two groups of tasks. We note that the Xiao penalty inducing orthogonality between W_1 and W_2 slightly outperforms standard Ridge regression approach. Interestingly, the optimal diagonal value for Xiao is 8.37 meaning that the atomic norm formulation is well suited for this problem. According to cross-validation on OrthoMTL and OrthoMTL-EN, the optimal parameters lead to select model with orthogonality between the two groups but no sparsity property ($\gamma \sim 0$). For this γ value, both OrthoMTL and OrthoMTL-EN are equivalent, explaining why the learning curves are similar. Given that observation, we do not evaluate Disjoint supports on this dataset.

In addition of the classification performances, we also investigate in Fig 3.5 the correlation pattern in the matrix product $[W_1, W_2]^\top [W_1, W_2]$, observed in models learned for the different approaches. We observe that for the Ridge regression (top left), there is no correlation structure within the two groups of tasks. However, the three others graphics (OrthoMTL, Xia and Disjoint supports) show block structures between the two groups of tasks that correspond to orthogonality constraints. Interestingly, for these three methods, correlation values are higher for the 10 subjects block (top right); it can be explained by the choice of a constant diagonal value for the K matrix between the two groups. Given the different group sizes, there can be a stronger weight on the Subjects constraints.

3.6.4 Disjoint supports for Mass-spectrometry data

We use MicroMass data for the final test of disjoint supports learning. This mass-sepctrometry dataset is described in Section 2.2. We recall that it consists in a reference panel of 571 spectra covering 9 bacterial genera. According to results shown in Section 2.5, genus-level classification is not a difficult task (good classification rate over 98%). Because of the relative high-dimensionality of the mass-spectra[1], we propose to use disjoint supports learning for feature selection. We aim to learn 9 orthogonal tasks (one for each genus) with genus-specific peaks, meaning that a peak will be discrimi-

[1]Each spectrum is a list of $d = 1300$ peaks.

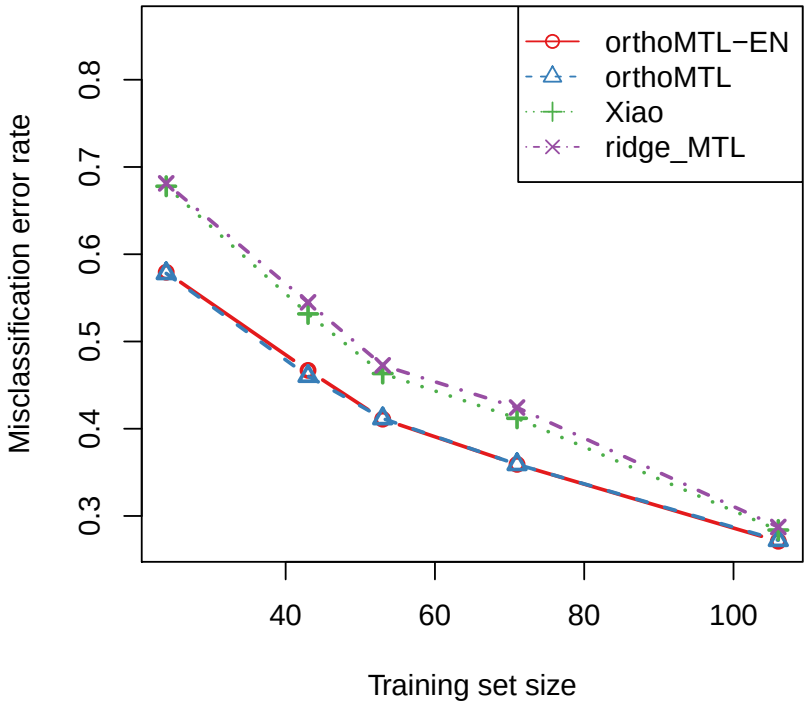

Figure 3.4: **JAFFE dataset: learning curves.** Misclassification rate function of training set size. Comparison between Xiao (green) penalty and Ridge regression (purple), OrthoMTL (blue), OrthoMTL-EN (red).

native for a genus against the other ones. In this case, the constraint matrix $K \in \mathbb{R}^{9\times 9}$ contains orthogonality constraints between all columns and is given by (3.23):

$$K = \begin{pmatrix} K_{11} & & 1 \\ & \ddots & \\ 1 & & K_{99} \end{pmatrix}. \tag{3.23}$$

We evaluate our method in the same way we did for 6-columns synthetic data in Section 3.6.1: we consider several diagonal values (0 to 10) covering non-convex and convex Ω_K and a regularization parameter grid $\lambda = 10^k$, where $k \in \{-3, ..., 2\}$. For each possible diagonal value, we learn models in 5-folds cross-validation and choose the best model according to good classification rate. It appears that the optimal accuracy of Disjoint supports is very similar (98.6%) to standard approaches benchmarked in Table 2.3 and is obtained in a non-strictly convex case. In addition, we also represent in Figure 3.6, the model structure learned with the sparsity inducing LASSO and our approach, Disjoint supports. These figures have a structure similar to the heatmap in Section 2.3 that represents the MicroMass dataset. Each line is a W column and

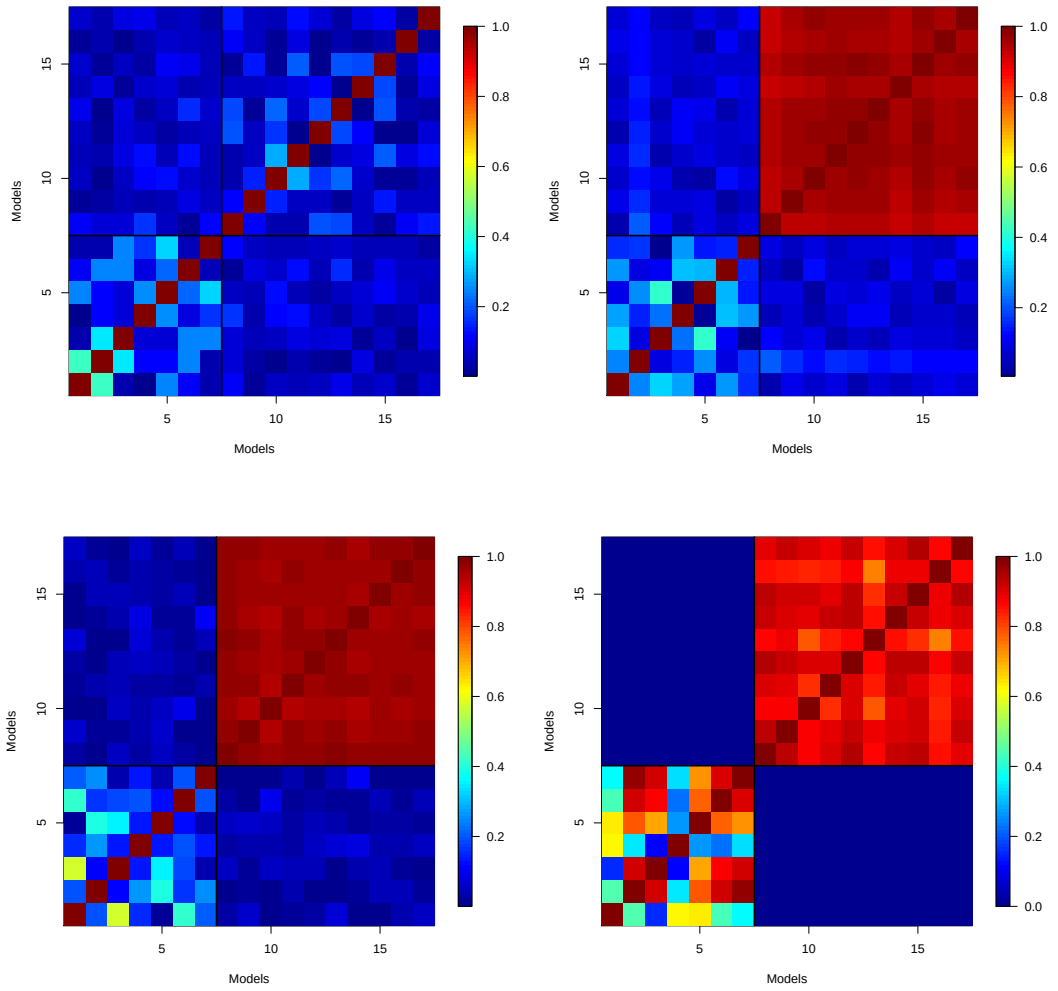

Figure 3.5: **JAFFE dataset: correlation in learned models.** Given a learned matrix W, we represent the matrix product $[W_1, W_2]^\top [W_1, W_2]$ with high correlation in red and orthogonality in blue, corresponding to no correlation. Comparison between Ridge (top left), OrthoMTL (top right), Xiao (bottom left) and Disjoints supports (bottom right).

corresponds to a specific genus-level model, and each column represents a feature weight in the different models. The 9 different colors correspond to the features affected to an unique genus and the black blocks are features that are present in at least two W columns. The white space on the right illustrates the number of features that are not used in any model and that are put to zero in all W columns. Interestingly, we observe that learning with Disjoint supports allows to affect a larger proportion of the features to a single model, while the LASSO model shows numerous features that are either used by two or more models, either discarded from the models.

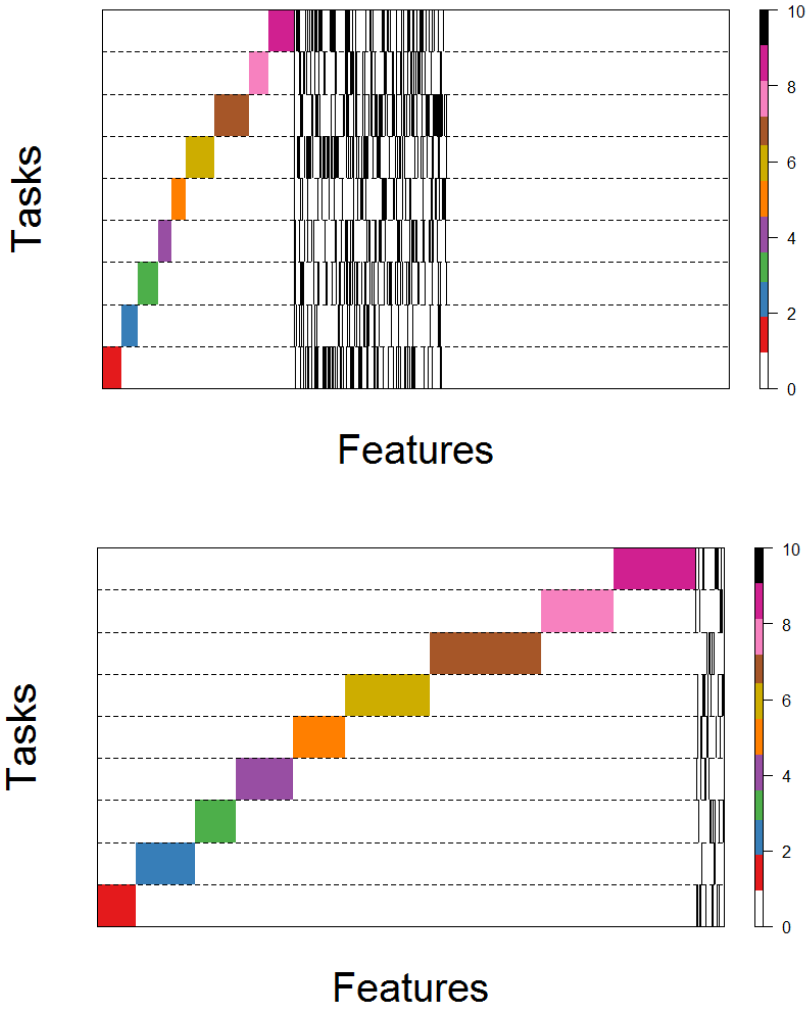

Figure 3.6: **MicroMass dataset: structured sparsity.** Comparison between Lasso (top) and Disjoint supports (bottom) approaches.

3.7 Conclusion

We have extended the work of [184] in two directions: on the one hand, we have investigated the possibility to work with non-strictly convex or non-convex formulations, leading to more aggressive control of model orthogonality, and on the other hand we have shown how models to learn orthogonal columns can be extended to learn sparse models with disjoint supports. In the two-columns case, we have proved that the penalty of [184] is an atomic norm derived from the set of scaled orthogonal matrices, and for the general case $T > 2$ we have shown that for suitable choices of parameters it can be written as a linear combination of atomic norms applied to pairs of columns. In terms of algorithms, the RDA algorithm proposed by [183] is only suitable to solve the problem (3.12) in the strictly convex case, and we have shown that in the limit case where Ω_K is convex but not strictly convex we can solve iteratively with a series of 6-dimensional SDP. Our simulations show that considering non-convex versions of

the penalty can be relevant, in particular for small noise regimes. Interestingly, we observed that non-convex formulations lead to more orthogonal models than convex formulations, and that the Disjoint Support model significantly outperformed all other models when the disjoint support hypothesis was met. We also evaluate described approaches on real datasets: JAFFE and MicroMass. On the first one, results show that the structure learned by Xiao approach is comparable to methods that are only dedicated to the specific case of two groups of unrelated tasks. For the second dataset, Disjoint supports achieves correct classification rate at the genus-level, similar to previously benchmarked Support Vector Machines. However the learned model shows interesting block structure with features uniquely affected to a given genus. In the future, we plan to investigate the relevance of these approaches on other classification problems, such as hierarchical document classification [184] or spoken letter-name identification with different speakers, like in ISOLET dataset [48].

Chapter 4

Large-scale Machine Learning for Metagenomics Sequence Classification

*This chapter has been submitted under a slightly different form as [**?**], a joint work with Pierre Mahé, Maud Tournoud, Jean-Baptiste Veyrieras and Jean-Philippe Vert.*

Abstract

Metagenomics characterizes the taxonomic diversity of microbial communities by sequencing DNA directly from an environmental sample. One of the main challenges in metagenomics data analysis is the binning step, where each sequenced read is assigned to a taxonomic clade. Due to the large volume of metagenomics datasets, binning methods need fast and accurate algorithms that can operate with reasonable computing requirements. While standard alignment-based methods provide state-of-the-art performance, compositional approaches that assign a taxonomic class to a DNA read based on the k-mers it contains have the potential to provide faster solutions. In this work, we investigate the potential of modern, large-scale machine learning implementations for taxonomic affectation of next-generation sequencing reads based on their k-mers profile. We show that machine learning-based compositional approaches benefit from increasing the number of fragments sampled from reference genome to tune their parameters, up to a coverage of about 10, and from increasing the k-mer size to about 12. Tuning these models involves training a machine learning model on about 10^8 samples in 10^7 dimensions, which is out of reach of standard softwares but can be done efficiently with modern implementations for large-scale machine learning. The resulting models are competitive in terms of accuracy with well-established alignment tools for problems involving a small to moderate number of candidate species, and for reasonable amounts of sequencing errors. We show, however, that compositional approaches are still limited in their ability to deal with problems involving a greater number of species, and more sensitive to sequencing errors. We finally confirm that compositional approach achieve faster

prediction times, with a gain of 3 to 15 times with respect to the BWA-MEM short read mapper, depending on the number of candidate species and the level of sequencing noise.

Résumé

La métagénomique permet de caractériser la diversité taxinomique de communités microbiennes, directement en séquençant un échantillon brut, sans étape de culture. Un des principaux défis en métagénomique est l'assignation de chaque séquence à une entité taxinomique. Les larges volumes considérés dans les données métagénomiques font que des algorithmes efficaces et rapides sont requis pour cette étape d'affectation. Alors que les approches classique par similarité offrent des performances de référence, les approches dites compositionnelles assignent une classe taxonomique à chaque séquence d'ADN en se basant sur son contenu en k-mers et ont le potentiel de fournir des solutions plus rapides. Dans cette étude, nous évaluons le potentiel d'approches modernes et grande échelle de classification pour l'affectation taxonomique de 'reads' en se basant sur leurs profils en k-mers. Nous montrons que les approches compositionnelles utilisant l'apprentissage statistique tirent avantage du grand nombre de fragments extraits de génomes de référence utilisés pour estimer leurs paramètres, jusqu'à une couverture de 10, ainsi que de longs k-mers, jusqu'à une longueur de 12. Construire ces modèles requiert une étape d'apprentissage sur environ 10^8 exemples représentés par des vecteurs de dimension 10^7, ce qui est hors de portée des algorithmes classiques, mais peut être efficacement avec des implémentations modernes à grande échelle. Ces modèles ont des performances comparables aux outils bio-informatiques standards utilisant l'alignement de séquences, pour des problèmes impliquant un nombre restreint d'espèces microbiennes et un niveau raisonnable de bruit de séquençage. Cependant, nos résultats suggèrent que les approches compositionnelles sont limitées lorsqu'il s'agit de considérer un très grand nombre d'espèces et sont plus sensibles à de hauts niveaux de bruit de séquençage. Nous confirmons également que les approches compositionnelles présentent un net avantage en terme de temps de prédiction par rapport à une méthode d'alignement. Notre approche est entre 3 et 15 fois plus rapide que BWA-MEM : cela dépend du nombre d'espèces considérées et du niveau de bruit de séquençage.

4.1 Introduction

Recent progress in next-generation sequencing (NGS) technologies allow to access large amounts of genomic data within a few hours at a reasonable cost [154]. In metagenomics, NGS is used to analyse the genomic content of microbial communities by sequencing all DNA present in an environmental sample [134]. It gives access to all organisms present in the sample even if they do not grow on culture media [77], and

allows us to characterize with an unprecedented level of resolution the diversity of the microbial realm [129].

The raw output of a metagenomics experiment is a large set of short DNA sequences (reads) obtained by high-throughput sequencing of the DNA present in the sample. There exist two main approaches to analyze these data, corresponding to slightly different goals. On the one hand, *taxonomic profiling* aims to estimate the relative abundance of the members of the microbial community, without necessarily affecting each read to a taxonomic class. Recent works like WGSQuikr [95] or GASIC [**?**] proved to be very efficient for this purpose. *Taxonomic binning* methods, on the other hand, explicitly affect each read to a taxonomic clade. This process can be unsupervised, relying on clustering methods to affect reads to operational taxonomic units (OTU), or supervised, in which case reads are individually affected to nodes of the taxonomy [109]. While binning is arguably more challenging that profiling, it is a necessary step for downstream applications which require draft-genome reconstruction. This may notably be the case in a diagnostics context, where further analyses could aim to detect pathogen micro-organisms [121] or antibiotic resistance mechanisms [145]. In this chapter we focus on the problem of supervised taxonomic binning, where we wish to assign each read in a metagenomics sample to a node of a pre-defined taxonomy. Two main computational strategies have been proposed for that purpose: (i) alignment-based approaches, where the read is searched against a reference sequence database with sequence alignment tools like BLAST [79] or short read mapping tools (e.g., BWA, [103]), and (ii) compositional approaches, where a machine learning model such as a naive Bayes (NB) classifier [174, 127] or a support vector machine (SVM, [118, 128]) is trained to label the read based on the set of k-mers it contains. Since the taxonomic classification of a sequence by compositional approaches is only based on the set of k-mers it contains, they can offer significant gain in terms of classification time over similarity-based approaches. Training a machine learning model for taxonomic binning can however be computationally challenging. Indeed, compositional approaches must be trained on a set of sequences with known taxonomic labels, typically obtained by sampling error-free fragments from reference genomes. In the case of NB classifiers, explicit sampling of fragments from reference genomes is not needed to train the model: instead, a global profile of k-mer abundance from each reference genome is sufficient to estimate the parameters of the NB model, leading to simple and fast implementations [174, 140, 127]. On the other hand, in the case of SVM and related discriminative methods, an explicit sampling of fragments from reference genomes to train the model based on the k-mer content of each fragment is needed, which can be a limitation for standard SVM implementations. For example, [128] sampled approximately 10,000 fragments from 1768 genomes to train a structured SVM (based on a k-mer representation with $k = 4, 5, 6$), and reported an accuracy competitive with similarity-based approaches. Increasing the number of fragments sampled to train a SVM may improve its accuracy, and allow us to investigate larger values of k. However

it also raises computational challenges, as it involves machine learning problems where a model must be trained from potentially millions or billions of training examples, each represented by a vector in 10^7 dimensions for, e.g., $k = 12$.

In this work, we investigate the potential of compositional approaches for taxonomic label assignment using modern, large-scale machine learning algorithms.

4.2 Linear models for read classification

In most of compositional metagenomics applications, a sequence is represented by its k-mer profile, namely, a vector counting the number of occurrences of any possible word of k letters in the sequence. Only the A, T, C, G nucleotides are usually considered to define k-mer profiles, that are therefore 4^k-dimensional vectors, like illustrated in Figure 4.1.

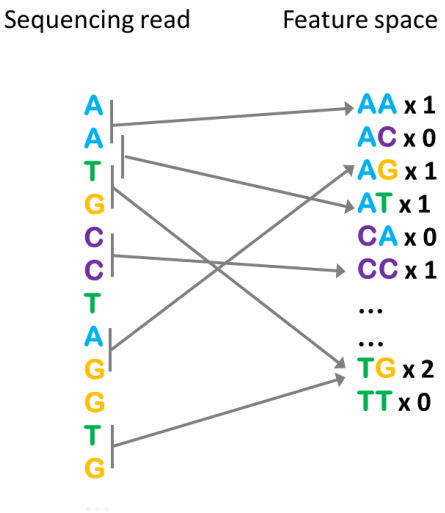

Figure 4.1: **From sequencing read to vector space representation.** The left side represents a nucleotide sequence that is changed into k-mer count vector (here $k = 2$). This k-mer profile (right) is used as a vector space representation for machine learning approaches.

Although the size of the k-mer profile of a sequence of length l increases exponentially with k, it contains at most $l - k + 1$ non-zero elements since a sequence of length k contains $l - k + 1$ different k-mers.

Given a sequence represented by its k-mer profile $x \in \mathbb{R}^{4^k}$, we consider linear models to assign it to one of K chosen taxonomic classes. A linear model is a set of weight vectors $w_1, \ldots, w_K \in \mathbb{R}^{4^k}$ that assign x to the class

$$\arg \max_{j=1,\ldots,K} w_j^\top x \,,$$

where $w^\top x$ is the standard inner product between vectors. To train the linear model,

we start from a training set of sequences $x_1, \ldots, x_n \in \mathbb{R}^{4^k}$ with known taxonomic labels $c_1, \ldots, c_n \in \{1, \ldots, K\}$. A NB classifier, for example, is a linear model where the weights are estimated from the k-mer count distributions on each class. Another class of linear models popular in machine learning, which include SVM, are the discriminative approaches that learn the weights by solving an optimization problem which aims to separate the training data of each class from each other. More precisely, to optimize the weight w_j of the j-th class, one typically assigns a binary label y_i to each training example ($y_i = 1$ if $c_i = j$, or $y_i = -1$ otherwise) and solves an optimization problem of the form

$$\min_w \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, w^\top x_i) + \lambda \|w\|^2 \,, \tag{4.1}$$

where $\ell(y, t)$ is a loss function quantifying how "good" the prediction t is if the true label is y, and $\lambda \geq 0$ is a regularization parameter to tune, helpful to prevent overfitting in high dimension, as detailed in Section 1.2.4. A SVM solves (4.1) with the hinge loss $\ell(y, t) = max(0, 1 - yt)$, but other losses such as the logistic loss $\ell(y, t) = \log(1 + \exp(-yt))$ or the squared loss $\ell(y, t) = (y - t)^2$ are also possible and often lead to models with similar accuracies. These models have met significant success in numerous real-world learning tasks, including compositional metagenomics [128]. In this work, we use the squared loss function and choose $\lambda = 0$, a setting that seemed appropriate from preliminary experiments.

4.2.1 Large-scale learning of linear models

Although learning linear models by solving (4.1) is now a mature technology implemented in numerous softwares, metagenomics applications raise computational challenges for most standard implementations, due to the large values that n (number of reads in the training set), $p = 4^k$ (dimension of the models) and K (number of taxonomic classes) can take.

The training set is typically obtained by sampling fragments from reference genomes with known taxonomic class. For example, [128] sampled approximately $n = 10,000$ fragments from $1,768$ genomes to train SVM models based on k-mer profiles of size $k = 4, 5, 6$. However, the number of distinct fragments that may be drawn from a genome sequence is approximately equal to its length (by sampling a fragment starting at each position in the genome), hence can reach several millions for each microbial genome, leading to potentially billions of training sequences when thousands of reference genomes are used. While considering every possible fragment from every possible genome may not be the best choice because of the possible redundancy between the reads, it may still be useful to consider a significant number of fragments to properly account for the intra and inter species genomic variability. Similarly, exploring models with k larger than 6, say 10 or 15, may be interesting but requires (i) the capacity to manipulate the corresponding 4^k-dimensional vectors ($4^{15} \sim 10^9$), and (ii) large training sets since many examples are needed to learn a model in high dimension. Fi-

nally, real-life applications involving actual environmental samples may contain several hundreds microbial species, casting the problem into a relatively massive multiclass scenario out of reach of most standard implementations of SVM.

To solve (4.1) efficiently when n, k and K take large values, we use a dedicated implementation of stochastic gradient descent (SGD) described in Section 1.2.5 and available in the Vowpal Wabbit software (VW, [98, 1]). In short, SGD exploits the fact that the objective function in (4.1) is an average of n terms, one for each training example, to approximate the gradient at each step using a single, randomly chosen term. Although SGD requires more steps to converge to the solution than standard gradient descent, each step is n times faster and the method is overall faster and more scalable. In addition, although the dimension $p = 4^k$ of the data is large, VW exploits the fact that each training example is sparse, leading to efficient memory storage and fast updates at each SGD step. In practice, VW can train a model with virtually no limit on n as long as the data can be stored on a disk (they are not loaded in memory). As for k, VW can handle up to 2^{32} distinct features, and the count of each k-mer is randomly mapped to one feature by a hash table. This means that we have virtually no limit on k, except that when k approaches or exceeds the limit (such that $4^k = 2^{32}$, i.e., $k = 16$), collisions will appear in the hash tables and different k-mers will be counted together, which may impact the performance of the model.

4.3 Data

We simulate metagenomics samples by generating reads from three different reference databases, which we refer to below as the *mini*, the *small* and the *large* databases.

The *mini* reference database contains 356 complete genome sequences covering 51 bacterial species, listed in Table 4.1. We use this database to train and extensively vary the parameters of the different models. To measure the performance of the different models, we generate new fragments from 52 genomes not present in the reference database, but originating from one of the 51 species[1].

The *small* and *large* databases are meant to represent more realistic situations, involving a larger number of candidate bacterial species and a larger number of reference genomes. To define the reference and validation databases that will respectively be used to build and evaluate the predictive models, we first downloaded the 5201 complete bacterial and archeal genomes available in the NCBI RefSeq database as of July 2014 [130], by means of a functionality embedded in the Fragment Classification Package (FCP) [127]. We then filtered these sequences according to a criterion proposed in [127]: we only kept genomes that belong to genera represented by at least 3 species. We also removed genomes represented by less than 10^6 nucleotides in order to filter draft genome sequences, plasmids, phages, contigs and other short sequences. The 2961

[1]Two genomes are indeed available for the *Francisella tularensis* species, one of which originating from the *novicida* subspecies.

Table 4.1: List of the 51 microbial species in the *mini* reference database.

Species name	Species name
Acetobacter pasteurianus	*Methylobacterium extorquens*
Acinetobacter baumannii	*Mycobacterium bovis*
Bacillus amyloliquefaciens	*Mycobacterium tuberculosis*
Bacillus anthracis	*Mycoplasma fermentans*
Bacillus subtilis	*Mycoplasma genitalium*
Bacillus thuringiensis	*Mycoplasma mycoides*
Bifidobacterium bifidum	*Mycoplasma pneumoniae*
Bifidobacterium longum	*Neisseria gonorrhoeae*
Borrelia burgdorferi	*Propionibacterium acnes*
Brucella abortus	*Pseudomonas aeruginosa*
Brucella melitensis	*Pseudomonas stutzeri*
Buchnera aphidicola	*Ralstonia solanacearum*
Burkholderia mallei	*Rickettsia rickettsii*
Burkholderia pseudomallei	*Shigella flexneri*
Campylobacter jejuni	*Staphylococcus aureus*
Corynebacterium pseudotuberculosis	*Streptococcus agalactiae*
Corynebacterium ulcerans	*Streptococcus equi*
Coxiella burnetii	*Streptococcus mutans*
Desulfovibrio vulgaris	*Streptococcus pneumoniae*
Enterobacter cloacae	*Streptococcus thermophilus*
Escherichia coli	*Thermus thermophilus*
Francisella tularensis	*Treponema pallidum*
Helicobacter pylori	*Yersinia enterocolitica*
Legionella pneumophila	*Yersinia pestis*
Leptospira interrogans	*Yersinia pseudotuberculosis*
Listeria monocytogenes	

remaining sequences originate from 774 species, among which 193 are represented by at least 2 strains. We split the sequences of these 193 species into two parts. We randomly pick one strain within each of these 193 species to define a validation database, that will be used to estimate classification performance, through the sampling of genomic fragments or the simulation of sequencing reads. The remaining sequences of these 193 species define a first reference database, referred to as *small* below. In addition we define a larger reference database by adding to the *small* database described above the genomes originating from the $774 - 193 = 581$ species represented by a single genome. The larger database, referred to as *large* below, therefore involves the 774 species available after filtering the NCBI database, and not solely the 193 represented in the validation database.

4.4 Results

4.4.1 Proof of concept on the *mini* database

In this section we present a proof of concept on the *mini* dataset aiming to evaluate the impact of different Vowpal Wabbit parameters, like the number of passes and the hash table size. For that purpose, we learn several classification models based on fragments of length $L = 200$ or $L = 400$ sampled from the 356 reference genomes in the *mini* reference database, represented by k-mers of size in $\{4, 6, 8, 10, 12\}$.

Once the VW parameters are optimized, we consider increasing the number of fragments used to train the model as well as the length of the k-mers considered. The number of fragments used to learn the models is gradually increased by drawing several "batches" of fragments in order to cover, on average, each nucleotide of the reference genomes a pre-defined number of times c. We vary the coverage c between 0.1 to its maximal value, equal to the length of the fragments considered. This leads to learning models from around $n = 2.7 \times 10^5$, for $c = 0.1$ and $L = 400$, up to around $n = 1.1 \times 10^9$ fragments, when c reaches its maximal value. This is way beyond the configurations considered for instance in [128], where SVM models were learned from approximately 10^4 fragments drawn from $1,768$ genomes.

To assess the performance of these models we consider two sets of $134,319$ fragments, of respective length 200 and 400, drawn from the 52 complete genomes that are not in the reference database used to train the models. Performance is measured by first computing, for each species, the proportion of fragments that are correctly classified, and considering its median value across species. In a multiclass setting, this indicator is indeed less biased towards over-represented classes than the global rate of correct classification.

Loss functions and classification strategies

There are multiple ways to formulate a multiclass classification problem solved by linear predictors, like previously described in Section 1.3.2. We evaluate OVA and ECT classification strategies and three loss functions, described in Section 1.2.3, on the *mini* datasets with a mean coverage c equal to 1 for the training step. We report results in Figure 4.2. These figures show the median accuracy at species level as a function of k-mer size and loss function for OVA multiclass strategy (left) and ECT strategy (right). We only report results for fragments of length 400, but comparable performances were observed on fragments of length 200. For both OVA and ECT, we note that squared loss and hinge loss achieve comparable performances while logistic loss function is less adapted to our problem. Interestingly, we observe that OVA model provides slightly better performances than ECT for all loss functions and k-mer sizes considered. As explained in Section 1.3.2, ECT achieves faster computation but usually leads to less accurate model. Based on these results, we consider a OVA multiclass strategy with a

squared loss function in following experiments.

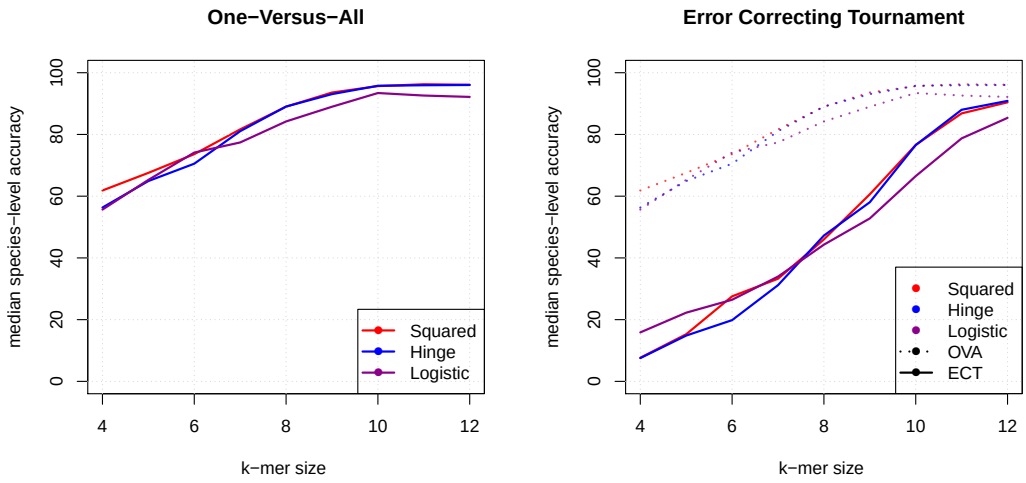

Figure 4.2: **Loss functions and classification strategies.** Left: One Versus All (OVA) strategy. Right: Error Correcting Tournament (ECT) strategy. These figures give median accuracy at species level for different loss functions: squared (red), hinge (blue) and logistic (purple). Performances are reported as a function of the k-mer sizes on fragments of length 400.

Number of training passes

As explained in Section 1.2.5, most of algorithms based on SGD, like Vowpal Wabbit, may require multiple passes over the training examples to achieve good convergence rate. Figure 4.3 shows the results obtained in terms of median accuracy at species level as a function of number of passes on the training set. For two mean coverage values $c = 0.1$ (left) and $c = 1$ (right), we note that the number of passes does not affect performances for any values of k. However, the training time for a classifier is directly linked to the number of passes, so we put it equal to one for the next experiments.

Features collisions in the hash table

In order to efficiently parse the k-mer content of each sequence fragment, algorithms like Liblinear or Vowpal Wabbit build a hash table [114] matching any possible k-mer with a hash key, acting as a feature index. Given the short read length we consider (< 1000), this step provides a sparse representation of each example in the training and evaluation sets. One reason for Vowpal Wabbit efficiency is the optimized feature hashing step, directly computing an hash table from the input bag-of-words, or here bag-of-kmers, representation. However, even an efficient hashing step leads to collisions in the table when two or more values receive the same key. Interestingly, we can illustrate hash table collisions by using the birthday paradox [119]: if 40 keys are randomly hashed into 365 slots, there is approximately a 90% chance that at least two of them end up hashed to the same slot. To limit collisions in its hash table, Vowpal Wabbit can be

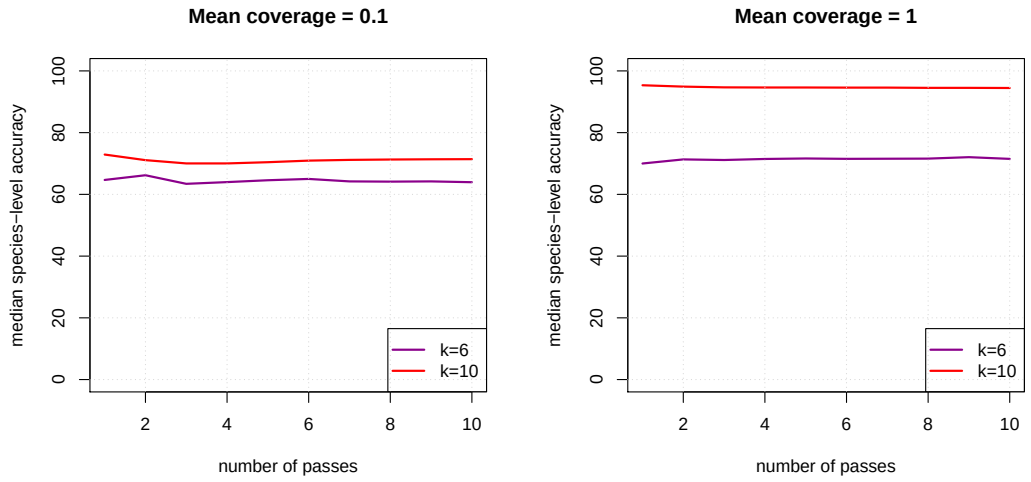

Figure 4.3: **Number of passes during the training step.** Left: Mean coverage=0.1. Right: Mean coverage=1. These figures give median accuracy at species level for different k-mer sizes: $k = 6$ (purple) and $k = 10$ (red). Performances are reported as a function of the number of passes on the training set for fragments of length 400. For clarity purposes, we only report results for two k-mer sizes ($k = \{6, 10\}$), but other k values lead to similar results.

parameterized with an hash table size from 1 to 2^{32} entries. Note that, in the multiclass case, VW stores one model for each different class. Given the dictionary size and the number of classes K in our problem, an upper bound of the required number of slots is given by $4^k \times K$. Considering a hash table size equal to this upper bound, one can compute the expected collision rate as the ratio between number of slots with collisions and total number of slots. For b slots, the probability that a key among n keys receives exactly the same slot than another particular key is given by $1 - \left(\frac{b-1}{b}\right)^{n-1}$. Then, the collision counting is equal to the sum of this probability function over all n keys, that is,

$$\sum_{k=1}^{n} \left(1 - \left(\frac{b-1}{b}\right)^{k-1}\right) = n - b + b\left(\frac{b-1}{b}\right)^{n}.$$

Interestingly, where $n = b$, the collision rate is around 36.8% corresponding to the probability for an instance of ending up in the test data in bootstrap setting described in [70, Chapter 7].

We evaluate the impact of different hash table sizes on classification performances and report results in Figure 4.4. These two figures (left: $k = 8$, right: $k = 10$) represent median accuracy by taxon and collision rate as a function of hash table size. Results are obtained on fragments of length 400 for a mean coverage $c = 0.1$. The purple dashed line corresponds to the expected number of features, given k. Interestingly, we note that performances drop with hash tables smaller than $\log_2(4^k \times K)$ bits. It is correlated with high collision rates ($> 50\%$), which means that half of hash table slots have at least 2 keys pointing on it. In addition, there is no negative effect on performances for

hash tables larger than $\log_2(4^k \times K)$ bits. So, if there is no limitation due to computer memory access, we propose to keep the hash table size equal to 2^{31}. On this dataset, 52 models have to be stored in the hash table, which reduces the number of entries available per model to $2^{32}/52 \sim 2^{32-6} = 4^{13}$.

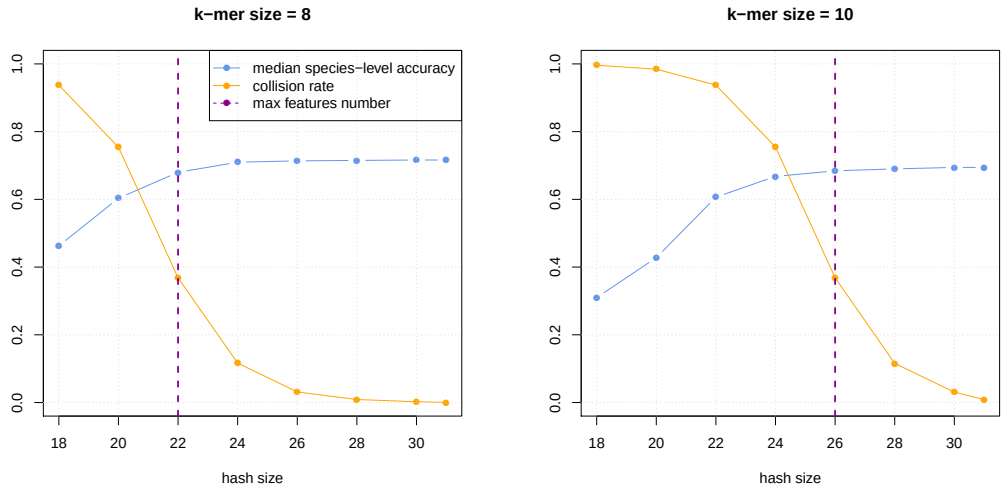

Figure 4.4: **Features collisions and accuracy in Vowpal Wabbit hash table** Left: $k = 8$. Right: $k = 10$. These figures give median accuracy by taxon (blue) and collision rate (orange) for different hash table sizes from 18 bits (default) to 31 bits (maximal value allowed by Vowpal Wabbit). Dashed purple lines represent expected number of features ($\log_2(4^k \times K)$). Performances are obtained on fragments of length 400 with a mean coverage $c = 0.1$.

Influence of the mean coverage value

Figure 4.5 shows the performance reached by models based on fragments of length 200 (left) or 400 (right), for different values of k (horizontal axis) and different coverages (different colors). We first note that for $c = 0.1$, that is, for a limited number of fragments, the classification performance starts by increasing with the size of the k-mers (up to $k = 8$ and $k = 10$ for fragments of length 200 and 400, respectively), and subsequently decreases. This suggests that the number of fragments considered in this setting is not sufficient to efficiently learn when the dimensionality of the feature space becomes too large. Note that twice as many fragments of length 200 as fragments of size 400 are drawn for a given coverage value, which may explain why performance still increases beyond $k = 8$ with smaller fragments. Increasing the number of fragments confirms this hypothesis : performance systematically increases or remains steady with k for $c \geq 1$, and for $k \geq 8$, the performance is significantly higher than that obtained at $c = 0.1$, for both length of fragments. Increasing the coverage from $c = 1$ to $c = 10$ has a positive impact in both cases, although marginally for fragments of length 400. Further increasing the number of fragments does not bring any improvement.

Altogether, the optimal configuration on this *mini* dataset involves k-mers of size

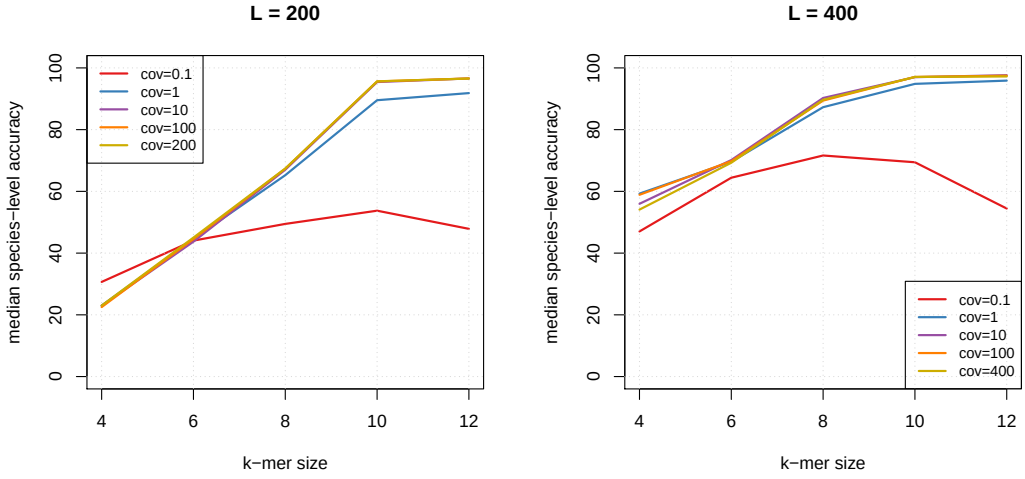

Figure 4.5: **Increasing the number of fragments and the k-mer size on the *mini* datasets**. Left: $L = 200$bp fragments. Right: $L = 400$bp fragments. These figures show median accuracy at species level for linear predictors trained with Vowpal Wabbit from fragments covering each reference genome with a mean coverage c from 0.1 (red) to L (purple). Performances are reported as a function of k-mer sizes.

12 and drawing fragments at a coverage $c \geq 10$ for the two lengths of fragments considered. Further increasing the size of the k-mers did not bring improvements, and actually proved to be challenging. Indeed, as mentioned above, VW proceeds by hashing the input features into a vector offering at most 2^{32} entries. This hashing operation can induce collisions between features, which can be detrimental to the model if the number of features becomes too high with respect to the size of the hash table. This issue is even more stressed in a multiclass setting, where the number of hash table entries available per model is divided by the number of classes considered. On this dataset, 51 models have to be stored in the hash table, which reduces the number of entries available per model to $2^{32}/51 \sim 2^{32-6} = 4^{13}$. We have empirically observed that performance could not increase for k greater than 12 and actually decreased for k-mers greater than 15, as shown in Figure 4.6.

Comparison with reference approaches

We now compare these results to two well-established approaches: a comparative approach based on the BWA-MEM sequence aligner [102] and a compositional approach based on the generative NB classifier [140]. The NB experiments rely on the FCP implementation [127] and are carried out in the same setting as VW: we compute profiles of k-mers abundance for the 356 genomes of the reference database, and use them to affect test fragments to their most likely genome. BWA-MEM is configured to solely return hits with maximal score (option `-T 0`). Unmapped fragments are counted as misclassifications, and a single hit is randomly picked in case of multiple hits, in order to obtain a species-level prediction. This random hit selection process is repeated 20

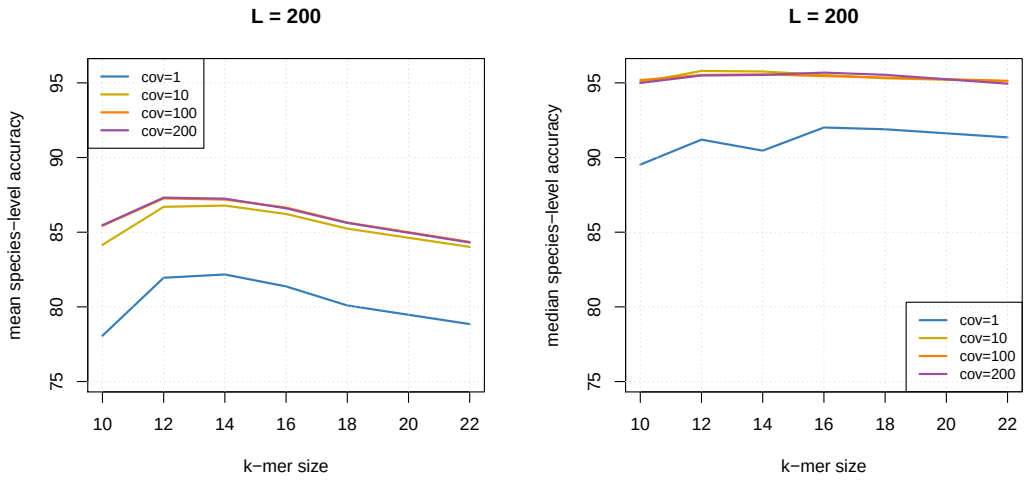

Figure 4.6: **Large k-mer sizes and collisions in hash table.** Mean species accuracy (right) and median species accuracy (left) given as a function of the k-mer size and the mean coverage.

times and the performance indicator reported below corresponds to its median value obtained across repetitions. Results are shown in Figure 4.7. We first note that k-mer based approaches, either generative or discriminative, never outperform the alignment-based approach. Comparable results are nevertheless obtained for $k \geq 10$ with VW, and $k = 12$ with the NB. Performances obtained for shorter k-mers are markedly lower than that obtained by BWA-MEM. We note finally that VW generally outperforms the NB classifier, except for small k-mers and short fragments ($k \leq 6$ and $L = 200$).

In summary, these experiments demonstrate the relevance and feasibility of large-scale machine learning for taxonomic binning: we obtain a performance comparable to that of the well-established alignment-based approach, provided a sufficient number of fragments and long enough k-mers are considered to learn the k-mers based predictive models.

4.4.2 Evaluation on the *small* and *large* reference databases

We now proceed to a more realistic evaluation involving a larger number of candidate microbial species and a larger number of reference genomes, using the *small* and *large* reference databases. We learn classification models according to the configuration suggested by the evaluation on the *mini* database: we consider k-mers of size 12 and a number of fragments allowing to cover each base of the reference genomes 10 times in average. We limit our analysis to fragments of length 200, which leads to learn models from around $n = 1.38 \times 10^8$ and $n = 2.56 \times 10^8$ fragments for the small and large reference databases, respectively. Note that due to the larger number of species involved, around $2^{32-8} = 4^{12}$ and $2^{32-10} = 4^{11}$ entries of the VW hash table are available per model for each of these reference databases. Based on the results with the *mini* database, this should be compatible with k-mers of size 12. We evaluate

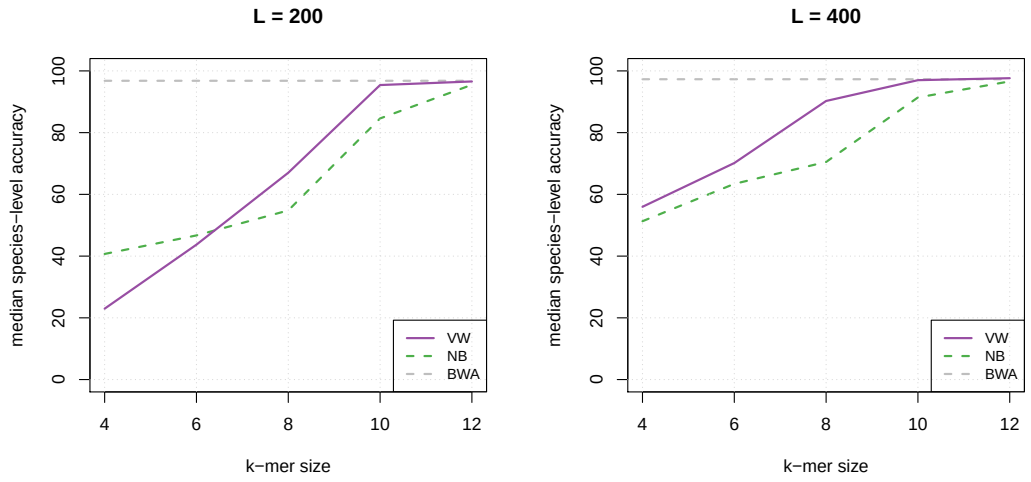

Figure 4.7: **Comparison between Vowpal Wabbit and reference methods on the *mini* datasets.** Left: $L = 200$bp fragments. Right: $L = 400$bp fragments. These figures give median accuracy by taxon for linear predictors (purple solid line) trained with VW from fragments covering each reference genome with a mean coverage equal to 10. Performances are reported as a function of k-mer sizes. Our approach is compared to standard compositional Naive Bayes approach (green dotted line), and alignment-based methods BWA (grey dotted line).

the performance of the models on fragments extracted from the 193 genomes of the validation database and draw a number of reads necessary to cover each base of each genome once in average, which represents around 3.5×10^6 sequences. Results obtained by VW are again compared to that obtained by the two baseline approaches involved in the previous proof of concept, namely BWA-MEM and NB, and are shown in Table 4.2. We first note that for the *small* reference database, the performance of VW and BWA-MEM are very similar (median species-level accuracy of 92.4% and 93%, respectively). The NB classifier, on the other hand, has a significantly lower performance, with 8% less in median accuracy than the alignment-based approach. Considering a larger number of candidate species in the *large* reference database has little impact on the alignment-based approach, where we observe a performance drop of only 1% (91.9% vs 93%). It impacts more severely compositional approaches, with both NB and VW accuracies dropping by about 5%. This suggests that k-mer based approach are still limited in their ability to deal with problems involving more than a few hundreds of candidate species.

4.4.3 Robustness to sequencing errors

The evaluation performed in the previous sections is based on taxonomic classification of DNA fragments drawn from reference genomes without errors. In real life, sequencing errors may alter the read sequences and make the classification problem more challenging. To evaluate the robustness of the classifiers to sequencing errors, we generate new

Table 4.2: **Performance on the *small* and *large* reference databases.** This table gives median accuracy by taxon for Vowpal Wabbit (VW), Naive Bayes (NB) and BWA-MEM by using the two reference databases smallDB and largeDB. Compositional approaches (VW and NB) performances are reported for a kmer length k equal to 12. VW performances are reported for a mean coverage $c = 10$.

	smallDB	largeDB
Vowpal Wabbit	92.4	87.7
Naive Bayes	85.1	79.8
BWA-MEM	93.0	91.9

reads simulating sequencing errors using the Grinder read simulation software [7]. We consider two types of sequencing errors models: homopolymeric stretches, which are commonly encountered in pyrosequencing technologies (e.g., Roche 454), and general mutations (substitutions and insertions/deletions). In order to be able to compare the results of the fragment- and read-based evaluations, we systematically simulate reads of length 200 (exactly), and simulate around 3.5×10^6 sequences as well.

Homopolymeric error models.

To evaluate the impact of homopolymeric errors, we consider the three error models implemented in Grinder : `Balzer` [12], `Richter` [133] and `Margulies` [112]. Results are shown in Figure 4.8. We first note that this kind of errors has a very limited impact on BWA-MEM: only the `Margulies` model turns out to be detrimental, with a drop of less than 1% for both the small and large reference databases. The `Balzer` and `Richter` models have a limited impact on the compositional approach as well: a drop of less than 1% is observed as well in most cases (except with the NB classifier, where a drop of almost 2% is observed using the large reference database and the `Richter` model). The `Margulies` model, on the other hand, has a much more severe impact on the performance of k-mer based approaches. While a relatively limited performance drop of around 1.5% is observed with VW using the small reference database, the NB shows a drop of more than 5%. Considering the large reference database, both approaches show a drop of almost 8%, which therefore leads to a gap of more than 10% and up to 20%, for VW and NB respectively, compared to the performance of the alignment-based approach. This discrepancy is therefore significantly higher than the one observed from fragments, where VW and NB have performance lower than that of BWA-MEM by around 4% and 12%, respectively, on the large reference database. Analyzing the error profile of the reads obtained by Grinder reveals that both the `Balzer` and `Richter` models lead to a median mutation rate of 0.5% (meaning that half of the 200 bp simulated reads show more than one modified base), while this rate raises to 3% with the `Margulies` model. While this can readily explain why this latter model had a stronger impact, it suggests that what may be seen as a relatively moderate modification of the sequences (6 bases out of 200) can have a severe impact on compositional approaches.

Mutation error model.

To study the impact of general mutation errors, we consider the 4th degree polynomial proposed by [94] and implemented in Grinder. Using the default values proposed by Grinder, we empirically observe a median mutation rate of 10.7%. This value is much more important than what is expected by current NGS technologies, and is probably due to the fact that this model was calibrated from shorter reads. Indeed, the median mutation rate decreases to around 1.5% when we reduce the length of the reads to 30, in agreement with the results of the original publication [94]. To investigate in details the impact of mutations within reads of length 200, we modify the parameters of the error model in order to gradually increase the median mutation rate from 1% to 10%, by 1%. This therefore leads to simulating 11 datasets, since we consider in addition the default Grinder configuration. Results are shown in Figure 4.9. We first note that this type of errors has a very limited impact on alignment-based approach: even at the higher rate of mutation considered (median mutation rate of 10.7%), the performance drops by around 1% with respect to the performance obtained with fragments, for both the small and large reference databases. On the other hand, the performance obtained with compositional approaches steadily decreases when the mutation rate increases. Using the small reference database, the impact is more severe for NB than for VW : a drop of up to 10% is observed in the former case (from 85.1% for fragments down to 75.2% for a mutation rate of 10.7%) and almost 6% in the latter (from 92.4% down to 86.7%). The drop is even more severe using the larger database in both cases. Interestingly while it remains relatively constant around 10% for mutation rate greater than 4% with NB (hence twice the gap observed between the small and large datasets using fragments), it keeps increasing with VW and reaches 24% at the highest mutation rate considered. As a result, VW is outperformed by NB using the large reference database for mutation rates greater than 8%. Although these extreme configurations are not realistic regarding the current state of the NGS technologies, we emphasize, in agreement with the previous experiment on homopolymeric errors, that significant drops are observed with compositional approaches for moderate mutation rates, especially for large number of candidate species. For instance, with a mutation rate of 2%, the performances of VW and NB drop respectively by 4 and 5% with the large reference database, while this has no impact on the alignment approach. In this more realistic setting, the alignment-based approach shows markedly higher performances: it provides a median species-level accuracy of 91.7%, while VW and the NB classifier reach 83.9% 74.8%, respectively.

4.4.4 Classification speed

Last but not least, we now turn to the comparison of the comparative and compositional approaches in terms of prediction time. This aspect is indeed of critical importance for the analysis of the large volumes of sequence data provided by next-generation

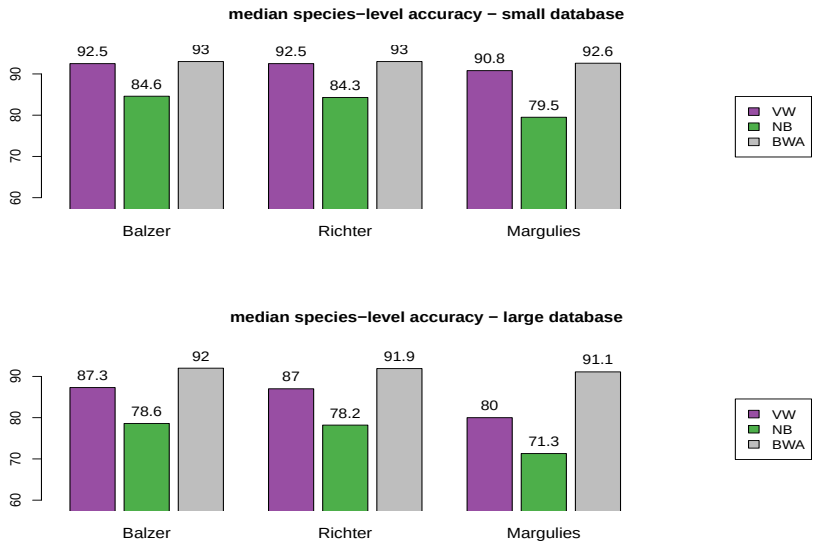

Figure 4.8: **Evaluation on FCP dataset: homopolymer-based models.** Top: smallDB reference. Bottom: largeDB reference. These figures give median accuracy by taxon for VW (purple), NB (green) and BWA (grey). Each approach has been evaluated on three datasets simulated according to three different error models: Balzer [12], Richter [133], and Margulies [112].

sequencing technologies, and constitutes the main motivation of resorting to k-mer based approaches. To perform this evaluation we measure the time taken by BWA-MEM and the k-mer based approaches to process the 30 test datasets involved in the previous experiments (1 fragments dataset, 3 reads datasets with homopolymeric errors and 11 reads datasets with mutation errors, for the two reference databases considered). This allows us to investigate the impact of the number of species involved in the reference database, as well as the amount of sequencing noise in the reads. We do not make a distinction between the two compositional approaches: both involve computing a score for each candidate species, defined as a dot product between the k-mer profile of the sequence to classify and a vector of weights obtained by training the model. To compute this dot-product efficiently, we implemented a procedure described in [153]. With this procedure, each A, T, G, C nucleotide is encoded by two bits, which allows to directly convert a k-mer as in integer between 0 and $4^k - 1$. Provided that the weight vector is loaded into memory, the score can be computed "on the fly" while evaluating the k-mer profile of the sequence to be classified, by adding the contribution of the current k-mer to the score. The drawback of this procedure lies in the fact that the vectors of weights defining the classification models need to be loaded into memory, which can be cumbersome in a multiclass setting. For 193 and 774 species and k-mers of size 12, this amounted to 12 and 48 gigabytes, respectively.

Computation times are measured on a single CPU (Intel XEON - 2.8 Ghz) equipped with 250 GB of memory, and summarized in Figure 4.10. The time needed to classify each read or fragment dataset by the k-mer approach shows little variation, for a given

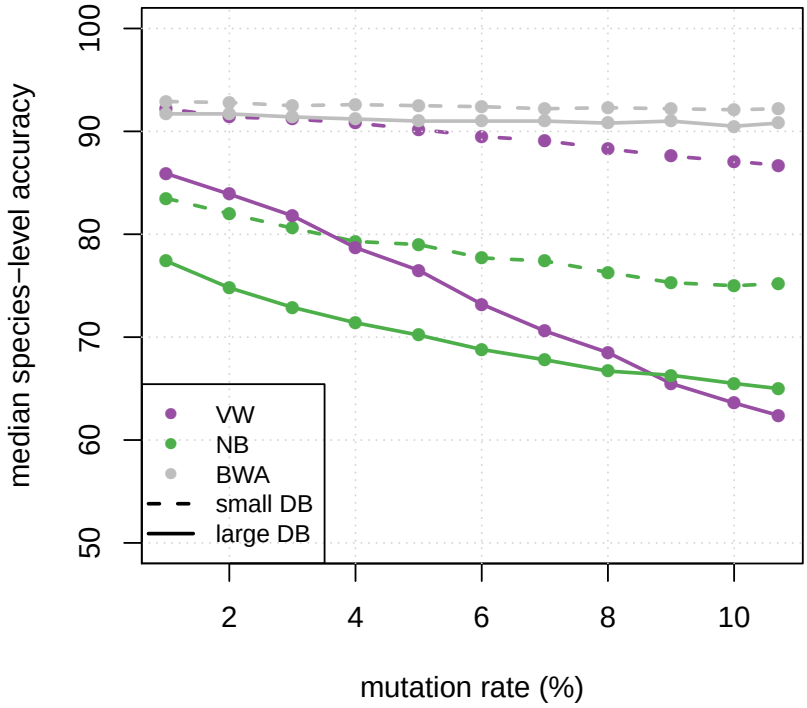

Figure 4.9: **Evaluation on FCP dataset: mutation-based models.** This figure gives median accuracy by taxon for VW (purple), NB (green) and BWA (grey), obtained with two reference databases: smallDB (dashed line) and largeDB (solid line). Each approach has been evaluated on 11 datasets simulated according to different mutation rates (from 1 to 10.7%) in the error model proposed in [94].

reference database. The median value obtained across test datasets reaches 5.4 and 9.1 minutes, using the small and large reference database, respectively, hence about a two-fold difference. This therefore amounts to classifying around 6.5×10^5 and 3.9×10^5 reads per minute, respectively. BWA-MEM shows a different behavior. We observe that the time varies more across reads and fragment datasets, and tends to increase with the amount of sequencing noise. On the other hand, the size of the reference database has a lesser impact, with at most an increase of 20% between the time needed to process a test dataset with the small or large reference databases. The compositional approach systematically offers shorter prediction times, with an improvement of 3 to almost 15 times, depending on the configuration.

4.5 Discussion

In this work, we investigate the potential of modern, large-scale machine learning approaches for taxonomic binning of metagenomics data. We extensively evaluate their performance when the scale of the problem increases regarding (i) the length of the k-mers considered to represent a sequence, (ii) the number of fragments used to learn the model, and (iii) the number of candidate species involved in the reference database. We also investigate in details their robustness to sequencing errors using simulated reads. We consider two baselines for this evaluation: a comparative approach based on the BWA-MEM sequence aligner and a compositional approach based on the generative NB classifier. We demonstrate in particular that increasing the number of fragments used to train the model has a significant impact on the accuracy of the model, and allows to estimate models based on longer k-mers. While this could be expected and is already highlighted by previous studies, the resulting configurations are out of reach of standard SVM implementations. We also show that discriminatively trained compositional models usually offer significantly higher performances than generative NB classifiers. The resulting models are competitive with well-established alignment tools for problems involving a small to moderate number of candidate species, and for reasonable amounts of sequencing errors. Our results suggest, however, that compositional approaches, both discriminative and generative, are still limited in their ability to deal with problems involving more than a few hundreds species. In this case, indeed, compositional approaches exhibit lower performance than alignment-based approaches and are much more negatively impacted by sequencing errors. Finally, we confirm that compositional approach achieve faster prediction times. This is indeed systematically the case in the various configurations listed above, with predictions obtained 3 to 15 times faster by compositional approaches, and, interestingly, depends on the number of candidate species and the level of sequencing noise. We emphasize, however, that fast predictions can only be obtained provided that the classification models are loaded in memory, hence for a memory footprint that scales linearly with the number of candidate species and exponentially with the size of the k-mers, which can become important for

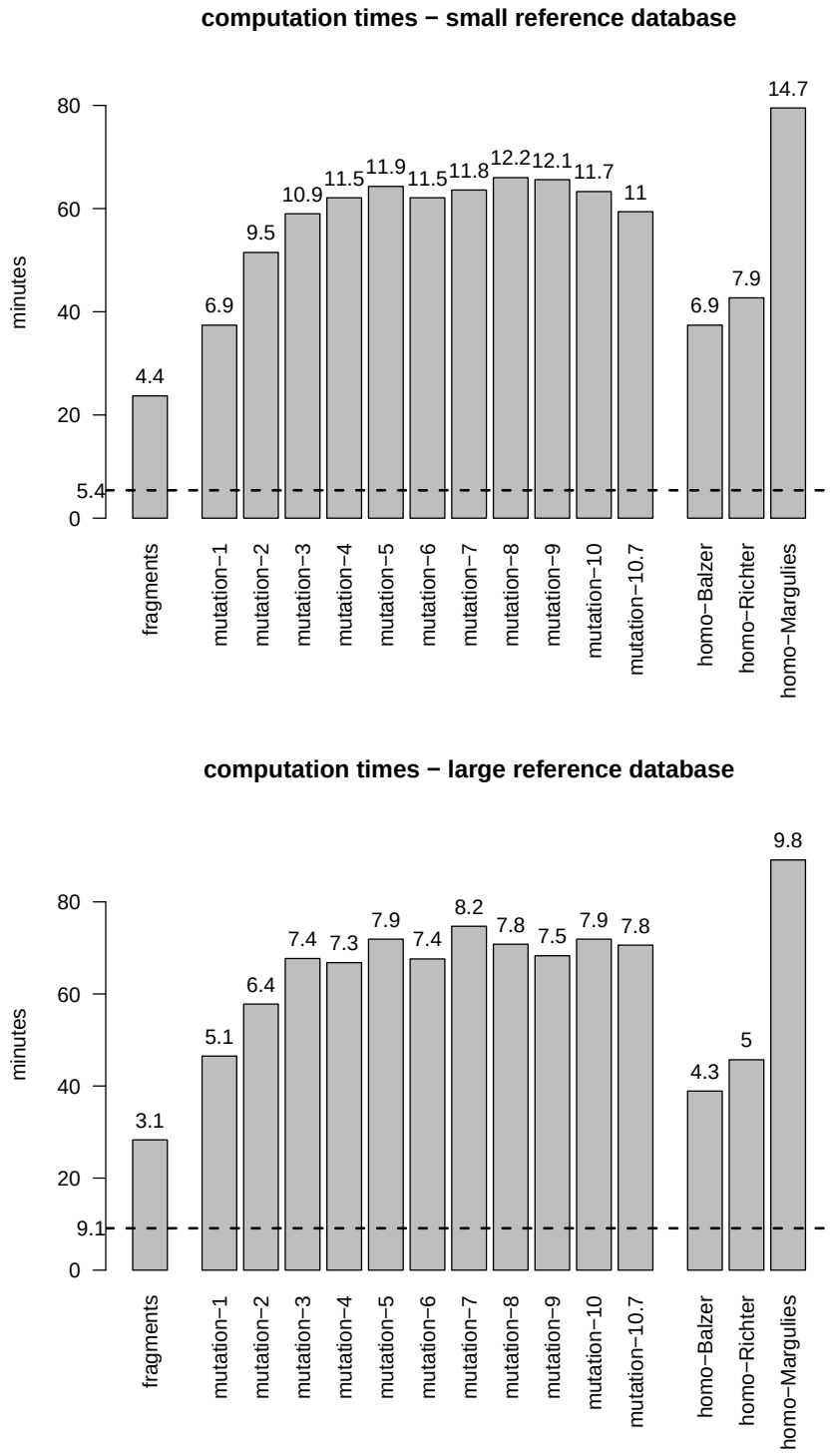

Figure 4.10: **Classification times.** The bars represent the time (in minutes) required to classify each fragment or read dataset using BWA-MEM, using the small (top) or large (bottom) reference database. The dashed horizontal lines represent the median time required by VW. The figures shown on top of the bars represent the ratio between the times taken by BWA-MEM and VW.

large reference databases and long k-mers.

At least three simple extensions could be envisioned to make compositional approaches more competitive in accuracy with the alignment-based approach, faster, and to limit their memory footprint. First, the robustness to sequencing errors may be improved by learning models from simulated reads instead of fragments. This could indeed allow to tune the model to the sequencing technology producing the reads to be analyzed, provided its error model is properly known and characterized. Second, introducing a sparsity-inducing penalty while learning the model would have the effect of reducing the number of features entering the model, hence to reduce the memory footprint required to load the model into memory. Finally, alternative strategies, known as error correcting tournaments [20], could be straightforwardly considered to reduce the number of models to learn, hence to store into memory during prediction, to address a multiclass problem. Our results indeed suggest that addressing these issues is critical to build state-of-the-art compositional classifiers to analyze metagenomics samples that may involve a broad spectrum of species. We emphasize however that such large scale models can remain competitive for realistic amounts of sequencing errors and a moderate number of species (around 200 in our study), hence can already be useful in cases where the number of species that can be encountered is limited, which may in particular be the case for diagnostic applications involving specific types of specimens.

Chapter 5

Discussion

In this chapter, I propose an overall discussion on some points that came to my mind these past three years, regarding the different problems I tackled, like the choice of a performance indicator or the concept of microbial species. I also give some thoughts on possible extensions of the studies presented, like using orthogonality constraints in hierarchically structured learning tasks.

Performance indicator for classification tasks. When conducting MS and metagenomics studies, we select multiple performance indicators and observe that conclusions change from one indicator to another. In our case, there was a large class imbalance for MS data (and also for metagenomics), with the largest class (*Escherichia coli*) represented by 5 times more mass-spectra than the smallest class (*Bacillus thuringiensis*). Because we use a cross-validation strategy for model learning, the bias in class representation is reflected in the final accuracy score. For medical applications, one may argue that imbalance in the training reference database mimics the clinical interest for some micro-organisms more than for others. In this context, a indicator like overall good classification rate, also called micro-accuracy, could be favorable. Indeed, if most of the database is covered by pathogens, a predictive model that achieves a high percentage of well-identified data will be efficient for diagnostics purposes.

However, often the class imbalance in a reference database is not representative of a particular clinical interest, but more a result of many factors such as the availability of sequenced genomes. Therefore we relied on other performance indicators (see Section 1.4) in our studies involving MS and metagenomics for bacterial identification. We considered indicators that measure the mean behavior of a predictive model, putting the same weight to all classes. We did not use any prior knowledge, like class importance from a clinical point of view. Interestingly, we observed that SVM-based approaches allow to apply different weights during the optimization process (e.g., by considering class-specific C regularization constants), which allows to both inject prior and control dataset class imbalance. Moreover, we recalled that cost-sensitive penalized SVMs (Treeloss and Structured) use variable class-to-class penalization contained in a distance matrix Δ. We empirically observed that one can manually tune this matrix to

arbitrarily bias the predictive model. For instance, in the intra-genus confusions (e.g., *Bacillus* genus), increasing the Δ matrix weight related to *Bacillus cereus* and *Bacillus thuringiensis*, from 2 (tree distance) to a large value (e.g., 100) leads to a classifier that will always predict *Bacillus cereus* when the input data belongs to the genus *Bacillus*. In this case, *Bacillus cereus* is a known pathogen agent where *Bacillus thuringiensis* is not harmful for human. These preliminary experiments could be extended to every case where confusions between pathogen and non-pathogen micro-organisms. From a clinical standpoint, mistaking a non-pathogen microbe for a pathogen is a less severe error than the reverse case.

The concept of microbial species and its impact on classification tasks. In the supervised learning context, and in particular for classification tasks, algorithms rely on a given dataset made of example-label pairs (x,y). In Chapters 2 and 4, we consider the information at the species-level for class labeling. In each study performed in this thesis, we rely on microbial taxonomies, like NCBI RefSeq, and consider the taxonomic nodes for our classes definition. However, conclusions made in MicroMass dataset about intra-genus confusions underline that, sometimes, microbial species can be close enough to be mistaken by using a given type of data (e.g., proteomics, genomics). For instance, the particular case of the confusions between *Escherichia coli* and *Shigella* genus is a good illustration. Indeed, those two genera are distinguished by their pathogenicity and actually belong to the same genus [187]. This raises the more general question of how to define a microbial species. Can one use the same rule for microbes and multicellular organisms ? There is a large scale difference between those two worlds and the diversity covered by known microbes is greater than all the animals and plants. Furthermore, our understanding of the microbial world is barely bounded to cultivable organisms and we do not know how much it can be extrapolated. Today, naming conventions for microbial species rely on a strict formalism based on thresholds on similarities computed on 16S rRNA marker or DNA-DNA Hybridization (DDH, [117]). This system is very convenient and forms a common way to describe microbes. However, it appears that such arbitrary cut-offs (DDH$>$ 70% or 16S rRNA$>$ 97%) are not well-suited and should change in function of micro-organisms. For instance, [83] demonstrated that a threshold value equal to 99% of 16S rRNA similarity should be used, instead of 97%. Another issue met in current taxonomies is the over-specification of certain microbial species based on phenotypic or pathotypic characteristics. Historically, Ferdinand Cohn (1872) demonstrated that the paradigm proposed by Carl Linnaeus for animals and plants could be applied to bacteria and that they could be divided into genera and species. At the beginning of microbial taxonomy, most of research was dedicated to medical concerns. One may observe that the most popular pathogen agents were discovered during the 19th century. During this period, microbiologists used features like pathogenic potential for identifying a new microbial species. This second problem can be illustrated by the *Bacillus anthracis*, *Bacillus cereus* and

Bacillus thuringiensis case. Here, one bacterium is a potential biological weapon causing the fatal disease anthrax, the second one is a soil bacterium and an opportunistic pathogen through food poisoning and the last one is used in pesticides production. It has been demonstrated [71] that these 3 distinct bacterial species are actually members of the same group following DDH criterion, but due to their different pathogenicity and thus a variable clinical interest, 3 different species names have been created. In summary, modern molecular techniques have enabled to measure finer distinctions between micro-organisms, but the microbiology community has still not agreed on a set of tests required to describe a new microbial species. Furthermore, the fact that some new micro-organisms do not grow on culture media is not yet taken into account in the standard criteria. In the classification context, it could be relevant to consider a post-treatment of the defined classes, according to primary results, and eventually, merge the classes that are too close to be discriminated by the technology that acquires the data.

Using orthogonality constraints in a hierarchical scheme. Considering orthogonality constraints during the model learning is not necessarily very straightforward. In the work present in Chapter 3, we began with the study of [184] and focus on the use of their penalty in a hierarchical classification context. They propose to add orthogonality constraints in a divide-and-conquer setting, where the predictors are tree-structured, and claim that their formulation avoids low-rank confusions, like the within-genus classification errors we observed in Chapter 2. However, I think that the way they are penalizing their predictors is not the most intuitive. Indeed, they enforce predictors belonging to the same tree path (e.g., parent and child nodes) to share the less possible information and features. Given that such a hierarchical structure, like a taxonomy, relies on generalization-specialization relations, I think that a better formulation could be the following one. Instead of considering one model w at each node that is able to predict one of its children nodes, I would rather consider a model matrix W at each tree node, where each column corresponds to a binary classifier for each child node. By enforcing orthogonality between the columns of each W matrix, the learning process would lead to classifiers at a given node that rely on features specific to one child node.

Taxonomic binning and clinical applications. Taxonomic binning does not necessarily consist in assigning sequencing reads to the taxonomic species level, as we do in Chapter 4. Indeed, most binning approaches proposed so far proceed with a so-called *rank-flexible* affectation mechanism, meaning that each sequencing read can be assigned to the most suitable taxonomic rank according to multiple alignment hits for instance. Most studies which compare such rank-flexible approaches conclude that a vast majority of affected reads are assigned to high taxonomic ranks, such as phylum or class. Even if clinical metagenomics studies demonstrate links between diseases and presence/absence of a whole phylum (e.g., Firmicutes and inflammatory bowel disease

[123]), we consider that the first clinical use of metagenome sequencing data would be a precise analysis of the microbial species present in the sample. In particular, we propose a read-by-read affectation at the species-level in order to reduce the computational burden for the next steps. These steps mostly rely on aligning each read to a reference genome database, in order to detect signs of drug resistance or virulence factors. In addition, rank-flexible approaches are also useful in a 'novelty detection' context, where one expects that some micro-organisms present in the sample are absent of the reference genome database. This mainly occurs when studying environmental samples, such as ocean samples [80], and leads to the addition of new high-rank nodes (e.g., class) in the taxonomy. However, this will be rarely the case in clinical studies, because most of the known pathogenic agents have already been sequenced. Even if a new micro-organism is present in a clinical sample, it will likely be a member of an already known genus and an affectation at the species level will not be a severe classification error.

Bibliography

[1] Alekh Agarwal, Olivier Chapelle, Miroslav Dudík, and John Langford. A reliable effective terascale linear learning system. *The Journal of Machine Learning Research*, 15(1):1111–1133, 2014.

[2] Ben Aisen. A comparison of multi class svm methods, 2006.

[3] A Aizerman, Emmanuel M Braverman, and LI Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.

[4] C Aldridge, PW Jones, S Gibson, J Lanham, M Meyer, R Vannest, and R Charles. Automated microbiological detection/identification system. *Journal of clinical microbiology*, 6(4):406–413, 1977.

[5] Mohamed Aly. Survey on multiclass classification methods. *Neural Netw*, pages 1–9, 2005.

[6] Rudolf I Amann, Wolfgang Ludwig, and Karl-Heinz Schleifer. Phylogenetic identification and in situ detection of individual microbial cells without cultivation. *Microbiological reviews*, 59(1):143–169, 1995.

[7] Florent E Angly, Dana Willner, Forest Rohwer, Philip Hugenholtz, and Gene W Tyson. Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic acids research*, 40(12):e94–e94, 2012.

[8] John P Anhalt and Catherine Fenselau. Identification of bacteria using mass spectrometry. *Analytical Chemistry*, 47(2):219–225, 1975.

[9] Andreas Argyriou, Rina Foygel, and Nathan Srebro. Sparse prediction with the k-support norm. In *Advances in Neural Information Processing Systems*, pages 1457–1465, 2012.

[10] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012.

[11] Bart Bakker and Tom Heskes. Task clustering and gating for bayesian multitask learning. *The Journal of Machine Learning Research*, 4:83–99, 2003.

[12] Susanne Balzer, Ketil Malde, Anders Lanzén, Animesh Sharma, and Inge Jonassen. Characteristics of 454 pyrosequencing data enabling realistic simulation with flowsim. *Bioinformatics*, 26(18):i420–i425, 2010.

[13] Samuel Baron and John A Washington. Principles of diagnosis. 1996.

[14] Alexander Barvinok. *A course in convexity*, volume 54. American Mathematical Soc., 2002.

[15] Sergei G Bavykin, Yuri P Lysov, Vladimir Zakhariev, John J Kelly, Joany Jackman, David A Stahl, and Alexey Cherni. Use of 16s rrna, 23s rrna, and gyrb gene sequence analysis to determine phylogenetic relationships of bacillus cereus group microorganisms. *Journal of clinical microbiology*, 42(8):3711–3730, 2004.

[16] Jonathan Baxter. A model of inductive bias learning. *J. Artif. Intell. Res.(JAIR)*, 12:149–198, 2000.

[17] Cinzia Benagli, Viviana Rossi, Marisa Dolina, Mauro Tonolla, and Orlando Petrini. Matrix-assisted laser desorption ionization-time of flight mass spectrometry for the identification of clinically relevant bacteria. *PLoS One*, 6(1):e16424, 2011.

[18] Younes Bennani and Khalid Benabdeslem. Dendogram-based svm for multi-class classification. *CIT. Journal of computing and information technology*, 14(4):283–289, 2006.

[19] David R Bentley, Shankar Balasubramanian, Harold P Swerdlow, Geoffrey P Smith, John Milton, Clive G Brown, Kevin P Hall, Dirk J Evers, Colin L Barnes, Helen R Bignell, et al. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456(7218):53–59, 2008.

[20] Alina Beygelzimer, John Langford, and Pradeep Ravikumar. Error-correcting tournaments. In *Algorithmic Learning Theory*, pages 247–262. Springer, 2009.

[21] Hemant Bhatta, Ewa M Goldys, and Robert P Learmonth. Use of fluorescence spectroscopy to differentiate yeast and bacterial cells. *Applied microbiology and biotechnology*, 71(1):121–126, 2006.

[22] Peter J Bickel, Ya'acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, pages 1705–1732, 2009.

[23] Adrian Bignami. *Economic potential for clinically significant in vitro diagnostics*. PhD thesis, Massachusetts Institute of Technology, 2009.

[24] A Bizzini and G Greub. Matrix-assisted laser desorption ionization time-of-flight mass spectrometry, a revolution in clinical microbial identification. *Clinical Microbiology and infection*, 16(11):1614–1619, 2010.

[25] Jonathan M Borwein and Adrian S Lewis. *Convex analysis and nonlinear optimization: theory and examples*, volume 3. Springer, 2010.

[26] Léon Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17:9, 1998.

[27] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[28] Léon Bottou and Yann Le Cun. On-line learning for very large data sets. *Applied Stochastic Models in Business and Industry*, 21(2):137–151, 2005.

[29] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2009.

[30] Erin J Bredensteiner and Kristin P Bennett. Multicategory classification by support vector machines. In *Computational Optimization*, pages 53–79. Springer, 1999.

[31] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[32] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[33] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[34] Leo Breiman and Philip Spector. Submodel selection and evaluation in regression. the x-random case. *International statistical review/revue internationale de Statistique*, pages 291–319, 1992.

[35] Louis Brickman. On the field of values of a matrix. *Proceedings of the American Mathematical Society*, 12(1):61–66, 1961.

[36] Petronella Catharina Adriana Maria Buijtels, HFM Willemse-Erix, PLC Petit, HP Endtz, GJ Puppels, HA Verbrugh, A Van Belkum, Dick van Soolingen, and Kees Maquelin. Rapid identification of mycobacteria by raman spectroscopy. *Journal of clinical microbiology*, 46(3):961–965, 2008.

[37] Mary T Cafferkey, Avril Sloane, Siobhan McCrae, and CA O'Morain. Yersinia frederiksenii infection and colonization in hospital staff. *Journal of Hospital Infection*, 24(2):109–115, 1993.

[38] Lijuan Cai and Thomas Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 78–87. ACM, 2004.

[39] Andrew J Calder, A Mike Burton, Paul Miller, Andrew W Young, and Shigeru Akamatsu. A principal component analysis of facial expressions. *Vision research*, 41(9):1179–1208, 2001.

[40] Rich Caruana. *Multitask learning*. Springer, 1998.

[41] Venkat Chandrasekaran, Benjamin Recht, Pablo A Parrilo, and Alan S Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[42] Chin-Liang Chang. Finding prototypes for nearest neighbor classifiers. *Computers, IEEE Transactions on*, 100(11):1179–1184, 1974.

[43] Abdessalam Cherkaoui, Jonathan Hibbs, Stéphane Emonet, Manuela Tangomo, Myriam Girard, Patrice Francois, and Jacques Schrenzel. Comparison of two matrix-assisted laser desorption ionization-time of flight mass spectrometry methods with conventional phenotypic identification for routine identification of bacteria to the species level. *Journal of clinical microbiology*, 48(4):1169–1175, 2010.

[44] Anna Choromanska and John Langford. Logarithmic time online multiclass prediction. *CoRR*, abs/1406.1822, 2014.

[45] Francesca D Ciccarelli, Tobias Doerks, Christian Von Mering, Christopher J Creevey, Berend Snel, and Peer Bork. Toward automatic reconstruction of a highly resolved tree of life. *science*, 311(5765):1283–1287, 2006.

[46] Jill E Clarridge. Impact of 16s rrna gene sequence analysis for identification of bacteria on clinical microbiology and infectious diseases. *Clinical microbiology reviews*, 17(4):840–862, 2004.

[47] Martin A Claydon, Simon N Davey, Valeria Edwards-Jones, and Derek B Gordon. The rapid identification of intact microorganisms using mass spectrometry. *Nature biotechnology*, 14(11):1584–1586, 1996.

[48] Ronald Cole and Mark Fanty. Spoken letter recognition. In *Proc. Third DARPA Speech and Natural Language Workshop*, pages 385–390, 1990.

[49] Kevin R Coombes, Keith A Baggerly, and Jeffrey S Morris. Pre-processing mass spectrometry data. In *Fundamentals of Data Mining in Genomics and Proteomics*, pages 79–102. Springer, 2007.

[50] Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[51] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[52] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

[53] Antony Croxatto, Guy Prod'hom, and Gilbert Greub. Applications of maldi-tof mass spectrometry in clinical diagnostic microbiology. *FEMS microbiology reviews*, 36(2):380–407, 2012.

[54] Katrien De Bruyne, Bram Slabbinck, Willem Waegeman, Paul Vauterin, Bernard De Baets, and Peter Vandamme. Bacterial species identification from maldi-tof mass spectra through data analysis and machine learning. *Systematic and applied microbiology*, 34(1):20–29, 2011.

[55] John E Dennis and Robert B Schnabel. Numerical methods for unconstrained optimization and nonlinear equations, 1983. *Reprinted as Classics in Applied Mathematics*, 16, 1996.

[56] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *arXiv preprint cs/9501101*, 1995.

[57] Erik R Dubberke, Dale N Gerding, David Classen, Kathleen M Arias, Kelly Podgorny, Deverick J Anderson, Helen Burstin, David P Calfee, Susan E Coffin, Victoria Fraser, et al. Strategies to prevent clostridium difficile infections in acute care hospitals. *Infection Control*, 29(S1):S81–S92, 2008.

[58] Susan Dumais and Hao Chen. Hierarchical classification of web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 256–263. ACM, 2000.

[59] John Eid, Adrian Fehr, Jeremy Gray, Khai Luong, John Lyle, Geoff Otto, Paul Peluso, David Rank, Primo Baybayan, Brad Bettman, et al. Real-time dna sequencing from single polymerase molecules. *Science*, 323(5910):133–138, 2009.

[60] U Eigner, A Schmid, U Wild, D Bertsch, and A-M Fahr. Analysis of the comparative workflow and performance characteristics of the vitek 2 and phoenix systems. *Journal of clinical microbiology*, 43(8):3829–3834, 2005.

[61] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. In *Journal of Machine Learning Research*, pages 615–637, 2005.

[62] Olivier Gaillot, Nicolas Blondiaux, Caroline Loïez, Frédéric Wallet, Nadine Lemaître, Stéphanie Herwegh, and René J Courcol. Cost-effectiveness of switch to matrix-assisted laser desorption ionization-time of flight mass spectrometry for routine bacterial identification. *Journal of clinical microbiology*, 49(12):4412–4412, 2011.

[63] C Gini. Concentration and dependency ratios. *Rivista di Politica Economica*, 87:769–792, 1997.

[64] Kristen Grauman, Fei Sha, and Sung J Hwang. Learning a tree of metrics with disjoint visual features. In *Advances in Neural Information Processing Systems*, pages 621–629, 2011.

[65] The Lewin Group. The value of diagnostics innovation, adoption and diffusion into health care. 2005.

[66] United Health Group. Personalized medicine: Trends and prospects for the new science of genetic testing and molecular diagnostics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 2012.

[67] THOMAS LARRY Hale. Genetic basis of virulence in shigella species. *Microbiological reviews*, 55(2):206–224, 1991.

[68] Richard W Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950.

[69] Trevor Hastie, Robert Tibshirani, et al. Classification by pairwise coupling. *The annals of statistics*, 26(2):451–471, 1998.

[70] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[71] Erlendur Helgason, Ole Andreas Økstad, Dominique A Caugant, Henning A Johansen, Agnes Fouet, Michéle Mock, Ida Hegna, and Anne-Brit Kolstø. Bacillus anthracis, bacillus cereus, and bacillus thuringiensisâĂŤone species on the basis of genetic evidence. *Applied and environmental microbiology*, 66(6):2627–2630, 2000.

[72] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[73] Thomas Hofmann, Lijuan Cai, and Massimiliano Ciaramita. Learning with taxonomies: Classifying documents and words. In *NIPS workshop on syntax, semantics, and statistics*, 2003.

[74] JG Holt, NR Krieg, PHA Sneath, et al. Bergey's manual of systematic bacteriology, vol. 1. *The Williams and Wilkins Co., Baltimore*, 1984.

[75] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[76] Jan Hudzicki. Kirby-bauer disk diffusion susceptibility test protocol. *American society for microbiology*, 2012.

[77] Philip Hugenholtz et al. Exploring prokaryotic diversity in the genomic era. *Genome Biol*, 3(2):1–0003, 2002.

[78] Philip Hugenholtz, Brett M Goebel, and Norman R Pace. Impact of culture-independent studies on the emerging phylogenetic view of bacterial diversity. *Journal of bacteriology*, 180(18):4765–4774, 1998.

[79] Daniel H Huson, Alexander F Auch, Ji Qi, and Stephan C Schuster. MEGAN analysis of metagenomic data. *Genome research*, 17(3):377–386, 2007.

[80] Vaughn Iverson, Robert M Morris, Christian D Frazar, Chris T Berthiaume, Rhonda L Morales, and E Virginia Armbrust. Untangling genomes from metagenomes: revealing an uncultured class of marine euryarchaeota. *Science*, 335(6068):587–590, 2012.

[81] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 433–440. ACM, 2009.

[82] Laurent Jacob and Jean-Philippe Vert. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics*, 24(19):2149–2156, 2008.

[83] J Michael Janda and Sharon L Abbott. 16s rrna gene sequencing for bacterial identification in the diagnostic laboratory: pluses, perils, and pitfalls. *Journal of clinical microbiology*, 45(9):2761–2764, 2007.

[84] Thorsten Joachims. *Advances in Kernel Methods: Making large-Scale SVM Learning Practical.* MIT Press, 1999.

[85] Thorsten Joachims. *Learning to classify text using support vector machines: Methods, theory and algorithms*. Kluwer Academic Publishers, 2002.

[86] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

[87] James R Johnson. Shigella and escherichia coli at the crossroads: machiavellian masqueraders or taxonomic treachery? *Journal of medical microbiology*, 49(7):583, 2000.

[88] Miyuki Kamachi, Michael Lyons, and Jiro Gyoba. The japanese female facial expression (jaffe) database. *URL http://www. kasrl. org/jaffe. html*, 21, 1998.

[89] James B Kaper, James P Nataro, and Harry LT Mobley. Pathogenic escherichia coli. *Nature Reviews Microbiology*, 2(2):123–140, 2004.

[90] Yoshiaki Kawamura, Xiao-Gang Hou, Ferdousi Sultana, Hiroaki Miura, and Takayuki Ezaki. Determination of 16s rrna sequences of streptococcus mitis and streptococcus gordonii and phylogenetic relationships among members of the genus streptococcus. *International journal of systematic bacteriology*, 45(2):406–408, 1995.

[91] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, volume 14, pages 1137–1145, 1995.

[92] A Kolmogorov. Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, 4:83âĂŞ–91, 1933.

[93] Konstantinos T Konstantinidis and James M Tiedje. Genomic insights that advance the species definition for prokaryotes. *Proceedings of the National Academy of Sciences of the United States of America*, 102(7):2567–2572, 2005.

[94] Jan O Korbel, Alexej Abyzov, Xinmeng Jasmine Mu, Nicholas Carriero, Philip Cayting, Zhengdong Zhang, Michael Snyder, and Mark B Gerstein. Pemer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biol*, 10(2):R23, 2009.

[95] David Koslicki, Simon Foucart, and Gail Rosen. Wgsquikr: Fast whole-genome shotgun metagenomic classification. *PloS one*, 9(3):e91784, 2014.

[96] Bernard La Scola. Intact cell maldi-tof mass spectrometry-based approaches for the diagnosis of bloodstream infections. 2011.

[97] Ruiting Lan and Peter R Reeves. *Escherichia coli* in disguise: molecular origins of shigella. *Microbes and infection*, 4(11):1125–1132, 2002.

[98] John Langford, Lihong Li, and Alexander Strehl. Vowpal Wabbit open source project. Technical report, Technical Report, Yahoo, 2007.

[99] Joanne M Langley, John C LeBlanc, Martha Hanakowski, and Olga Goloubeva. The role of clostridium difficile and viruses as causes of nosocomial diarrhea in children. *Infection Control*, 23(11):660–664, 2002.

[100] Peter Lasch, Wolfgang Beyer, Herbert Nattermann, Maren Stämmler, Enrico Siegbrecht, Roland Grunow, and Dieter Naumann. Identification of bacillus anthracis by using matrix-assisted laser desorption ionization-time of flight mass spectrometry and artificial neural networks. *Applied and environmental microbiology*, 75(22):7229–7242, 2009.

[101] Yann LeCun, Leon Bottou, Genevieve B Orr, and Klaus-Robert Muller. Neural networks-tricks of the trade. *Springer Lecture Notes in Computer Sciences*, 1524:5–50, 1998.

[102] Heng Li. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997*, 2013.

[103] Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[104] Carl von Linnaeus. 1735. systema naturae, sive regna tria naturae systematice proposita per classes, ordines, genera, & species. *Lugduni Batavorum: de Groot. Staffan Müller-Wille*, 1964.

[105] László Lovász and Alexander Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

[106] Ian M Mackay. Real-time pcr in the microbiology laboratory. *Clinical Microbiology and Infection*, 10(3):190–212, 2004.

[107] Michele Magrane, UniProt Consortium, et al. Uniprot knowledgebase: a hub of integrated protein data. *Database*, 2011:bar009, 2011.

[108] Pierre Mahé, Maud Arsac, Sonia Chatellier, Valérie Monnin, Nadine Perrot, Sandrine Mailler, Victoria Girard, Mahendrasingh Ramjeet, Jérémy Surre, Bruno Lacroix, et al. Automatic identification of mixed bacterial species fingerprints in a maldi-tof mass-spectrum. *Bioinformatics*, page btu022, 2014.

[109] Sharmila S Mande, Monzoorul Haque Mohammed, and Tarini Shankar Ghosh. Classification of metagenomic sequences: methods and challenges. *Briefings in bioinformatics*, page bbs054, 2012.

[110] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[111] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.

[112] Marcel Margulies, Michael Egholm, William E Altman, Said Attiya, Joel S Bader, Lisa A Bemben, Jan Berka, Michael S Braverman, Yi-Ju Chen, Zhoutao Chen, et al. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, 2005.

[113] D. Martiny. Comparison of the microflex lt and vitek ms systems for routine identification of bacteria by matrix-assisted laser desorption ionization-time of flight mass spectrometry. *Journal of Clinical Microbiology*, 50(1):1313–1325, 2012.

[114] Ward Douglas Maurer and Theodore Gyle Lewis. Hash table methods. *ACM Computing Surveys (CSUR)*, 7(1):5–19, 1975.

[115] Ernst Mayr et al. Animal species and evolution. *Animal species and their evolution.*, 1963.

[116] Andrew McCallum, Ronald Rosenfeld, Tom M Mitchell, and Andrew Y Ng. Improving text classification by shrinkage in a hierarchy of classes. In *ICML*, volume 98, pages 359–367, 1998.

[117] BJ McCarthy and ET Bolton. An approach to the measurement of genetic relatedness among organisms. *Proceedings of the National Academy of Sciences of the United States of America*, 50(1):156, 1963.

[118] Alice Carolyn McHardy, Hector Garcia Martin, Aristotelis Tsirigos, Philip Hugenholtz, and Isidore Rigoutsos. Accurate phylogenetic classification of variable-length DNA fragments. *Nature methods*, 4(1):63–72, 2007.

[119] Earl H McKinney. Generalized birthday problem. *American Mathematical Monthly*, pages 385–387, 1966.

[120] Alexander Mellmann, Dag Harmsen, Craig A Cummings, Emily B Zentz, Shana R Leopold, Alain Rico, Karola Prior, Rafael Szczepanowski, Yongmei Ji, Wenlan Zhang, et al. Prospective genomic characterization of the german enterohemorrhagic escherichia coli o104: H4 outbreak by rapid next generation sequencing technology. *PloS one*, 6(7):e22751, 2011.

[121] Ruth Miller, Vincent Montoya, Jennifer Gardy, David Patrick, and Patrick Tang. Metagenomics for pathogen detection in public health. *Genome Medicine*, 5(9):81, 2013.

[122] LH MOORE, EC MOORE, RGE MURRAY, E STACKEBRANDT, and MP STARR. Report of the ad hoc committee on reconciliation of approaches to bacterial systematics. *International Journal of Systematic Bacteriology*, pages 463–464, 1987.

[123] Xochitl C Morgan, Timothy L Tickle, Harry Sokol, Dirk Gevers, Kathryn L Devaney, Doyle V Ward, Joshua A Reyes, Samir A Shah, Neal LeLeiko, Scott B Snapper, et al. Dysfunction of the intestinal microbiome in inflammatory bowel disease and treatment. *Genome Biol*, 13(9):R79, 2012.

[124] Donald F Morrison. Multivariate statistical methods. 3. *New York, NY. Mc*, 1990.

[125] Guillaume Obozinski, Ben Taskar, and Michael I Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.

[126] Norman R Pace. A molecular view of microbial diversity and the biosphere. *Science*, 276(5313):734–740, 1997.

[127] Donovan Parks, Norman MacDonald, and Robert Beiko. Classifying short genomic fragments from novel lineages using composition and homology. *BMC Bioinformatics*, 12(1):328, 2011.

[128] Kaustubh Raosaheb Patil, Linus Roune, and Alice Carolyn McHardy. The PhyloPythiaS web server for taxonomic assignment of metagenome sequences. *PloS one*, 7(6):e38581, 2012.

[129] Jane Peterson, Susan Garges, Maria Giovanni, Pamela McInnes, Lu Wang, Jeffery A Schloss, Vivien Bonazzi, Jean E McEwen, Kris A Wetterstrand, Carolyn Deal, et al. The nih human microbiome project. *Genome research*, 19(12):2317–2323, 2009.

[130] Kim D Pruitt, Tatiana Tatusova, Garth R Brown, and Donna R Maglott. Ncbi reference sequences (refseq): current status, new features and genome annotation policy. *Nucleic acids research*, 40(D1):D130–D135, 2012.

[131] Piyush Rai, Hal Daumé III, and Suresh Venkatasubramanian. Streamed learning: One-pass svms. In *IJCAI*, volume 9, pages 1211–1216, 2009.

[132] Jason D Rennie, Lawrence Shih, Jaime Teevan, David R Karger, et al. Tackling the poor assumptions of naive bayes text classifiers. In *ICML*, volume 3, pages 616–623. Washington DC), 2003.

[133] Daniel C Richter, Felix Ott, Alexander F Auch, Ramona Schmid, and Daniel H Huson. MetasimâĂŤa sequencing simulator for genomics and metagenomics. *PloS one*, 3(10):e3373, 2008.

[134] Christian S Riesenfeld, Patrick D Schloss, and Jo Handelsman. Metagenomics: genomic analysis of microbial communities. *Annu. Rev. Genet.*, 38:525–552, 2004.

[135] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *The Journal of Machine Learning Research*, 5:101–141, 2004.

[136] Ryan Rifkin and Aldebaro Klautau. Parallel networks that learn to pronounce english text. *Journal of Machine Learning Research*, pages 101–141, 2004.

[137] R Tyrrell Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1997.

[138] R Tyrrell Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1997.

[139] Bernardino Romera-Paredes, Andreas Argyriou, Nadia Berthouze, and Massimiliano Pontil. Exploiting unrelated tasks in multi-task learning. In *International Conference on Artificial Intelligence and Statistics*, pages 951–959, 2012.

[140] Gail L Rosen, Erin R Reichenberger, and Aaron M Rosenfeld. Nbc: the naive bayes classification tool webserver for taxonomic classification of metagenomic reads. *Bioinformatics*, 27(1):127–129, 2011.

[141] Ramon Rosselló-Mora and Rudolf Amann. The species concept for prokaryotes. *FEMS microbiology reviews*, 25(1):39–67, 2001.

[142] Jonathan M Rothberg, Wolfgang Hinz, Todd M Rearick, Jonathan Schultz, William Mileski, Mel Davey, John H Leamon, Kim Johnson, Mark J Milgrew, Matthew Edwards, et al. An integrated semiconductor device enabling non-optical genome sequencing. *Nature*, 475(7356):348–352, 2011.

[143] DE Rumelhart, GE Hinton, and RJ Williams. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986, 1986.

[144] Sascha Sauer and Magdalena Kliem. Mass spectrometry tools for the classification and identification of bacteria. *Nature Reviews Microbiology*, 8(1):74–82, 2010.

[145] Robert Schmieder and Robert Edwards. Insights into antibiotic resistance through metagenomic approaches. *Future microbiology*, 7(1):73–89, 2012.

[146] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[147] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for l 1-regularized loss minimization. *The Journal of Machine Learning Research*, 12:1865–1892, 2011.

[148] Beletshachew Shiferaw, Sue Shallow, Ruthanne Marcus, Suzanne Segler, Dana Soderlund, Felicia P Hardnett, Thomas Van Gilder, et al. Trends in population-based active surveillance for shigellosis and demographic variability in foodnet sites, 1996–1999. *Clinical infectious diseases*, 38(Supplement 3):S175–S180, 2004.

[149] Naum Z Shor. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25(6):1–11, 1987.

[150] Charles G Sibley and Jon E Ahlquist. The phylogeny of the hominoid primates, as indicated by dna-dna hybridization. *Journal of molecular evolution*, 20(1):2–15, 1984.

[151] N Smirnov. Sur les écarts de la courbe de distribution empirique (russian, french summary). *Matematicheskii Sbornik*, 48(1):3–26, 1939.

[152] Le Song, Alex Smola, Arthur Gretton, and Karsten M Borgwardt. A dependence maximization view of clustering. In *Proceedings of the 24th international conference on Machine learning*, pages 815–822. ACM, 2007.

[153] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006.

[154] Wendy Weijia Soon, Manoj Hariharan, and Michael P Snyder. High-throughput sequencing for biology and medicine. *Molecular systems biology*, 9(1), 2013.

[155] O David Sparkman. Mass spec desk reference. *Global View Publishing, Pittsburgh, Pennsylvania*, page 25, 2006.

[156] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pages 1329–1336, 2004.

[157] EaBMG Stackebrandt and BM Goebel. Taxonomic note: a place for dna-dna reassociation and 16s rrna sequence analysis in the present species definition in bacteriology. *International Journal of Systematic Bacteriology*, 44(4):846–849, 1994.

[158] Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 521–528. IEEE, 2001.

[159] Dipika Sur, T Ramamurthy, Jacqueline Deen, and SK Bhattacharya. Shigellosis: challenges & management issues. *The Indian journal of medical research*, 120(5):454–462, 2004.

[160] Scott Sutton. Qualification of a microbial identification system. *Journal of Validation Technology, 17 (4): 46*, 49, 2011.

[161] Neal J Sweeney, Per Klemm, Beth A McCormick, Eva Moller-Nielsen, Maryjane Utley, Mark A Schembri, David C Laux, and Paul S Cohen. The escherichia coli k-12 gntp gene allows e. coli f-18 to occupy a distinct nutritional niche in the streptomycin-treated mouse large intestine. *Infection and immunity*, 64(9):3497–3503, 1996.

[162] KE Tan, BC Ellis, R Lee, PD Stamper, SX Zhang, and KC Carroll. Prospective evaluation of a matrix-assisted laser desorption ionization–time of flight mass

spectrometry system in a hospital clinical microbiology laboratory for identification of bacteria and yeasts: a bench-by-bench study for assessing the impact on time to identification and cost-effectiveness. *Journal of clinical microbiology*, 50(10):3301–3308, 2012.

[163] Vedat Taşkın, Berat Doğan, and Tamer Ölmez. Prostate cancer classification from mass spectrometry data by using wavelet analysis and kernel partial least squares algorithm. *International Journal of Bioscience, Biochemistry and Bioinformatics*, 3(2), 2013.

[164] Sebastian Thrun and Lorien Pratt. Learning to learn. In *Learning to learn*. Springer, 1998.

[165] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[166] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. In *Journal of Machine Learning Research*, pages 1453–1484, 2005.

[167] Alex van Belkum, Martin Welker, Marcel Erhard, and Sonia Chatellier. Biomedical mass spectrometry in today's and tomorrow's clinical microbiology laboratories. *Journal of clinical microbiology*, 50(5):1513–1517, 2012.

[168] BH Van Herendael, P Bruynseels, M Bensaid, T Boekhout, T De Baere, I Surmont, and AH Mertens. Validation of a modified algorithm for the identification of yeast isolates using matrix-assisted laser desorption/ionisation time-of-flight mass spectrometry (maldi-tof ms). *European journal of clinical microbiology & infectious diseases*, 31(5):841–848, 2012.

[169] Vladimir N Vapnik. An overview of statistical learning theory. *Neural Networks, IEEE Transactions on*, 10(5):988–999, 1999.

[170] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 2. Wiley New York, 1998.

[171] Kevin Vervier, Pierre Mahé, Alexandre D'Aspremont, Jean-Baptiste Veyrieras, and Jean-Philippe Vert. On learning matrices with orthogonal columns or disjoint supports. In *Machine Learning and Knowledge Discovery in Databases*, volume 8726 of *Lecture Notes in Computer Science*, pages 274–289. Springer Berlin Heidelberg, 2014.

[172] Kévin Vervier, Pierre Mahé, Jean-Baptiste Veyrieras, and Jean-Philippe Vert. Benchmark of structured machine learning methods for microbial identification from mass-spectrometry data. *submitted to Plos One*, 2015.

[173] Gerrit J Viljoen, Louis H Nel, and John R Crowther. *Molecular diagnostic PCR handbook*. Springer, 2005.

[174] Qiong Wang, George M Garrity, James M Tiedje, and James R Cole. Naive bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and environmental microbiology*, 73(16):5261–5267, 2007.

[175] Jan Weile and Cornelius Knabbe. Current applications and future trends of molecular diagnostics in clinical bacteriology. *Analytical and bioanalytical chemistry*, 394(3):731–742, 2009.

[176] Jason Weston, Chris Watkins, et al. Support vector machines for multi-class pattern recognition. In *ESANN*, volume 99, pages 219–224, 1999.

[177] Jason G Whalen, Thaddeus W Mully, and Joseph C English. Spontaneous citrobacter freundii infection in an immunocompetent patient. *Archives of dermatology*, 143(1):115–126, 2007.

[178] David L Wheeler, Tanya Barrett, Dennis A Benson, Stephen H Bryant, Kathi Canese, Vyacheslav Chetvernin, Deanna M Church, Michael DiCuccio, Ron Edgar, Scott Federhen, et al. Database resources of the national center for biotechnology information. *Nucleic acids research*, 35(suppl 1):D5–D12, 2007.

[179] JM Willey. *Prescott, Harley, and Klein's Microbiology-7th international ed./Joanne M. Willey, Linda M. Sherwood, Christopher J. Woolverton*. New York [etc.]: McGraw-Hill Higher Education, 2008.

[180] Yulanda M Williamson, Hercules Moura, Adrian R Woolfitt, James L Pirkle, John R Barr, Maria Da Gloria Carvalho, Edwin P Ades, George M Carlone, and Jacquelyn S Sampson. Differentiation of streptococcus pneumoniae conjunctivitis outbreak isolates by matrix-assisted laser desorption ionization-time of flight mass spectrometry. *Applied and environmental microbiology*, 74(19):5891–5897, 2008.

[181] Carl R Woese and George E Fox. Phylogenetic structure of the prokaryotic domain: the primary kingdoms. *Proceedings of the National Academy of Sciences*, 74(11):5088–5090, 1977.

[182] J Wolcott, A Schwartz, C Goodman, and Lewin Group. Laboratory medicine: a national status report. may 2008, 2013.

[183] Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. In *Advances in Neural Information Processing Systems*, pages 2116–2124, 2009.

[184] Lin Xiao, Dengyong Zhou, and Mingrui Wu. Hierarchical classification via orthogonal transfer. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 801–808, 2011.

[185] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[186] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[187] Guanghong Zuo, Zhao Xu, and Bailin Hao. Shigella strains are not clones of escherichia coli but sister species in the genus escherichia. *Genomics, proteomics & bioinformatics*, 11(1):61–65, 2013.

Méthodes d'apprentissage structuré pour la microbiologie: spectrométrie de masse et séquençage haut-débit.

RÉSUMÉ: L'utilisation des technologies haut débit est en train de changer aussi bien les pratiques que le paysage scientifique en microbiologie. D'une part la spectrométrie de masse a d'ores et déjà fait son entrée avec succès dans les laboratoires de microbiologie clinique. D'autre part, l'avancée spectaculaire des technologies de séquençage au cours des dix dernières années permet désormais à moindre coût et dans un temps raisonnable de caractériser la diversité microbienne au sein d'échantillons cliniques complexes. Aussi ces deux technologies sont pressenties comme les piliers de futures solutions de diagnostic.

L'objectif de cette thèse est de développer des méthodes d'apprentissage statistique innovantes et versatiles pour exploiter les données fournies par ces technologies haut-débit dans le domaine du diagnostic *in vitro* en microbiologie. Le domaine de l'apprentissage statistique fait partie intégrante des problématiques mentionnées ci-dessus, au travers notamment des questions de classification d'un spectre de masse ou d'un "read" de séquençage haut-débit dans une taxonomie bactérienne.

Sur le plan méthodologique, ces données nécessitent des développements spécifiques afin de tirer au mieux avantage de leur structuration inhérente: une structuration en "entrée" lorsque l'on réalise une prédiction à partir d'un "read" de séquençage caractérisé par sa composition en nucléotides, et un structuration en "sortie" lorsque l'on veut associer un spectre de masse ou d'un "read" de séquençage à une structure hiérarchique de taxonomie bactérienne.

Mots-clés: Apprentissage statistique, Diagnostic in vitro, Microbiologie, Spectrométrie de masse, Séquençage haut-débit.

Structured machine learning methods for microbiology: mass spectrometry and high-throughput sequencing.

ABSTRACT: Using high-throughput technologies is changing scientific practices and landscape in microbiology. On one hand, mass spectrometry is already used in clinical microbiology laboratories. On the other hand, the last ten years dramatic progress in sequencing technologies allows cheap and fast characterization of microbial diversity in complex clinical samples. Consequently, the two technologies are approached in future diagnostics solutions. This thesis aims to play a part in new *in vitro* diagnostics (IVD) systems based on high-throughput technologies, like mass spectrometry or next generation sequencing, and their applications in microbiology.

Because of the volume of data generated by these new technologies and the complexity of measured parameters, we develop innovative and versatile statistical learning methods for applications in IVD and microbiology. Statistical learning field is well-suited for tasks relying on high-dimensional raw data that can hardly be used by medical experts, like mass-spectrum classification or affecting a sequencing read to the right organism.

Here, we propose to use additional known structures in order to improve quality of the answer. For instance, we convert a sequencing read (raw data) into a vector in a nucleotide composition space and use it as a *structured input* for machine learning approaches. We also add prior information related to the hierarchical structure that organizes the reachable micro-organisms (*structured output*).

Keywords: Machine learning, in vitro diagnostics, Microbiology, Mass spectrometry, High-throughput sequencing.

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

MINES ParisTech