



**HAL**  
open science

# Contributions to the estimation of probabilistic discriminative models: semi-supervised learning and feature selection

Nataliya Sokolovska

► **To cite this version:**

Nataliya Sokolovska. Contributions to the estimation of probabilistic discriminative models: semi-supervised learning and feature selection. Mathématiques [math]. Télécom ParisTech, 2010. Français. NNT : . pastel-00006257

**HAL Id: pastel-00006257**

**<https://pastel.hal.science/pastel-00006257>**

Submitted on 27 Jul 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



École Doctorale  
d'Informatique,  
Télécommunications  
et Électronique de Paris

Thèse de doctorat de  
Télécom ParisTech  
Spécialité : Informatique et réseaux  
présentée par

**Nataliya Sokolovska**

---

**Contributions à l'estimation de modèles  
probabilistes discriminants: apprentissage  
semi-supervisé et sélection de caractéristiques**

---

Soutenue le 25 février 2010 devant le jury composé de :

Thierry Artières	Président
Francis Bach	Examineur
Yves Grandvalet	Rapporteur
Marc Tommasi	Rapporteur
Olivier Cappé	Directeur de thèse
François Yvon	Directeur de thèse



---

# CONTENTS

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Contributions . . . . .	2
1.3	Structure of the Thesis . . . . .	3
<b>I</b>	<b>Learning in Discriminative and Generative Models</b>	<b>5</b>
<b>2</b>	<b>Discriminative and Generative Learning in Probabilistic Graphical Models</b>	<b>7</b>
2.1	Discriminative and Generative Learning . . . . .	8
2.1.1	Generative models . . . . .	8
2.1.2	Discriminative models . . . . .	9
2.2	Logistic Regression and Naive Bayes for Classification Tasks . . . . .	9
2.2.1	Logistic Regression . . . . .	9
2.2.2	Naive Bayes Classifier . . . . .	12
2.3	Graphical Models . . . . .	12
2.3.1	Directed Graphs . . . . .	14
2.3.2	Undirected Models . . . . .	15
2.4	Hidden Markov Models and Maximum Entropy Markov Models . . . . .	16
2.4.1	Hidden Markov Models . . . . .	16
2.4.2	Maximum Entropy Markov Models . . . . .	19
2.5	Conclusions . . . . .	21
<b>3</b>	<b>Conditional Random Fields</b>	<b>23</b>
3.1	Model Description . . . . .	24
3.2	Training and Decoding in Conditional Random Fields . . . . .	25
3.2.1	Training in CRFs . . . . .	25

3.2.2	Decoding . . . . .	26
3.3	Optimization Methods for Conditional Random Fields . . . . .	28
3.3.1	Conjugate Gradient . . . . .	28
3.3.2	BFGS and L-BFGS . . . . .	29
3.3.3	Stochastic Gradient Descent . . . . .	31
3.4	Applications and generalizations of CRFs . . . . .	32
3.4.1	Application Domains of CRFs . . . . .	32
3.4.2	Generalizations and Alternative Estimation Methods . . . . .	34
3.5	Performance of Conditional Random Fields . . . . .	37
3.5.1	Performance of Conditional Random Fields on Nettealk Corpus . . . . .	38
3.5.2	Performance of Conditional Random Fields on CoNLL Data Sets . . . . .	41
3.6	Conclusions . . . . .	45
<b>II</b>	<b>Semi-Supervised Discriminative Estimator</b>	<b>47</b>
<b>4</b>	<b>Semi-Supervised Learning of Discriminative Models</b>	<b>49</b>
4.1	Brief Overview of Semi-Supervised Learning Methods . . . . .	50
4.1.1	Four Assumptions Proposed for Semi-Supervised Learning . . . . .	52
4.1.2	Categories of Algorithms . . . . .	53
4.1.3	Semi-Supervised CRFs . . . . .	57
4.2	Marginal Probability in Discriminative Models . . . . .	59
4.2.1	Asymptotically Optimal Semi-Supervised Estimation . . . . .	60
4.2.2	Covariate Shift . . . . .	65
4.3	Application to Binary Logistic Regression . . . . .	67
4.4	Conclusions . . . . .	68
<b>5</b>	<b>Semi-Supervised Learning Experiments</b>	<b>69</b>
5.1	A Small Scale Experiment . . . . .	69
5.1.1	Multinomial Model and Artificial Data . . . . .	70
5.1.2	Experiments with the Criterion of Bengio-Grandvalet . . . . .	71
5.1.3	Performance of the Proposed Semi-Supervised Estimator . . . . .	71
5.2	Text Classification Experiments . . . . .	74
5.3	Conclusions . . . . .	75

<b>III</b>	<b><math>L_1</math> Norm Based Model Selection in Discriminative Models</b>	<b>77</b>
<b>6</b>	<b>Sparsity and Model Selection in Discriminative Models</b>	<b>79</b>
6.1	Empirical Study: Sparsity in Conditional Random Fields . . . . .	80
6.1.1	How Many Features Can Be Eliminated? . . . . .	80
6.1.2	Most Influential Features . . . . .	83
6.2	Brief Overview of Feature Selection Techniques . . . . .	85
6.2.1	Naive Model Selection Methods for CRFs . . . . .	85
6.2.2	Heuristic Approaches Applied to CRFs . . . . .	86
6.2.3	Penalty Terms Including the $L_1$ Norm . . . . .	86
6.3	Numerical Optimization of Criteria Including the $L_1$ Norm . . . . .	88
6.3.1	Orthant-Wise Limited-Memory Quasi-Newton . . . . .	89
6.3.2	Coordinate-Wise Descent . . . . .	90
6.4	Conclusions . . . . .	92
<b>7</b>	<b>Application of Coordinate-wise Optimization Approach to CRFs</b>	<b>93</b>
7.1	Coordinate-wise Method for Conditional Random Fields . . . . .	94
7.1.1	Coordinate Descent and Discussion on the Approximation of the Second Derivatives . . . . .	94
7.1.2	Blockwise Coordinate Descent for CRFs . . . . .	96
7.2	Implications of Sparsity: Sparse Forward-Backward . . . . .	97
7.3	Experiments with Elastic Net Conditional Random Fields . . . . .	98
7.3.1	Artificial Data Set . . . . .	98
7.3.2	Nettalk Corpus (Phonetisation Task) . . . . .	103
7.3.3	CoNLL 2000 and CoNLL 2003 Data Sets . . . . .	105
7.4	Conclusions . . . . .	112
<b>8</b>	<b>Conclusions and Future Directions</b>	<b>113</b>
<b>A</b>	<b>Asymptotic Performance of the Semi-Supervised Estimator for <math>K</math>-classes Logistic Regression</b>	<b>115</b>
<b>B</b>	<b>Nettalk Corpus</b>	<b>117</b>
<b>C</b>	<b>CoNLL 2000 and CoNLL 2003 Data Sets</b>	<b>119</b>
<b>D</b>	<b>Expression of the Full Hessian for the Block of Parameters Associated with <math>\mu_{y,x}</math> and <math>\lambda_{y',y,x}</math></b>	<b>123</b>



---

# LIST OF FIGURES

---

1.1	Example of ambiguity in sentence parsing: two possible parsings of a same sentence. . . . .	2
2.1	Graphical representations of the generative model naive Bayes (left) and discriminative model logistic regression (right). . . . .	13
2.2	Example of a (rather complex) directed graphical model . . . . .	14
2.3	Example of a directed graphical model representing certain conditional independence assumptions . . . . .	14
2.4	Example of an undirected graphical model, the nodes $x_4$ and $x_5$ are independent from $x_1$ given $x_2$ , and $x_3$ respectively. . . . .	15
2.5	Graphical representation of hidden Markov models (left) and maximum entropy Markov models (right). . . . .	16
2.6	Tables. Partial results of $\alpha$ -pass calculations (left) and backtracking procedure (right). . . . .	19
3.1	Graphical representation of linear-chain conditional random fields for a sequence of length 3. . . . .	24
3.2	A clique modeling the dependency $(y'', y', y, x)$ . . . . .	35
3.3	Example of node-splitting of the clique represented on Figure 3.2. . . . .	35
3.4	Graphical scheme of dynamic conditional random fields (left) and hidden conditional random fields (right). . . . .	36
4.1	On the left: generative framework; on the right: discriminative framework. . . . .	54
5.1	Boxplots of the scaled squared parameter estimation error as a function of the number of observations. Left: for logistic regression, $n\ \hat{\theta}_n - \theta_\star\ ^2$ ; right: for the semi-supervised estimator, $n\ \hat{\theta}_n^s - \theta_\star\ ^2$ . . . . .	72
5.2	Boxplots of the scaled excess logarithmic risk as a function of the number of observations. Left: for logistic regression, $n(\mathbb{E}_\pi[\ell(Y X; \hat{\theta}_n)] - \mathbb{E}_\pi[\ell(Y X; \theta_\star)])$ ; right: for the semi-supervised estimator, $n(\mathbb{E}_\pi[\ell(Y X; \hat{\theta}_n^s)] - \mathbb{E}_\pi[\ell(Y X; \theta_\star)])$ . . . . .	72



5.3	Boxplots of the scaled squared parameter estimation error as a function of the number of observations for the case with the covariate shift. Left: for the logistic regression, $n\ \hat{\theta}_n - \theta_\star\ ^2$ ; right: for the semi-supervised estimator, $n\ \hat{\theta}_n^s - \theta_\star\ ^2$ . . . . .	73
5.4	Boxplots of the scaled excess logarithmic risk as a function of the number of observations for the case with a covariate shift. Left: for the logistic regression, $n(\mathbb{E}_\pi[\ell(Y X; \hat{\theta}_n)] - \mathbb{E}_\pi[\ell(Y X; \theta_\star)])$ ; right: for the semi-supervised estimator, $n(\mathbb{E}_\pi[\ell(Y X; \hat{\theta}_n^s)] - \mathbb{E}_\pi[\ell(Y X; \theta_\star)])$ . . . . .	73
5.5	Boxplots of the probability of error as a function of the number of observations for a well-specified model. Left: for the logistic regression; right: for the semi-supervised estimator. . . . .	74
5.6	Boxplots of the error rates for, L50: logistic regression with $n = 50$ ; S50: semi-supervised estimator with $n = 50$ ; L300 and S300, idem with $n = 300$ . . . . .	75
6.1	$L_1$ norm of the parameters estimated with standard $L_2$ -regularized maximum likelihood for the Nettetalk task. Left: $ \mu_{y,x} $ for the 53 phonemes $y$ and 26 letters $x$ . Right: $\sum_{y'}  \lambda_{y',y,x} $ for the 53 phonemes $y$ and 26 letters $x$ . . . . .	81
6.2	Nettalk. Number of active features $(y', y, x)$ for every possible $(y, x)$ dependency. Left: features from the interval $(-\infty, -0.25], [0.25, +\infty)$ . Right: features from the interval $(-\infty, -2], [2, +\infty)$ . . . . .	84
6.3	CoNLL 2000. Left: The number of dependencies based on words which are active with performance being close to the baseline. Right: the number of dependencies based on POS tags that are active with performance being close to the baseline. . . . .	84
7.1	Distribution generating synthetic data. Left: distribution $p(y_t y_{t-1})$ , right: $p(x_t y_t)$ . . . . .	99
7.2	Performance of the models on artificial data. Models $M1 - M3$ are trained with $L_2$ penalty (L-BFGS), models $M4 - M8$ with the $L_1$ penalty term (block coordinate-wise descent). Left: performance on training set. Right: performance on testing set. . . . .	100
7.3	Average selection frequency of unigram $\mu$ features (on the left) and their estimated average values (on the right). . . . .	100
7.4	Average selection frequency of bigram $\lambda$ features for $X = 1, \dots, 5$ . . . . .	101
7.5	Performance comparison of coordinate-wise method, block-wise method, and coordinate-wise method that does not revisit points which have been zeroed in a previous iteration . . . . .	102
7.6	Logarithmic loss comparison of coordinate-wise method, block-wise method, and coordinate-wise method that does not revisit points which have been zeroed in a previous iteration . . . . .	102
7.7	Nettalk experiments, $\rho_1 = 0.2$ , $\rho_2 = 0.05$ . Left: feature values of type unigram. Right: feature values of type bigram: $\sum_{y_{t-1}}  \lambda_{y_{t-1}, y_t, x_t} $ . . . . .	103

7.8	Running time as a function of the number of active features for the SBCD algorithm on the Nettealk corpus. The blue line corresponds to the running time when using non-sparse forward-backward. . . . .	105
7.9	CoNLL 2003, $\rho_2 = 0.001$ . . . . .	106
7.10	CoNLL 2000, error. . . . .	106
7.11	CoNLL 2003. Left: number of active parameters that depend on POS tags and syntactic chunks. Right: number of active parameters that depend on words. . . . .	107
7.12	CoNLL 2000 ( $\rho_1 = 0.5$ , $\rho_2 = 1e - 05$ ). Left: values of unigram parameters that depend on words for which $\sum_y  \mu_{y,x^1}  > 5$ . Right: values of unigram parameters that depend on POS tags. . . . .	108
7.13	CoNLL 2000. Left: Unnormalized frequency of POS tags. Right: Unnormalized frequency of chunks. . . . .	108
7.14	CoNLL 2000 ( $\rho_1 = 0.5$ , $\rho_2 = 1e - 05$ ). Left: values of bigram parameters ( $\sum_{y_{t-1}}  \lambda_{y_{t-1},y_t,x_t^1} $ ) that depend on words for which $\sum_y  \mu_{y,x^1}  > 5$ . Right: values of bigram parameters ( $(\sum_{y_{t-1}}  \lambda_{y_{t-1},y_t,x_t^2} )$ ) that depend on POS tags. . . . .	109
7.15	CoNLL 2000. Left: positive unigram POS parameters. Right: negative unigram POS parameters. . . . .	109
7.16	CoNLL 2000. Left: positive $(y_{t-1}, y_t)$ parameters. Right: negative $(y_{t-1}, y_t)$ parameters. . . . .	110
7.17	CoNLL 2003 Data. Dependence of performance on the number of active features. Left: on set Test A, right: on set Test B. . . . .	111
7.18	CoNLL 2003 Data. Dependence of F-measure on the number of active features. Left: on set Test A, right: on set Test B. . . . .	111



---

# LIST OF TABLES

---

3.1	Example of named-entity segmentation . . . . .	34
3.2	Different patterns on Nettealk corpus, estimation carried with CRF++ except for features marked with a star (our Matlab implementation). . . . .	40
3.3	The most frequent confusions (IPA) made by conditional random fields on Nettealk corpus. Left: training carried out with the feature $(y_{t-1}, y_t, x_t)$ . Right: training carried out with the feature combination $(y_{t-1}, y_t, x_t) + (y_t, x_t)$ . . . . .	41
3.4	Words with their correct and predicted labels for that $score(\mathbf{y}_{\text{labeled}}) - score(\mathbf{y}_{\text{correct}}) > 8$ . . . . .	42
3.6	Classification confusion matrix . . . . .	42
3.5	Performance of naive features . . . . .	43
3.7	Performance on CoNLL 2000, CRF++ ( $\sigma^2 = 1$ ) . . . . .	43
3.8	Performance on CoNLL 2003, CRF++ ( $\sigma^2 = 50$ ) . . . . .	44
6.1	Empirical study of sparsity patterns (CoNLL 2000 Corpus, English) . Dependencies $\lambda_{y',y,x^j}, \mu_{y,x^j}, j \in \{1, 2\}$ . . . . .	81
6.2	Empirical study of sparsity patterns (CoNLL 2003 Corpus, English). Dependencies $\lambda_{y',y,x^j}, \mu_{y,x^j}, j \in \{1, 2, 3\}$ . . . . .	82
6.3	Empirical study of sparsity patterns (Nettealk). Dependencies $\lambda_{y',y,x}, \mu_{y,x}$ . . . . .	83
7.1	Impact of $\rho_1$ on the number of active features ( $\rho_2 = 0.001$ ). . . . .	99
7.2	Upper part: summary of results for various values of $\rho_1$ for the proposed Sparse Blockwise Coordinate Descent (SBCD) algorithm (with $\rho_2 = 0.001$ ) and orthant-wise L-BFGS (OWL-QN). Lower part: results obtained with $\rho_2$ regularization only, for L-BFGS and stochastic gradient descent (SGD). . . . .	103
7.3	Results for CoNLL 2000, with $\rho_1 = 0.5, \rho_2 = 1e - 05$ . . . . .	107
7.4	Results for CoNLL 2003, with $\rho_1 = 0.1, \rho_2 = 1e - 05$ . . . . .	107
7.5	CoNLL 2000, words with the maximal absolute unigram values, $\sum_y  \mu_{y,x^1}  > 10$ . . . . .	110
7.6	CoNLL 2003, words with the maximal absolute unigram values, $\sum_y  \mu_{y,x^1}  > 7$ . . . . .	110

C.1	Part of Speech Tags. . . . .	120
C.2	Chunks of CoNLL 2000. . . . .	121
C.3	Corpora CoNLL 2000 and CoNLL 2003 details. . . . .	121

---

# LIST OF ALGORITHMS

---

1	Sequence Generation in a First Order Markov Process . . . . .	17
2	The Viterbi Algorithm . . . . .	19
3	Training CRF . . . . .	27
4	Conjugate Gradient . . . . .	29
5	Quasi-Newton Algorithm . . . . .	31
6	Expectation-Maximization Algorithm . . . . .	54
7	Algorithm of C.T. Ireland and S. Kullback . . . . .	57
8	Coordinate-wise Descent for CRF . . . . .	95
9	Blockwise Coordinate Descent for CRF with Diagonal Hessian Approximation	97



---

# REMERCIEMENTS

---

Je tiens tout d’abord à remercier les membres du jury qui m’ont fait l’honneur de bien vouloir consacrer une partie de leur temps à l’évaluation de ce travail. Je remercie Thierry Artières d’avoir accepté de présider le jury de thèse. Merci à Yves Grandvalet et Marc Tommasi d’avoir conduit la tâche de rapporteur.

Je remercie mes directeurs de thèse, Olivier Cappé et François Yvon pour le sujet intéressant et stimulant de cette thèse et pour leur soutien scientifique. J’ai eu également le plaisir de collaborer avec Thomas Lavergne.

Merci aux organisateurs du séminaire SMILE (Statistical Machine Learning in Paris), surtout à Jean-Philippe Vert et Francis Bach. J’ai appris beaucoup pendant ces séances.

Je voudrais remercier Boris Tkachouk (Professeur à NTUU “KPI”) qui a joué un rôle fondamental dans ma formation.

Cette thèse ne serait pas ce qu’elle est sans tous mes collègues à Télécom ParisTech. Merci pour des moments joyeux et instructifs.

Je remercie également l’ensemble du personnel de l’équipe STA au département TSI à Télécom ParisTech pour m’avoir accueilli pendant les années de ma thèse.

Je tiens à remercier ma famille, mes parents Tetyana et Valentin, mon frère Sergey pour leur soutien constant tout au long de mes études et de mon doctorat. Ma grand-mère Irina, merci pour me toujours donner le bon exemple. Merci à Wolfgang d’être toujours à mes côtés.



---

# ABSTRACT

---

In this thesis, we investigate the use of parametric probabilistic models for classification tasks in the domain of natural language processing. We focus in particular on discriminative models, such as logistic regression and its generalization, conditional random fields (CRFs).

Discriminative probabilistic models design directly conditional probability of a class given an observation. The logistic regression has been widely used due to its simplicity and effectiveness. Conditional random fields allow to take structural dependencies into consideration and therefore are used for structured output prediction. In this study, we address two aspects of modern machine learning, namely, semi-supervised learning and model selection, in the context of CRFs.

The contribution of this thesis is twofold. First, we consider the framework of semi-supervised learning and propose a novel semi-supervised estimator and show that it is preferable to the standard logistic regression. Second, we study model selection approaches for discriminative models, in particular for CRFs and propose to penalize the CRFs with the elastic net. Since the penalty term is not differentiable in zero, we consider coordinate-wise optimization. The comparison with the performances of other methods demonstrates competitiveness of the CRFs penalized by the elastic net.

---

# RÉSUMÉ

---

Dans cette thèse nous étudions l'estimation de modèles probabilistes discriminants, surtout des aspects d'apprentissage semi-supervisé et de sélection de caractéristiques.

Le but de l'apprentissage semi-supervisé est d'améliorer l'efficacité de l'apprentissage supervisé en utilisant des données non-étiquetées. Cet objectif est difficile à atteindre dans les cas des modèles discriminants.

Les modèles probabilistes discriminants permettent de manipuler des représentations linguistiques riches, sous la forme de vecteurs de caractéristiques de très grande taille. Travailler en grande dimension pose des problèmes, en particulier computationnels, qui sont exacerbés dans le cadre de modèles de séquences tels que les champs aléatoires conditionnels (CRF). Sélectionner automatiquement les caractéristiques pertinentes s'avère alors intéressant et donne lieu à des modèles plus compacts et plus faciles à utiliser.

Notre contribution est double. Nous introduisons une méthode originale et simple pour intégrer des données non étiquetées dans une fonction objectif semi-supervisée. Nous démontrons alors que l'estimateur semi-supervisé correspondant est asymptotiquement optimal. Le cas de la régression logistique est illustré par des résultats d'expériences.

Dans cette étude, nous proposons un algorithme d'estimation pour les CRF qui réalise une telle sélection, par le truchement d'une pénalisation  $L_1$ . Nous présentons également les résultats d'expériences menées sur des tâches de traitement des langues (le *chunking* et la détection des entités nommées), en analysant les performances en généralisation et les caractéristiques sélectionnées. Nous proposons finalement diverses pistes pour améliorer l'efficacité computationnelle de cette technique.

## Analyse asymptotique de l'apprentissage semi-supervisé pour les modèles probabilistes discriminants

---

Dans de nombreux problèmes de classification (pour l'image, le son ou le texte), on dispose de masses de données non-étiquetées facilement accessibles, alors que les données étiquetées sont incomparablement moins volumineuses et sont coûteuses à acquérir.

Une question importante est donc de trouver des méthodes qui utilisent des données non-étiquetées pour améliorer les performances de l'apprentissage supervisé. Dans les dernières années, ce problème a suscité le développement de nombreux algorithmes (voir (Chapelle et al., 2006) pour un état de l'art récent).

Nous considérons des méthodes d'apprentissage probabilistes, c'est-à-dire, des méthodes qui associent une mesure de confiance probabiliste à chaque décision: en particulier, le modèle de régression logistique et ses extensions polytomiques nous serviront d'illustration de nos techniques. Ces méthodes ne sont pas nécessairement les meilleures du point de vue, par exemple, des performances en classification, mais elles sont importantes pour des applications qui sont basées sur l'erreur de généralisation, des applications de ranking, des combinaisons de décisions de sources multiples, etc. Dans le contexte de l'apprentissage semi-supervisé, on distingue généralement modèles *génératifs* et modèles *discriminants*. Les modèles probabilistes génératifs peuvent, en effet, s'accommoder de données non-étiquetées d'une manière très intuitive, en spécifiant des modèles à données latentes, qu'il est possible d'estimer par l'algorithme EM (*Expectation-Maximization*) (voir, par exemple (Nigam et al., 2000, Klein and Manning, 2004) pour des mises en pratique réussies de cette idée).

Les modèles discriminants permettent en général d'atteindre de meilleures performances que les modèles génératifs pour des problèmes de classifications (Ng and Jordan, 2002). Malheureusement, l'intégration de données non-étiquetées, est, dans ce cadre, beaucoup moins évidente. La raison en est claire: supposons que l'on veut apprendre à prédire une étiquette  $y$  à partir de l'observation de  $x$ ; l'apprentissage discriminant d'un modèle réalisant cette tâche va typiquement chercher à maximiser  $P(y|x;\theta)$ , où  $\theta$  est un vecteur de paramètres. Dans ce contexte, toute connaissance supplémentaire sur la distribution marginale  $P(x)$  que pourraient apporter des données non-étiquetées semble essentiellement inutile; c'est du moins la thèse défendue par (Seeger, 2002, Lasserre et al., 2006). Une des contributions de notre étude est de prouver que cette intuition s'appuie sur une hypothèse implicite que le modèle est *bien spécifié*, (au sens où l'espace des modèles considérés lorsque le paramètre varie contient le "vrai" modèle); nous aurons l'occasion de montrer que lorsque cette hypothèse n'est pas vérifiée, alors les données non-étiquetées peuvent avoir leur utilité.

Pour sortir de cette impasse, l'approche la plus répandue consiste à faire dépendre le vecteur de paramètres  $\theta$ , soit directement, soit indirectement, des données non-étiquetées. Une manière d'introduire une telle dépendance consiste à poser des contraintes sur la forme de  $P(y|x)$ : "l'hypothèse de cluster" (*cluster assumption*), par exemple, stipule que les frontières de décision se trouvent dans des régions de faible densité de  $P(x)$  (Seeger, 2002, Chapelle and Zien, 2005). (Grandvalet and Bengio, 2004) utilisent cette intuition dans une méthode d'apprentissage semi-supervisé (*régularisation de l'entropie*), qui introduit une nouvelle fonction objectif combinant le terme habituel de log-vraisemblance (conditionnelle) avec une pénalité basée sur l'entropie, qui impose que les paramètres soit positionnés de façon à classer sans ambiguïté les exemples non-étiquetés. Cette idée est appliquée aux Champs Aléatoires Conditionnels (Lafferty et al., 2001) par (Jiao et al., 2006); on se reportera également à (Corduneanu and Jaakkola, 2003) pour des idées similaires.

La proposition de (Grandvalet and Bengio, 2004), comme presque toutes les approches qui introduisent des termes supplémentaires dans la fonction objectif du modèle d'apprentissage supervisé, se heurte aux problèmes suivants: (i) la convexité de la fonction objective n'est plus garantie, ce qui rend le problème d'optimisation plus difficile et très sensible aux conditions initiales; (ii) la consistance asymptotique de l'estimateur usuel (maximisant la vraisemblance conditionnelle) est également perdue: concrètement, cela signifie qu'il est possible de construire des configurations dans lesquelles l'utilisation des données non-étiquetées conduit en fait à dégrader les performances de l'estimateur

usuel. Pour résumer, obtenir des résultats positifs avec ces techniques demande de régler finement les différents paramètres contrôlant le comportement de l’optimisation.

L’“hypothèse de cluster” est également au fondement des méthodes à base de graphes, qui utilisent l’intuition que les points non-étiquetés doivent recevoir les mêmes étiquettes que leur(s) (proche(s)) voisin(s) étiqueté(s): dans (Zhu and Ghahramani, 2002), cette idée est réalisée par un algorithme propageant de manière itérative des étiquettes dans un graphe de voisinage: initialement, seuls sont connus les labels des données étiquetées, labels qui sont propagés dans le graphe vers les points non-étiquetés jusqu’à convergence.

(Lasserre et al., 2006) propose une autre approche, plus directe, pour faire dépendre le modèle de classification des données non-étiquetées; elle consiste à considérer deux ensembles de paramètres: un pour la probabilité conditionnelle  $P(y|x;\theta)$ , et l’autre pour la probabilité marginale  $P(x;\nu)$ . Le cas où  $\theta$  et  $\nu$  sont indépendants est celui d’un modèle discriminant “pur” dans lequel on ne peut pas tirer partie des données non-étiquetées. Le cas  $\theta = \nu$  correspond à un modèle génératif traditionnel ( $P(x) = \sum_y P(x|y;\theta)$ ); l’introduction (via une distribution *a priori*) de dépendances entre  $\theta$  et  $\nu$  permet de construire toute une gamme de situations intermédiaires, correspondant à modèles hybrides. Mentionnons finalement (Mann and McCallum, 2007b), qui étudie toutefois une situation assez différente de la nôtre, dans laquelle on dispose d’une connaissance *a priori* de la distribution marginale des labels  $Y$ ; cette méthode semble donner des résultats prometteurs.

Dans cette étude, nous essayons de remettre en cause le point de vue selon lequel les données non-étiquetées seraient inutiles dans des modèles discriminants. à cet effet, nous introduisons un nouvel estimateur, dénommé l’estimateur “semi-supervisé”, du paramètre  $\theta$  dont nous montrons qu’il est asymptotiquement optimal et qu’il est, dans certains cas, préférable à l’estimateur usuel (maximisant la vraisemblance conditionnelle). Pour cela, nous nous plaçons dans une situation idéale dans laquelle la probabilité marginale est complètement connue; cette supposition est vraie à la limite où l’on dispose d’un nombre infini de données non-étiquetées. Dans ce cadre, une observation intéressante est que la méthode proposée est la plus efficace quand l’erreur de Bayes est très faible. Cette observation correspond très bien avec l’intuition précédemment évoquée selon laquelle les algorithmes semi-supervisés sont le plus efficaces lorsque les classes sont bien séparées. Pour compléter les résultats asymptotiques, nous discutons également les résultats empiriques obtenus en utilisant l’estimateur semi-supervisé dans un modèle de régression logistique (classification binaire).

## ESTIMATEUR SEMI-SUPERVISÉ

Soit  $g(y|x;\theta)$  la fonction de densité de probabilité conditionnelle correspondant à un modèle probabiliste discriminant paramétré par  $\theta \in \Theta$ . Dans la suite, nous supposons que la variable de classe  $Y$  prend ses valeurs dans un ensemble fini  $\mathcal{Y}$ ; un cas particulier que nous développons plus longuement est celui où les labels de classes sont binaires  $\mathcal{Y} = \{0, 1\}$ . Nous supposons également que les observations  $X$  appartiennent à un ensemble fini  $\mathcal{X}$ , qui peut être arbitrairement grand. La procédure d’apprentissage a accès à un ensemble de  $n$  observations i.i.d. étiquetées,  $(X_i, Y_i)_{1 \leq i \leq n}$  ainsi qu’à des observations non-étiquetées. Le nombre d’observations non-étiquetées peut être infini, nous supposons qu’il est suffisamment grand pour que la probabilité marginale des observations soit complètement

connue.

Pour une fonction  $f : \mathbb{R}^p \mapsto \mathbb{R}$ ,  $\nabla_z f(z_*)$  dénote un vecteur de gradient de dimension  $p \times 1$  et  $\nabla_{z^T} \nabla_z f(z_*)$  une matrice hessienne  $p \times p$  au point  $z_*$ . Si  $f : \mathbb{R}^p \mapsto \mathbb{R}^r$ , on note  $\nabla_{z^T} f(z_*)$  une matrice jacobienne  $r \times p$  au point  $z_*$ . Enfin,  $E_q(f)$  et  $V_q(f)$  désignent respectivement l'espérance et la variance de  $f$  sous la loi  $q$ .

## Un cas simple

Pour débiter, nous considérons le cas d'un modèle très simple, que nous désignerons par la suite sous le terme de "modèle complètement spécifié". Soit  $\pi(x, y)$  la probabilité jointe complète de  $X$  et  $Y$  estimée par le modèle. Soient  $\eta(y|x)$  et  $q(x)$  respectivement la probabilité conditionnelle et la probabilité marginale associées à  $\pi$ . Bien que ce cas ne soit pas très intéressant pour les applications réelles de l'apprentissage statistique, il fournit un cadre simple pour étudier le rôle qui joue la loi marginale  $q$  dans l'apprentissage semi-supervisé.

Pour ce modèle, il est bien connu que l'estimateur du maximum de vraisemblance de  $\pi(x, y)$  défini par

$$\hat{\pi}_n(x, y) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{X_i = x, Y_i = y\} \quad (1)$$

est asymptotiquement efficace avec une variance asymptotique  $v(x, y) = \pi(x, y)(1 - \pi(x, y))$  (on suppose que  $0 < \pi(x, y) < 1$ ).

On suppose maintenant que  $q(x)$ , la probabilité marginale de  $X$  est connue, avec  $0 < q(x) < 1$ . Il est facile de vérifier que l'estimateur du maximum de vraisemblance de  $\pi(x, y)$  sous contrainte marginale  $\sum_{y \in \mathcal{Y}} \pi(x, y) = q(x)$  est défini par:

$$\hat{\pi}_n^s(x, y) = \frac{\sum_{i=1}^n \mathbb{1}\{X_i = x, Y_i = y\}}{\sum_{i=1}^n \mathbb{1}\{X_i = x\}} q(x) \quad (2)$$

où l'indice  $s$  indique que l'estimateur est "semi-supervisé". La fraction (2) correspond à l'estimateur de maximum de vraisemblance de la probabilité conditionnelle  $\eta(y|x)$ .

$\hat{\pi}_n^s(x, y)$  étant un rapport de deux estimateurs simples, il est possible de calculer sa variance asymptotique avec la méthode  $\delta$ :

$$v^s(x, y) = \pi(x, y)(1 - \pi(x, y)/q(x))$$

Comme  $0 < \pi(x, y) \leq q(x) < 1$ ,  $v^s(x, y)$  est plus petit que  $v(x, y)$ .

Ce résultat élémentaire montre qu'en général, les estimateurs semi-supervisés  $\hat{\pi}_n^s(x, y)$  et  $\hat{\pi}_n(x, y)$  ne sont pas équivalents asymptotiquement, et que  $\hat{\pi}_n^s(x, y)$ , qui a une plus petite variance asymptotique, est préférable. Plus précisément,  $v^s(x, y)/v(x, y) = (1 - \pi(x, y)/q(x))/(1 - \pi(x, y))$ , qui tend vers zéro quand  $q(x)$  s'approche de  $\pi(x, y)$ . Autrement dit, la performance de  $\hat{\pi}_n^s(x, y)$  est d'autant meilleure que celle de  $\hat{\pi}_n(x, y)$  que  $y$  est une étiquette rare pour  $x$ . Dans ce cas,  $\hat{\pi}_n^s(x, y)$ , qui tire profit de l'observation de  $x$  avec d'autres labels que  $y$ , est préférable à  $\hat{\pi}_n(x, y)$ , qui n'utilise pas la connaissance de la distribution marginale  $q(x)$ , et ne peut donc bénéficier de ces informations supplémentaires.

## Modèle discriminant général

Les résultats énoncés dans la section précédente peuvent en fait être énoncés dans un cadre beaucoup plus général, dans lequel la loi conditionnelle est paramétrée par un vecteur  $\theta \in \Theta$ . La principale différence entre les deux situations est qu'un modèle paramétrique  $\{g(y|x; \theta)\}_{\theta \in \Theta}$  n'est pas nécessairement capable d'approcher exactement la distribution conditionnelle  $\eta(y|x)$  des données. Comme dans le cas complètement spécifié, il est pourtant possible de construire un estimateur semi-supervisé qui est asymptotiquement optimal et dont nous allons prouver qu'il est préférable à l'estimateur usuel du maximum de vraisemblance défini par:

$$\hat{\theta}_n = \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n \ell(Y_i|X_i; \theta) \quad (3)$$

où  $\ell(y|x; \theta) = -\log g(y|x; \theta)$  est l'opposé de la fonction de log-vraisemblance conditionnelle.

Sous les hypothèses (classiques) du théorème 4.1 (voir ci-dessous),  $\frac{1}{n} \sum_{i=1}^n \ell(Y_i|X_i; \theta)$  tend uniformément en  $\theta$  vers  $E_\pi[\ell(Y|X; \theta)]$  et la valeur limite de  $\hat{\theta}_n$  est

$$\theta_\star = \arg \min_{\theta \in \Theta} E_\pi[\ell(Y|X; \theta)] \quad (4)$$

L'estimateur du maximum de vraisemblance dans (3) peut être interprété comme  $\hat{\theta}_n = \arg \min_{\theta \in \Theta} E_{\hat{\pi}_n}[\ell(Y|X; \theta)]$  où

$$\hat{\pi}_n(x, y) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{X_i = x, Y_i = y\}$$

dénote la mesure empirique associée avec un couple  $(X_i, Y_i)_{1 \leq i \leq n}$ , qui coïncide également avec l'estimation du maximum de vraisemblance de  $\pi(x, y)$  définie dans (1).

Si l'on considère maintenant que la loi marginale  $q(x)$  est connue,  $\hat{\pi}_n(x, y)$  est dominé (asymptotiquement) par l'estimateur  $\hat{\pi}_n^s(x, y)$ , défini dans (2), récrit ici:

$$\hat{\pi}_n^s(x, y) = \begin{cases} \frac{\sum_{i=1}^n \mathbb{1}\{X_i=x, Y_i=y\}}{\sum_{i=1}^n \mathbb{1}\{X_i=x\}} q(x) & \text{si } \sum_{i=1}^n \mathbb{1}\{X_i = x\} > 0 \\ 0 & \text{sinon} \end{cases} \quad (5)$$

Par analogie avec l'estimateur construit précédemment, on introduit l'estimateur semi-supervisé de la façon suivante:  $\hat{\theta}_n^s = \arg \min_{\theta \in \Theta} E_{\hat{\pi}_n^s}[\ell(Y|X; \theta)]$ , où la notation  $E_{\hat{\pi}_n^s}[f(Y, x)] = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \hat{\pi}_n^s(x, y) f(x, y)$  est utilisée ici de manière un peu abusive, puisque pour  $n$  fini, il est possible que l'on ait  $\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \hat{\pi}_n(x, y) < 1$ , bien que  $\hat{\pi}_n(x, y)$  somme à un avec probabilité un pour  $n$  assez grand.

Il est facile de vérifier que l'on peut récrire  $\hat{\theta}_n^s$  comme:

$$\hat{\theta}_n^s = \arg \min_{\theta \in \Theta} \sum_{i=1}^n \frac{q(X_i)}{\sum_{j=1}^n \mathbb{1}\{X_j = X_i\}} \ell(Y_i|X_i; \theta) \quad (6)$$

L'équation (6) est donc une version pondérée de (3), où la pondération de  $x$  reflète la connaissance a priori de la loi marginale  $q(x)$ .

## CONCLUSION

Dans cette contribution, nous nous sommes intéressés à la question de l'apprentissage semi-supervisé de modèles probabilistes discriminants (ou conditionnels) en essayant d'éviter toute forme d'a priori sur le rôle qui devrait être joué par les données non-étiquetées. Cette approche est assez originale dans la mesure où les critères d'estimation semi-supervisée proposés dans la littérature reposent généralement sur des postulats visant, le plus souvent, à diminuer l'incertitude de la décision pour les données non étiquetées ou à garantir que peu de données non étiquetées se trouvent "au voisinage" des frontières de décision. Le théorème 4.1 détaille le comportement asymptotique le plus favorable atteignable par un algorithme d'estimation supposé connaître la loi marginale des observations. Ce résultat fournit une confirmation du fait que les données non-étiquetées n'améliorent pas la performance asymptotique *lorsque le modèle est bien spécifié*. Le théorème 4.1 montre, a contrario, que la connaissance de la loi marginale des observations permet de construire un estimateur asymptotiquement plus efficace que le maximum de vraisemblance (conditionnel) lorsque le modèle est mal spécifié. En particulier, le théorème 4.1 confirme l'intuition que les données non-étiquetées sont le plus utiles dans les cas où l'erreur de Bayes est faible. Par ailleurs, l'avantage de l'estimateur semi-supervisé proposé est qu'il ne compromet pas la simplicité de l'approche par maximum de vraisemblance dans la mesure où le critère semi-supervisé pondéré reste convexe. Nous avons proposé un moyen pour étendre la méthode à des problèmes de plus grande dimension, en particulier des applications où les observations sont continues ou possèdent une structure plus complexe (séquence ou graphe par exemple).

Une limitation des travaux exposés ici est leur caractère asymptotique qui ne permet pas de rendre compte du comportement du critère proposé lorsque  $n$  est faible. Nous avons toutefois pu constater empiriquement qu'il pouvait être beaucoup plus favorable que ce que l'analyse asymptotique suggère. Par ailleurs, une autre voie non explorée pour l'instant consisterait à introduire des connaissances a priori dans le cadre de cette méthode. On pourrait par exemple utiliser un estimateur bayésien des probabilités conditionnelles avec un a priori destiné à lier les valeurs obtenues pour des  $x$  comparables. Cette façon de procéder qui s'inspire des estimateurs de type "back-off" utilisés pour les modèles de langage pourrait permettre de dépasser les performances de l'approche proposée ici lorsque l'hypothèse que les observations proches ont tendance à avoir des étiquettes similaires s'applique effectivement.

## Sélection de caractéristiques pour les champs aléatoires conditionnels par pénalisation $L_1$

---

Les méthodes à base d'apprentissage automatique ont profondément bouleversé la méthodologie de développement d'applications de TAL.

Les approches fondées sur l'accumulation de règles symboliques produites par des experts ont progressivement été supplantées par des méthodes numériques qui s'appuient principalement sur l'analyse statistique de corpus annotés. Le cadre d'utilisation le plus commun est celui de l'apprentissage supervisé de règles de classification, qui permettent

d’assigner une ou des étiquettes catégorielles à la représentation d’une entité linguistique. Une analyse rétrospective des avancées dans ce domaine réalisées durant la dernière décennie permet de dégager deux axes d’innovations majeurs: d’une part la diffusion des modèles d’apprentissage discriminants (ou conditionnels), qui permettent d’intégrer des traits linguistiques riches et variés; d’autre part le développement de techniques à même de traiter les dépendances statistiques qui existent entre les diverses sous-parties de représentations linguistiques structurées telles que les séquences, les arbres ou les graphes acycliques.

Sur le premier axe, citons en particulier l’introduction pour le TAL de modèles de régression logistique multinomiale (aussi appelés maxent) Rathnaparkhi (1998), des classificateurs à vaste marge (SVM) Cortes and Vapnik (1995) ou encore du boosting Freund and Schapire (1996). Du point de vue statistique, ces modèles présentent l’intérêt de résoudre directement, plutôt qu’indirectement par la règle de Bayes, le problème de classification visé, en modélisant (dans le cas des modèles d’entropie maximale) la distribution conditionnelle de la classe sachant l’observation  $p_\theta(y|x)$  sous la forme d’une distribution exponentielle selon

$$p_\theta(y|x) = \frac{\exp(\theta^T F(x, y))}{Z_\theta(x)} \quad (7)$$

Dans cette équation,  $F(x, y)$  est un vecteur de caractéristiques arbitraires de l’entrée  $x$  et de la classe  $y$ , chacune des composantes correspondant à un test atomique réalisé conjointement sur  $x$  et  $y$ ;  $\theta$  est le vecteur de paramètres correspondant, contenant une composante par caractéristique;  $Z_\theta(x)$  est un terme de normalisation qui garantit que cette formulation définit une distribution de probabilité:  $Z_\theta(x) = \sum_y p_\theta(y|x)$ . D’un point de vue computationnel, l’estimation de ces modèles discriminants conduit à résoudre des problèmes d’optimisation convexe (ce qui assure l’unicité de la solution): la maximisation de la marge pour les SVM ou la maximisation de la log-vraisemblance conditionnelle pour les modèles d’entropie maximale. Dans ce dernier cas, l’estimation des paramètres donne lieu au programme suivant (les sommes portent sur les instances d’apprentissage, indicées de  $n = 1$  à  $N$ )

$$\theta^* = \operatorname{argmin}_\theta \sum_{n=1}^N -\log(p_\theta(y^{(n)}|x^{(n)})) = \operatorname{argmin}_\theta \sum_{n=1}^N \log(Z_\theta(x^{(n)})) - \theta^T F(x^{(n)}, y^{(n)}) \quad (8)$$

Il existe, pour ces problèmes, des techniques d’optimisation bien rodées, qui permettent de trouver efficacement les valeurs optimales des paramètres y compris dans des espaces de très grande dimension. L’efficacité de ces méthodes est conditionnée par l’ajout d’un terme de *régularisation* (ou de *pénalisation*) à la fonction objectif, qui permet d’obtenir une stabilité numérique de la solution même en très grande dimension. Ce terme de régularisation prend le plus souvent la forme d’une fonction linéaire du carré de la norme  $L_2$  du vecteur de paramètres, ce qui préserve la différentiabilité et la convexité de la fonction objectif et se prête à une interprétation bayésienne Chen and Rosenfeld (2000) en termes de distribution *a priori* sur les paramètres. Concrètement, cela revient à ajouter un terme  $\rho \|\theta\|_2^2$  à la fonction objectif du programme défini en (8). Du point de vue de l’analyse linguistique enfin, ces modèles ont l’avantage de pouvoir intégrer des traits linguistiques riches et variés, permettant l’incorporation au sein du modèle de multiples sources de connaissances. En témoigne l’étude exemplaire de Toutanova and Manning (2000), qui, à partir d’une analyse serrée des erreurs commises par un étiqueteur morpho-syntaxique, spécifie un ensemble



de caractéristiques linguistiquement fondées qui conduisent à une amélioration très sensible des performances en généralisation. Cette flexibilité de modélisation a conduit au développement de modèles de complexité croissante, comportant un très grand nombre de paramètres: ainsi, l'étude précitée propose de prendre en compte non seulement des traits lexicaux (l'identité du mot à étiqueter, ou encore celle de ces voisins proches), mais également des traits typographiques, des traits visant à modéliser grossièrement la morphologie (présence de préfixes ou de suffixes de longueur bornée dans le mot), ou encore à mieux caractériser le contexte syntaxique.

Les développements menés sur le second axe visent à prendre en compte le caractère *structuré* de nombreuses représentations linguistiques, en intégrant de manière plus explicite les dépendances qui existent entre les divers sous-constituants de ces représentations (on se reportera à Bakir et al. (2007) pour un état de l'art actuel de ces techniques). Par exemple, le modèle des champs aléatoires conditionnels (CRF), introduit dans Lafferty et al. (2001), permet d'étendre les modèles d'entropie maximale à des séquences d'étiquettes. Si la forme générale du modèle reste celle donnée dans l'équation (7), les observations  $x$  et les sorties  $y$  correspondent, dans ce nouveau modèle, à des séquences complètes et les caractéristiques peuvent simultanément intégrer des tests portant sur plusieurs étiquettes au sein de la séquence à prédire. Même si, pour des raisons computationnelles, ces tests portent sur des étiquettes voisines, cette extension permet d'obtenir de nouveaux gains très significatifs en généralisation: ainsi Toutanova et al. (2003), toujours sur une tâche d'étiquetage morpho-syntaxique, rapporte que la modélisation explicite de ces dépendances conduit à une réduction de près de 2 points du taux d'erreur. Les modèles probabilistes discriminants pour les données structurées ont été généralisés pour traiter des tâches de réordonnement (ranking) Collins and Duffy (2002), Charniak and Johnson (2005), d'étiquetage d'arbres Jousse et al. (2006b), d'analyse syntaxique en constituants Rozenknop (2002), Finkel et al. (2008), d'analyse en dépendances Koo et al. (2007) ou encore d'alignement de mots en traduction automatique Blunsom and Cohn (2006). La prise en compte de dépendances dans des représentations structurées a toutefois pour conséquence directe l'augmentation massive du nombre de paramètres impliqués dans la modélisation, puisque, pour s'en tenir au simple cas des séquences, la prise en compte des dépendances entre étiquettes adjacentes requiert un nombre de paramètres qui croît comme le carré du nombre d'étiquettes possibles.

## SÉLECTION DE CARACTÉRISTIQUES

Au final, l'effet de ce double mouvement a été l'utilisation de modèles de complexité toujours croissante, intégrant typiquement des centaines de milliers, voire des millions de paramètres. Si cette augmentation de la complexité s'accompagne le plus souvent d'une amélioration des performances, elle n'en pose pas moins problème. La première difficulté est computationnelle: ces millions de caractéristiques doivent être évaluées pour chaque exemple d'apprentissage et de test; les paramètres correspondants doivent être stockés en mémoire et conduisent à des problèmes d'optimisation en très grande dimension. Estimer ces modèles conduit finalement à se placer dans une situation où le nombre de paramètres dépasse de plusieurs ordres de grandeur le nombre d'instances d'apprentissage au risque d'instabilité numérique des solutions obtenues, même en présence de régularisation (voir par exemple les difficultés rencontrées par Sha et Pereira 2003, qui construisent un modèle

intégrant près de 4 millions de caractéristiques pour une tâche d’analyse syntaxique de surface). La seconde difficulté est d’ordre statistique: la présence de caractéristiques inutiles ou redondantes dans le modèle peut conduire à des phénomènes de sur-apprentissage et amener une dégradation des performances en généralisation Kazama and Tsujii (2003). En fait, de nombreux auteurs s’étonnent d’observer des dégradations des performances lorsque certaines caractéristiques sont injectées dans le modèle (voir infra). La troisième difficulté porte sur la modélisation linguistique. Il n’existe en pratique pas de limite aux traits que l’on voudrait pouvoir inclure dans un modèle: faute de critère (autres que les performances globales) pour décider de l’utilité de tel ou tel trait, la pratique la plus répandue consiste à ajouter tous les traits possibles et imaginables (dans la limite du raisonnable) et à évaluer empiriquement l’intérêt de telle ou telle combinaison de caractéristiques. Cette situation n’est pas satisfaisante et suscite des interrogations. Ainsi, dans Toutanova and Manning (2000), déjà mentionné, les auteurs constatent qu’ajouter des caractéristiques qui testent à la fois les mots suivants et précédents conduit à une petite dégradation des performances en comparaison à n’utiliser que des tests sur le mot suivant. De même, l’utilisation de caractéristiques portant sur les préfixes semble avoir un effet négatif:

*Conversely, empirically it was found that the prefix features for rare words were having a net negative effect on accuracy. We do not at present have a good explanation for this phenomenon.*

Ces constatations plaident pour le développement de techniques de sélection automatique des caractéristiques les plus utiles. Les méthodes proposées dans la littérature pour ce faire sont toutefois très heuristiques. Une pratique commune consiste à ne conserver que les caractéristiques qui sont suffisamment fréquentes dans les données d’apprentissage. Ainsi, Toutanova and Manning (2000) impose un seuil de fréquence minimum pour considérer des caractéristiques, heuristique qui est reprise sans autre forme de discussion dans de nombreux travaux sur les modèles exponentiels: Lafferty et al. (2001) se limite ainsi à l’examen de quelques préfixes, Bender et al. (2003) utilisent la même stratégie pour sélectionner les caractéristiques incluses dans leur détecteur d’entités nommées, etc. Une approche plus fondée est proposée par McCallum and Li (2003), McCallum (2003) qui s’inspire de Della Pietra et al. (1997) pour développer un algorithme glouton de sélection des caractéristiques sur la base d’une approximation de leur contribution à la log-vraisemblance globale.

La question de la sélection automatique de variables explicatives a pourtant donné lieu à une vaste littérature dans le domaine des statistiques, et au développement de méthodes efficaces Guyon and Elisseeff (2003). Parmi celles-ci, une approche initialement introduite dans un cadre de régression linéaire Tibshirani (1996) consiste à employer une pénalisation de la norme  $L_1$  du vecteur de paramètres. Concrètement, cela revient à ajouter dans la fonction objectif un terme de la forme  $\rho\|\theta\|_1$  en place de  $\rho\|\theta\|_2^2$ . Ce changement a pour effet d’annuler tous les paramètres dont la contribution à la log-vraisemblance est insuffisante pour contrebalancer le “coût” de leur inclusion dans le modèle, alors qu’avec une pénalisation  $L_2$ , ces paramètres prennent des valeurs arbitrairement faibles, mais non nulles. Seules les caractéristiques associées à des paramètres non-nuls sont alors sélectionnées. Le comportement particulier de l’optimisation avec cette forme de régularisation est, en dernière analyse, du à la non différentiabilité du terme de régularisation en tout point où l’une des coordonnées de  $\theta$  est nulle (voir également sur ce point la discussion de (Hastie et al., 2001, p.68 et suivantes)) . Cette propriété, malheureusement, interdit l’utilisation de techniques usuelles d’optimisation, qui présupposent

l'existence du gradient de la fonction objectif. Pour les modèles d'entropie maximale, des techniques alternatives d'optimisation des paramètres qui restent valides dans ce cas ont été récemment développées Kazama and Tsujii (2003), Dudík et al. (2004), Riezler and Vasserman (2004), Friedman et al. (2007): nous étendons ici la dernière de ces propositions au cas des CRF.

## CHAMPS ALÉATOIRES CONDITIONNELS POUR L'ÉTIQUETAGE DE SÉQUENCES

### Champs aléatoires conditionnels

Les champs aléatoires conditionnels Lafferty et al. (2001), Sutton and McCallum (2006) correspondent à un modèle discriminant de prédiction supervisée de séquences appartenant à la famille logistique généralisée ou d'entropie maximale. Supposons donnée une séquence d'entrée  $\mathbf{x} = (x_1, \dots, x_T)$  ainsi qu'une séquence d'étiquettes à prédire  $\mathbf{y} = (y_1, \dots, y_T)$ . Le modèle dit d'ordre un ou *linear chain* instancie l'équation (7) en postulant une distribution de probabilité conditionnelle de la séquence d'étiquettes donnée par

$$p_{\theta}(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\theta}(\mathbf{x})} \exp \left\{ \sum_{t=1}^T \sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t) \right\} \quad (9)$$

Dans l'équation ci-dessus,  $Z_{\theta}(\mathbf{x})$  désigne le facteur de normalisation défini par

$$Z_{\theta}(\mathbf{x}) = \sum_{\mathbf{y} \in Y^T} \exp \left\{ \sum_{t=1}^T \sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t) \right\} \quad (10)$$

où  $Y$  désigne l'ensemble des valeurs prises par  $y_t$  (de même  $X$  désignera l'ensemble des valeurs prises par  $x_t$ ). Par convention, l'étiquette  $y_0$  correspond à une valeur conventionnelle toujours observée indiquant le début de séquence et on note  $\bar{Y}$  l'ensemble  $Y$  complété par cette étiquette de début de séquence. Dans l'équation (9),  $\theta = (\theta_1, \dots, \theta_K)$  désigne le vecteur de paramètres du modèle tandis que les fonctions  $f_k$  correspondent aux caractéristiques sur lesquelles la prédiction des étiquettes va reposer. Par rapport à la structure très générale de l'équation (7), la contrainte principale qui apparaît dans (9), et justifie le terme de modèle d'ordre un, est due au fait que chaque caractéristique ne fait intervenir, au plus, que des bigrammes d'étiquettes successives  $(y_{t-1}, y_t)$ . Ce choix implique que la loi conditionnelle  $p_{\theta}(\mathbf{y}|\mathbf{x})$  possède une structure d'indépendance conditionnelle dans laquelle, sachant  $\mathbf{x}$ ,  $y_{t-1}$  et  $y_{t+1}$ ,  $y_t$  est indépendant de  $y_s$  pour  $s < t - 1$  ou  $s > t + 1$  et sa loi ne dépend que de  $x_t$ ,  $y_{t-1}$  et  $y_{t+1}$ . La raison de ce choix est essentiellement de nature computationnelle. En ce qui concerne la dépendance des caractéristiques vis à vis de la séquence d'entrée  $\mathbf{x}$ , les possibilités sont en fait beaucoup plus larges et la forme retenue pour l'équation (9) ne constitue qu'un exemple, choisi pour sa simplicité, où chaque caractéristique est une fonction du triplet  $(y_{t-1}, y_t, x_t)$ . En pratique, et selon les applications envisagées, il est fréquent que les caractéristiques ne portent pas uniquement sur la valeur  $x_t$  de la séquence d'entrée à la position  $t$  mais plutôt sur le contenu de la séquence autour de la position  $t$ , par exemple sur le trigramme  $(x_{t-1}, x_t, x_{t+1})$ . Nous rencontrerons un autre cas de figure fréquent où la séquence d'entrée est en fait multimodale,  $x_t = (x_t^1, \dots, x_t^D)$  et où on utilise une superposition de caractéristiques portant

sur chacune des modalités de la séquence d'entrée en remplaçant  $\sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t)$  dans (9) par  $\sum_{k=1}^K \sum_{d=1}^D \theta_k^d f_k^d(y_{t-1}, y_t, x_t^d)$ . Nous verrons cependant ci-dessous qu'en ce qui concerne l'estimation des paramètres  $\theta$  du modèle, la dépendance des caractéristiques vis à vis de la séquence d'entrée ne pose pas de problème particulier dans la mesure où on suppose toujours cette séquence observée. Pour des raisons de simplicité d'écriture, nous conservons donc la forme présentée dans l'équation (9) qui permet d'illustrer l'ensemble des enjeux liés à l'utilisation des champs aléatoires conditionnels d'ordre un.

### Choix des caractéristiques

Dans le cadre du traitement automatique des langues, où les séquences tant d'entrée que de sortie sont assimilables à des variables catégorielles, le choix le plus naturel pour les caractéristiques  $f_k$  consiste à utiliser des fonctions booléennes qui valent 1 ou 0 selon que le triplet  $(y_{t-1}, y_t, x_t)$  est ou n'est pas dans une configuration particulière. Plus précisément, nous considérerons deux types de caractéristiques, dites respectivement de type *unigramme* et *bigramme*, ce qui permet de décomposer le terme  $\sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t)$  selon

$$\begin{aligned} \sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t) &= \sum_{y \in Y, x \in X} \mu_{y,x} \mathbb{1}(y_t = y, x_t = x) \\ &+ \sum_{(y', y) \in \bar{Y} \times Y, x \in X} \lambda_{y', y, x} \mathbb{1}(y_{t-1} = y', y_t = y, x_t = x) \end{aligned} \quad (11)$$

où  $\mathbb{1}(\text{test}) = 1$  si le test est positif et vaut 0 sinon. Dans l'expression ci-dessus, le vecteur de paramètres  $\mu = (\mu_{y,x})_{y \in Y, x \in X}$  correspond aux caractéristiques de type unigramme qui testent la cooccurrence d'une étiquette particulière  $y$  et d'une entrée  $x$  à la même position. Le vecteur de paramètres  $\lambda = (\lambda_{y', y, x})_{(y', y) \in \bar{Y} \times Y, x \in X}$  correspond aux caractéristiques de type bigramme qui testent la succession de deux étiquettes conjointement avec l'occurrence d'une valeur particulière de l'entrée. Il est clair également que dans (11) la sommation sur les caractéristiques est dorénavant relativement fictive puisque l'on pourrait réécrire (11) sous la forme  $\mu_{y_t, x_t} + \lambda_{y_{t-1}, y_t, x_t}$ .

Quelques commentaires s'imposent. Tout d'abord il n'est pas évident à ce stade que l'utilisation simultanée de caractéristiques des deux types soit nécessaire dans la mesure où pour toute valeur de  $\mu$  et  $\lambda$ , il existe une valeur  $\lambda'$  des paramètres pour laquelle la partie bigramme seule réalise de façon équivalente (11) (en prenant  $\lambda'_{y', y, x} = \lambda_{y', y, x} + \mu_{y, x}$ ). Nous verrons cependant que l'utilisation simultanée des deux types de caractéristiques est nécessaire pour obtenir des jeux de caractéristiques réduits, conduisant à de bonnes performances de classification. Notons également que l'utilisation des caractéristiques unigramme seules correspondrait à un modèle plus simple de régression logistique position par position dans lequel

$$p_{\mu}(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^T \frac{\exp(\mu_{y_t, x_t})}{\sum_{y \in Y} \exp(\mu_{y, x_t})}$$

L'aspect séquentiel du modèle, qui tient compte des corrélations entre étiquettes successives est donc uniquement le fait des caractéristiques de type bigramme. Par ailleurs, les caractéristiques unigramme et bigramme sont en nombres très différents puisque le total des caractéristiques unigramme disponibles est de  $|Y||X|$  (où  $|Y|, |X|$  désignent le

cardinal de  $Y$  et  $X$ ) tandis qu'il y a  $|Y|(|Y| + 1)|X|$  caractéristiques de type bigramme, le terme  $(|Y| + 1)$  venant de l'utilisation de l'étiquette supplémentaire qui correspond au début de séquence. Les caractéristiques de type bigramme sont donc très majoritaires numériquement dès que le nombre d'étiquettes distinctes est important. On peut se demander si ces effectifs théoriques de caractéristiques ne peuvent pas être fortement élagués en pratique au vu des fréquences d'occurrence de ces caractéristiques dans les jeux de données disponibles pour l'entraînement des modèles. Quelle que soit la méthode d'inférence utilisée, il est aisé de vérifier que les caractéristiques unigramme ou bigramme portant sur une éventuelle modalité d'entrée  $x$  qui n'est jamais observée dans le corpus d'apprentissage peuvent être éliminées sans aucun risque, puisque les paramètres correspondant  $\mu_{y,x}$  et  $\lambda_{y',y,x}$  seront de toute façon estimés à zéro. Toute autre forme d'élagage *a priori* basé sur les statistiques d'occurrence dans le corpus d'apprentissage modifie par contre le résultat d'estimation. En particulier, le fait de forcer  $\mu_{y,x}$  à zéro même si l'occurrence  $(y_t = y, x_t = x)$  n'est jamais apparue dans le corpus d'apprentissage peut avoir des conséquences importantes si le motif  $(y_t = y', x_t = x)$  est fréquent pour d'autres valeurs  $y'$  de l'étiquette. Un des intérêts de l'approche discutée dans cette étude est précisément de sélectionner des caractéristiques pertinentes de façon beaucoup plus efficace qu'en se contentant d'examiner, *a priori*, les fréquences d'occurrence des motifs.

Dans la suite, nous utiliserons à la fois la représentation en termes des caractéristiques unigramme et bigramme paramétrée par  $\mu$  et  $\lambda$  et, lorsque c'est plus simple, la représentation totalement vectorisée dans laquelle les deux types de caractéristiques ne sont pas distinguées et  $\theta$  désigne l'ensemble des paramètres du modèle.

## ALGORITHME D'OPTIMISATION COORDONNÉE PAR COORDONNÉE POUR LE CRITÈRE AVEC PÉNALITÉ $L_1$

### Choix de la pénalisation

L'approche la plus commune pour estimer le paramètre  $\theta$  consiste à ajouter un terme de pénalisation  $L_2$  au critère de perte logarithmique, auquel cas la fonction objectif à minimiser devient  $l(\theta) + \rho_2 \|\theta\|_2^2$ , où  $\rho_2$  est un paramètre de régularisation. Outre ses bonnes performances empiriques, l'intérêt pratique de cette approche est que l'évaluation de la fonction objectif et de son gradient nécessitent les mêmes calculs que dans le cas de  $l(\theta)$  et n'importe quelle approche numérique de minimisation d'une fonction différentiable et convexe, de surcroît sans contrainte de domaine, peut être utilisée. Les limitations principales de cette approche standard sont, d'une part, liées au temps d'exécution avec la nécessité de réaliser la récursion *forward-backward* pour toutes les séquences d'apprentissage lors de chaque évaluation de la fonction ou du gradient et, d'autre part, liées à l'empreinte mémoire du code du fait de la taille habituellement très grande du vecteur de paramètres. En pratique, ce deuxième aspect interdit l'utilisation d'algorithmes cherchant à estimer directement le hessien ou son inverse et se traduit par l'utilisation prépondérante d'algorithmes de type gradient conjugué ou quasi-Newton à mémoire limitée (de type L-BFGS, *Limited Memory BFGS*, en particulier).

La pénalisation  $L_2$ , si elle est efficace pour éviter le sur-apprentissage au moment de l'entraînement du modèle, ne réalise pas à proprement parler de sélection de car-

actéristiques : les paramètres  $\theta_k$  estimés sont tous non nuls et la sélection de caractéristiques basée sur leur amplitude conduit à des performances relativement dégradées. Pour effectivement incorporer l'objectif de sélection de caractéristiques dans le terme de pénalisation, nous considérons l'inclusion d'un terme de pénalité additionnel de type  $L_1$ . La fonction objectif obtenue

$$l(\theta) + \rho_1 \|\theta\|_1 + \frac{\rho_2}{2} \|\theta\|_2^2 \quad (12)$$

est dite *elastic net* par Zou and Hastie (2005) et comporte maintenant deux paramètres de régularisation  $\rho_1$  et  $\rho_2$ <sup>1</sup>. Cette forme de pénalisation fournit un degré de liberté supplémentaire pour ajuster le compromis entre le caractère creux des solutions et la qualité des performances en généralisation. En particulier, elle permet d'atteindre, pour  $\rho_1 = 0$  ou  $\rho_2 = 0$  les solutions obtenues avec chacune des pénalités utilisée séparément, mais rend également accessible d'autres solutions qui sont peut-être plus intéressantes. Dans la mesure toutefois où le terme de pénalisation  $L_1$  joue un rôle déterminant pour la sélection de caractéristiques, nous continuerons, par la suite, à parler de *modèle avec régularisation  $L_1$*  pour désigner notre modèle.

### Algorithme d'optimisation coordonnée par coordonnée

Le principe de l'approche proposée par Friedman et al. (2008) consiste à remarquer que si la minimisation directe du critère (12) est un problème délicat du fait de la non différentiabilité de la fonction objectif en les points où l'un au moins des  $\theta_k$  vaut zéro, la minimisation coordonnée par coordonnée d'une approximation quadratique locale de (12) est un problème très simple qui admet une résolution explicite. L'idée est que l'inefficacité intrinsèque de la minimisation coordonnée par coordonnée peut être compensée par l'extrême simplicité de la mise à jour à effectuer pour chaque coordonnée associée à la possibilité d'utiliser des schémas plus efficaces de balayage des coordonnées, possibilité dont on verra qu'elle est particulièrement attractive dans les cas des champs aléatoires conditionnels.

étant donnée une valeur courante  $\bar{\theta}$  du vecteur de paramètres, l'approximation quadratique locale vis à vis de la  $k$ -ième coordonnée prend la forme suivante

$$l_{k,\bar{\theta}}(\theta_k) = C^{st} + \frac{\partial l(\bar{\theta})}{\partial \theta_k} (\theta_k - \bar{\theta}_k) + \frac{1}{2} \frac{\partial^2 l(\bar{\theta})}{\partial \theta_k^2} (\theta_k - \bar{\theta}_k)^2 + \rho_1 |\theta_k| + \frac{\rho_2}{2} \theta_k^2 \quad (13)$$

En écrivant les conditions d'optimalité au premier ordre (dite Karush-Kuhn-Tucker), il est aisé de vérifier que le minimum de l'approximation quadratique locale (13) est atteint en

$$\theta_k = \frac{s \left\{ \frac{\partial^2 l(\bar{\theta})}{\partial \theta_k^2} \bar{\theta}_k - \frac{\partial l(\bar{\theta})}{\partial \theta_k}, \rho_1 \right\}}{\frac{\partial^2 l(\bar{\theta})}{\partial \theta_k^2} + \rho_2} \quad (14)$$

où  $s$  désigne la fonction de seuillage progressif ou seuillage doux définie par

$$s(z, \rho) = \begin{cases} z - \rho & \text{if } z > \rho \\ z + \rho & \text{if } z < -\rho \\ 0 & \text{sinon} \end{cases} \quad (15)$$

---

<sup>1</sup>Zou and Hastie (2005) utilisent une paramétrisation différente de la régularisation qui, dans le type de problèmes considérés ici, s'est avérée plus difficile à régler car chaque paramètre joue simultanément sur les deux types de régularisation.

Il est intéressant de noter que Dudík et al. (2004) présente une version alternative de la même idée dans laquelle le comportement local de  $l$  est approximé sous une forme, équivalente au premier ordre, mais non quadratique qui conduit elle aussi à une minimisation coordonnée par coordonnée explicite. Cette forme d'approximation repose cependant cruciallement sur le fait que chaque coordonnée  $\theta_k$  du vecteur de paramètres est multipliée par une caractéristique à valeur dans  $\{0, 1\}$ . Cette propriété n'est malheureusement pas vérifiée dans les cas des champs aléatoires conditionnels, puisque la  $k$ -ième coordonnée du vecteur de paramètres est multipliée par  $\sum_{t=1}^T f_k(y_{t-1}, y_t, x_t)$ . Ce terme peut être strictement supérieur à 1, même si  $f_k$  est à valeur dans  $\{0, 1\}$ , dès que la caractéristique correspondante est présente à plusieurs positions distinctes dans la séquence d'apprentissage.

### Applications aux champs aléatoires conditionnels

Pour appliquer l'approche précédente dans le cas des champs aléatoires conditionnels, il est nécessaire de disposer des dérivées d'ordre un et deux de  $l(\theta)$ . Nous considérons maintenant le cas de la dérivée seconde de  $l(\theta)$ . Un calcul direct donne

$$\frac{\partial^2 l(\theta)}{\partial \theta_k^2} = \sum_{n=1}^N \left\{ E_{p_\theta(\mathbf{y}|\mathbf{x}^{(n)})} \left( \sum_{t=1}^{T_n} f_k(y_{t-1}, y_t, x_t^{(n)}) \right)^2 - \left( E_{p_\theta(\mathbf{y}|\mathbf{x}^{(n)})} \sum_{t=1}^{T_n} f_k(y_{t-1}, y_t, x_t^{(n)}) \right)^2 \right\} \quad (16)$$

Le premier terme est problématique, car il implique des termes qui ne dépendent pas uniquement des probabilités jointes conditionnelles  $P_\theta(y_{t-1} = y', y_t = y | \mathbf{x}^{(n)})$  et ne sont donc pas calculables à partir de la récursion *forward-backward*. Même si des solutions de calcul exactes existent (cf. chapitre 4 de Cappé et al. (2005)), celles-ci ne semblent pas praticables étant donné l'échelle des modèles auxquels nous nous intéressons ici et nous proposons d'utiliser l'approximation suivante

$$\frac{\partial^2 l(\theta)}{\partial \theta_k^2} \approx \sum_{n=1}^N \sum_{t=1}^{T_n} \left\{ E_{p_\theta(\mathbf{y}|\mathbf{x}^{(n)})} f_k^2(y_{t-1}, y_t, x_t^{(n)}) - \left( E_{p_\theta(\mathbf{y}|\mathbf{x}^{(n)})} f_k(y_{t-1}, y_t, x_t^{(n)}) \right)^2 \right\} \quad (17)$$

Cette approximation correspond à l'hypothèse que conditionnellement à  $\mathbf{x}^{(n)}$ , les caractéristiques  $f_k(y_{t-1}, y_t, x_t^{(n)})$  et  $f_k(y_{s-1}, y_s, x_s^{(n)})$  sont décorrélées dès que  $s \neq t$ . Lorsqu'on utilise des caractéristiques unigramme ou bigramme, cette approximation est exacte pour la  $n$ -ième séquence d'apprentissage dès lors que  $\theta_k$  correspond soit à un paramètre unigramme  $\mu_{y,x}$  soit à paramètre bigramme  $\lambda_{y',y,x}$  pour lequel le symbole  $x$  n'est présent qu'à une unique position dans la séquence d'apprentissage  $\mathbf{x}^{(n)}$ . On remarque d'ailleurs également que si le symbole  $x$  n'est pas présent dans la séquence d'apprentissage  $\mathbf{x}^{(n)}$ , celle-ci ne contribue en aucune façon à la minimisation de l'approximation quadratique locale. Cette remarque essentielle nous permet, lorsque l'on met à jour un paramètre  $\mu_{y,x}$  ou  $\lambda_{y',y,x}$ , de limiter la sommation dans (17) aux séquences dans lesquelles le symbole  $x$  apparaît, c'est à dire de n'effectuer la récursion *forward-backward* que pour les séquences

correspondantes. En terme de temps de calcul, le gain est donc éventuellement très significatif, même s’il ne compense pas la nécessité de remettre à jour successivement chacune des coordonnées du vecteur de paramètres. L’algorithme correspondant est décrit ci-dessous.

### Mise à jour simultanée de blocs de coordonnées

L’utilisation de l’algorithme décrit ci-dessus bute sur la très grande dimensionalité du vecteur de paramètres utilisé dans les applications. Dans le cas d’une utilisation conjointe de caractéristiques unigramme et bigramme selon (11), le nombre total de paramètres est de  $|Y||X|$  pour les caractéristiques unigramme, plus  $|Y|(|Y| + 1)|X|$  pour les caractéristiques bigramme. Même si la mise à jour de chaque coordonnée n’implique qu’un nombre réduit de séquences parmi l’ensemble des séquences d’apprentissage, la dimension excessive du vecteur de paramètres rend difficilement envisageable la mise à jour coordonnée par coordonnée. Pour imaginer des schémas plus efficaces de mise à jour des coordonnées bloc par bloc, il est important de noter que pour mettre à jour un paramètre de type unigramme  $\mu_{y,x}$  ou de type bigramme  $\lambda_{y',y,x}$ , il est nécessaire d’effectuer la récursion *forward-backward* pour le sous-ensemble des séquences d’apprentissage qui comportent le symbole  $x$ . Or, on constate à l’examen du gradient et (17) que le coût de calcul de la dérivée première et de l’approximation de la dérivée seconde est marginal une fois que les probabilités jointes conditionnelles  $P_\theta(y_{t-1} = y', y_t = y | \mathbf{x}^{(n)})$  ont été obtenues pour l’ensemble des indices  $n$  pour lesquels la séquence  $\mathbf{x}^{(n)}$  contient le symbole  $x$ . En d’autres termes, pour un surcoût de calcul marginal, il est possible d’évaluer simultanément le gradient et (17) pour l’ensemble des paramètres  $(\mu_{y,x})_{y \in Y}$  et  $(\lambda_{y',y,x})_{(y',y) \in \bar{Y} \times Y}$ . Cette observation conduit à regrouper les caractéristiques par blocs correspondant à l’ensemble des caractéristiques unigramme ou bigramme qui partagent un même symbole d’entrée  $x$ . Il est intéressant de constater que cette contrainte computationnelle conduit à choisir des blocs de caractéristiques qui sont orthogonaux à ceux utilisés dans le cas de la régression logistique par Friedman et al. (2008) dans lequel les caractéristiques sont regroupées par valeur commune de l’étiquette  $y$ . L’algorithme correspondant est donné ci-dessous.

Le coût de calcul associé à une itération de cet algorithme, l’itération correspondant à la mise à jour de toutes les coordonnées du vecteur de paramètres, est donc de l’ordre de  $|X|$  (le nombre de symboles d’entrée) multiplié par le nombre moyen de séquences d’apprentissage contenant chaque symbole. Dans les expériences, ce coût est en moyenne du même ordre de grandeur que celui associé à chaque itération d’un algorithme d’optimisation globale du vecteur  $\theta$  qui requiert le traitement de l’ensemble des  $N$  séquences d’apprentissage pour l’évaluation du vecteur gradient de  $l(\theta)$ .

Un problème qui peut survenir lors de l’utilisation de l’algorithme est celui de l’instabilité numérique qui se manifeste par une convergence peu régulière vers la solution pour certaines valeurs de l’initialisation. L’interprétation à donner de ce phénomène est liée à la mise à jour simultanée d’un bloc de coordonnées ce qui revient à approcher le hessien de chaque bloc par une matrice diagonale dont les éléments diagonaux sont donnés par (17). Il est connu que dans ce type d’algorithme de mise à jour par blocs avec calcul approché du hessien, il est en général nécessaire d’ajuster le pas de l’algorithme pour garantir la stabilité globale de l’optimum Nocedal and Wright (2006). Dans le cas qui nous préoccupe, la stabilité peut être garantie en recherchant, pour chaque coordonnée, la valeur  $0 < \alpha_k \leq 1$



la plus grande possible du pas telle que la mise à jour

$$s \left( \frac{\alpha_k^{-1} \frac{\partial^2 l(\bar{\theta})}{\partial \theta_k^2} \bar{\theta}_k - \frac{\partial l(\bar{\theta})}{\partial \theta_k}, \rho_1 \right) \\ \alpha_k^{-1} \frac{\partial^2 l(\bar{\theta})}{\partial \theta_k^2} + \rho_2$$

conduise effectivement à une diminution de la fonction objectif. Cette façon de procéder conduirait toutefois à une mise à jour au coût prohibitif, impliquant en particulier la nécessité de recalculer  $l(\theta)$  de façon répétée lors de chaque mise à jour. Face à ce problème nous avons utilisé une solution heuristique consistant à fixer le pas  $\alpha$  globalement pour le bloc complet des paramètres remis à jour simultanément en s'ajustant sur la taille du plus grand pas potentiellement effectué par l'algorithme de Newton ignorant les termes de pénalités; c'est à dire en utilisant

$$\alpha^{-1} = \kappa \times \max \left\{ 1, \max \left( \left| \frac{\partial l(\bar{\theta})}{\partial \theta_k} \right| / \left| \frac{\partial^2 l(\bar{\theta})}{\partial \theta_k^2} \right| \right) \right\}$$

Cette heuristique utilisée avec  $\kappa = 1.5$  conduit à un algorithme très stable, avec des pas de taille suffisamment grande pour ne pas trop ralentir la convergence : typiquement  $\alpha^{-1}$  peut être de l'ordre de plusieurs centaines initialement lorsque le paramètre est très mal estimé mais se fixe, lorsque l'on approche de la convergence, à des valeurs de l'ordre de 5.

## BILAN ET CONCLUSION

Dans cette étude, nous avons proposé un nouvel algorithme pour réaliser l'étape d'estimation dans les modèles CRF en présence d'une régularisation  $L_1$ . Nos résultats sont conformes à l'état de l'art, en ce sens qu'ils démontrent qu'il est possible d'élaguer très fortement les paramètres du modèle sans dégrader de manière significative les performances. En revanche, pour les données de test et les caractéristiques utilisées, nous n'avons pas observé de cas où l'utilisation d'une pénalité  $L_1$  améliore les performances par rapport à l'utilisation d'une pénalité  $L_2$ . Nous avons également comparé cette méthode de sélection avec des méthodes heuristiques usuellement utilisées et démontré sa supériorité empirique.

L'autre contribution de ce travail a été d'analyser les caractéristiques qui sont extraites, ce qui nous a permis de constater qu'elles avaient le plus souvent une interprétation linguistique claire (pour les associations positives); à l'inverse, les paramètres négatifs sont plus délicats à interpréter, dans la mesure où la valeur de ces caractéristique est le plus souvent fixée par compensation avec une ou plusieurs associations positives.

Pour ces deux raisons, il semble que l'utilisation d'une pénalité  $L_1$  permette simultanément d'estimer de manière robuste des modèles discriminants, tout en effectuant une sélection parmi les caractéristiques les plus utiles. Ces deux facteurs plaident simultanément pour son utilisation systématique (à la place d'une pénalité  $L_2$ ) dans tous les problèmes d'apprentissage dans lesquels le nombre de caractéristiques potentiellement utiles est très grand, comme c'est souvent le cas dans des applications de traitement des langues.

---

# CHAPTER 1

## INTRODUCTION

---

### Contents

---

<b>1.1</b>	<b>Motivation</b>	<b>1</b>
<b>1.2</b>	<b>Contributions</b>	<b>2</b>
<b>1.3</b>	<b>Structure of the Thesis</b>	<b>3</b>

---

### 1.1 MOTIVATION

---

Ambiguity is the major problem in natural language processing, and it is present on all linguistic levels: phonetic (and phonological), morphological, syntactic, semantic, and pragmatic. An illustration of phonological ambiguity are lexical items with different graphic but similar phonetic transcriptions (e.g. *ice cream* and *I scream*). Morphological ambiguity comes from internal structure of a word, for example, *look* can be infinitive, first or second person singular or plural. Syntactic ambiguity can be illustrated by the short sentence *I saw a cat with a telescope* which can be parsed in two different ways represented by Figure 1.1, and therefore, two interpretations: either I used a telescope, or the cat. Semantic ambiguity, e.g., lexical semantics ambiguity comes from that a lexical item can have several meanings, for example, a noun *party*. Ambiguity on the level of expression is ambiguity of pragmatics, e.g., novels of Graham Greene telling about tender murderers and pious atheists contain oxymorons on the phrase as well as on the whole text level.

The ambiguous nature of language demands modeling of uncertainty. Probabilistic methods, that is, methods designed to provide a probabilistic confidence measure associated with each decision, are a natural and effective solution. Probabilistic models, especially discriminative probabilistic models, are used in numerous natural language applications. Discriminative probabilistic approaches (among them maximum entropy models, conditional random fields) model directly probability of a label  $y \in \mathcal{Y}$  given an observation  $x \in \mathcal{X}$ .

The models are trained on large sets of labeled data. Labeled data are expensive to produce and always limited, unlabeled data are plentiful and cheap. Semi-supervised

learning is a natural approach to address the problem of lack of labeled data. Therefore, the first problem which motivated our research is **how to use efficiently unlabeled data**.

Probabilistic models such as conditional random fields introduced by Lafferty et al. (2001) allow to model structure and arbitrary numerous dependencies. The disambiguation rules involve complex patterns of features that are partially redundant. As we will see, models are sparse what motivates model selection. Hence, the second problem considered in this thesis is **how to cope with the unnecessarily huge number of structural dependencies and how to achieve computational efficiency if the vector of parameters is sparse**.

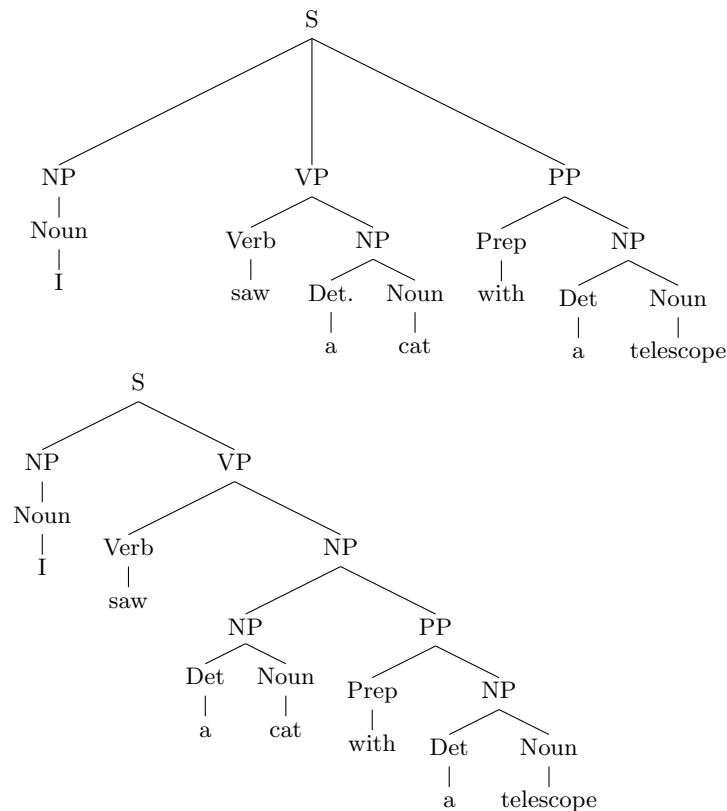


Figure 1.1: Example of ambiguity in sentence parsing: two possible parsings of a same sentence.

## 1.2 CONTRIBUTIONS

---

In this thesis, we consider probabilistic discriminative approaches, and our contributions are twofold.

- **Semi-Supervised Learning.** Based on the assumption that the number of unlabeled data is sufficient to estimate the true marginal distribution of observations, we propose a novel semi-supervised criterion for a discriminative probabilistic model.

We demonstrate that the criterion achieves the minimal variance and is asymptotically preferable to the standard logistic regression. We compare the performance of the introduced criterion to the standard logistic regression and carry out experiments on synthetic and real world data (Spam Assassin corpus).

- **Sparsity and Model Selection.** Since penalties including the  $L_1$  norm are not differentiable in zero and numerical methods can not be applied directly, we explore coordinate-wise and blockwise gradient descent methods for conditional random fields penalized by the elastic net. The second order approximation includes the second order derivative which has to be recomputed at every iteration of an optimization procedure. We propose to group variables so as to perform blockwise rather than coordinate-wise optimization. We discuss approximation of the matrix of the second derivatives by its diagonal term. We use the sparsity of the vector of parameters to speed up the forward-backward algorithm. The experiments were carried out on Nettalk, CoNLL 2000, and CoNLL 2003 data sets. The results show that blockwise descent is competitive and produces sparse and interpretable models.

## 1.3 STRUCTURE OF THE THESIS

---

This thesis consists of three parts.

1. The first part is dedicated to aspects of statistical learning in generative and discriminative models. The models can be represented as graphical, directed and undirected, models (Lauritzen, 1996, Jordan et al., 1999, Jordan, 1999, Wainwright and Jordan, 2003). Chapter 2 discusses briefly the probabilistic models for supervised learning, the naive Bayes and the logistic regression, and models for sequential processing, hidden Markov models (Bilmes, 1998, Cappé et al., 2005) and maximum entropy Markov models (McCallum et al., 2000).

Chapter 3 introduces the model of conditional random fields, the framework that allows to take arbitrary sequential dependencies into account. We provide training and inference approaches for conditional random fields, and some results on standard data sets in the domain of natural language processing.

2. The second part of the thesis is devoted to problems of semi-supervised learning. In Chapter 4, we provide a brief overview of the state-of-the-art approaches and ideas concerning semi-supervised learning (Chapelle et al., 2006). We introduce our semi-supervised criterion.

In Chapter 5, we test the introduced semi-supervised criterion that asymptotically achieves the minimal variance on synthetic and real-world data. We compare it to the standard logistic regression, and to the Shimodaira weighted criterion (Shimodaira, 2000) which proposes a compensation for the covariate shift case. We discuss briefly some difficulties encountered when applying the semi-supervised criterion to real data sets.

Chapters 4 and 5 are partly based on the papers

- *N. Sokolovska, O. Cappé, and F. Yvon.* The asymptotics of semi-supervised learning in discriminative probabilistic models. In A. McCallum and S. Roweis, editors, Proc. Int. Conf. Machine Learning (ICML), pages 984-991. Omnipress, 2008.
  - *N. Sokolovska, O. Cappé, and F. Yvon.* Analyse asymptotique de l'apprentissage semi-supervisé pour les modèles probabilistes discriminants. In Proceedings of Conférence d'Apprentissage (CAP), Porquerolles, France, 2008.
3. The third part of the thesis concerns the sparsity of models and ways to discover sparsity patterns. Chapter 6 illustrates the existence of sparsity patterns on real data and discusses state-of-the-art of the methods that return sparse parameter vectors, including the  $L_1$  and elastic net penalties. We examine optimization approaches and among them a class of coordinate-wise optimization procedures.

Chapter 7 illustrates the results of our model selection experiments on artificial and real applications. We apply blockwise coordinate descent to conditional random fields. The number of dependencies that can be eliminated from a model without performance degradation is impressive. We achieve an acceptable performance keeping a relatively small number of parameters. We compare our results to those obtained using simple heuristics as well as to the orthant-wise limited quasi Newton approach introduced in (Andrew and Gao, 2007).

Chapters 6 and 7 are partly based on the papers

- *N. Sokolovska, T. Lavergne, O. Cappé, and F. Yvon.* Efficient learning of sparse conditional random fields for supervised sequence labeling. Submitted to Journal of Selected Topics in Signal Processing, 2009.
- *N. Sokolovska, O. Cappé, and F. Yvon.* Sélection de caractéristiques pour les champs aléatoires conditionnels par pénalisation  $L_1$ . Accepted to Traitement Automatique des Langues. Volume 50, Number 3/2009.

Chapter 8 provides our conclusions and discusses some future directions.

## Part I

# Learning in Discriminative and Generative Models



---

## CHAPTER 2

# DISCRIMINATIVE AND GENERATIVE LEARNING IN PROBABILISTIC GRAPHICAL MODELS

---

### Contents

---

<b>2.1 Discriminative and Generative Learning . . . . .</b>	<b>8</b>
2.1.1 Generative models . . . . .	8
2.1.2 Discriminative models . . . . .	9
<b>2.2 Logistic Regression and Naive Bayes for Classification Tasks .</b>	<b>9</b>
2.2.1 Logistic Regression . . . . .	9
2.2.2 Naive Bayes Classifier . . . . .	12
<b>2.3 Graphical Models . . . . .</b>	<b>12</b>
2.3.1 Directed Graphs . . . . .	14
2.3.2 Undirected Models . . . . .	15
<b>2.4 Hidden Markov Models and Maximum Entropy Markov Mod-</b>	<b>16</b>
<b>els . . . . .</b>	<b>16</b>
2.4.1 Hidden Markov Models . . . . .	16
2.4.2 Maximum Entropy Markov Models . . . . .	19
<b>2.5 Conclusions . . . . .</b>	<b>21</b>

---

In this chapter, we explore probabilistic parametric learning approaches. A model with a finite number of parameters is used to predict an output  $y \in \mathcal{Y}$  given an input  $x \in \mathcal{X}$ . Probabilistic machine learning methods rely on probability theory in the procedures of training and inference. Uncertainty in the prediction can be measured in terms of probability.

We investigate generative and discriminative approaches, usually opposed to each other in the context of probabilistic learning for classification tasks. The former aims at estimating joint distribution and uses Bayes rule, the latter aims at estimating conditional distribution and performs direct inference. Both a joint and a conditional distributions can be associated with a graphical structure, which allows to visualize dependencies and to represent the distributions by a product of local functions.



In this chapter, we consider and compare supervised generative and discriminative learning methods such as the naive Bayes and the logistic regression, as well as approaches that take more elaborate sequential dependencies into account, namely hidden Markov models and maximum entropy Markov models.

## 2.1 DISCRIMINATIVE AND GENERATIVE LEARNING

---

In many applications of machine learning, the goal is to assign to an observation  $x \in \mathcal{X}$  a label  $y \in \mathcal{Y}$ . The estimation is performed by a parameterized decision function. The parameters are learnt from a set of training input data  $X = \{x_1, \dots, x_N\}$  and a corresponding set of labels  $Y = \{y_1, \dots, y_N\}$ . Generative and discriminative learning approaches widely used nowadays are usually opposed to each other; in (Jebara, 2004), for instance, they are even presented as two different schools of thought.

### 2.1.1 GENERATIVE MODELS

Generative probabilistic models (Naive Bayes, mixtures of multinomials, hidden Markov models, Markov random fields, etc.) design a joint probability distribution and produce a probability density model over all variables. An alternative name of a “generative approach” is “informative approach”, as mentioned in (Rubinstein and Hastie, 1997).

The optimal way to assign a label  $\hat{y}$  for a new sample  $x$  is to choose  $\hat{y}$  maximizing  $p(y|x)$ . According to Bayes rule,

$$\begin{aligned}\hat{y} &= \arg \max_y p(y, x) \\ &= \arg \max_y \frac{p(x|y)p(y)}{p(x)} \\ &= \arg \max_y p(x|y)p(y) \\ &= \arg \max_y p(y|x).\end{aligned}$$

At first glance, what can be better than creating a model that is complete, i.e., a generative model? The most intuitive explanation why generative classifiers are excessive and can be outperformed by less elaborated models has been provided by Vapnik (1998): “one should solve the problem directly and never solve a more general problem as an intermediate step”. The generative models provide a generator of data but if the goal is classification, modeling the data generator is an intermediate and often more complex problem.

Generative models are known to obtain the correct posterior if the training data are drawn according to the true distribution. We know that, in real-world applications, the true distribution is unknown, and it is therefore unnecessary to construct the full underlying distribution if  $p(y|x)$ , the distribution we need, can be modeled directly.

### 2.1.2 DISCRIMINATIVE MODELS

The discriminative approach avoids modeling the underlying distribution and is aimed at directly mapping the observations into labels. Discriminative models design directly  $p(y|x)$ . Among the efficient probabilistic discriminative models, one can mention logistic regression and its generalizations, such as maximum entropy Markov models and conditional random fields. The discriminative learning paradigm does not include marginal probability neither of labels, nor of observations (see Section 4.1.2 for details). Discriminative models, in their turn, have some disadvantages. Discriminative approach is focused on construction of classification boundaries as it is mentioned in (Tu, 2007), and to construct them one needs both positive and negative examples, whereas negative examples are not always available.

Given a training data set, a parametric family of probability models can either fit a joint likelihood  $p(y, x)$  and result in a generative classifier, or conditional likelihood  $p(y|x)$ , and result in a conditional classifier. Such generative and conditional classifiers are called in (Ng and Jordan, 2002) generative-discriminative pairs. To provide deeper insights in generative and discriminative learning, let us consider two such pairs, the logistic regression and naive Bayes models on the one hand and hidden Markov model and maximum entropy Markov model on the other hand.

## 2.2 LOGISTIC REGRESSION AND NAIVE BAYES FOR CLASSIFICATION TASKS

---

In this section, we investigate a generative-discriminative pair of classification methods widely used for supervised classification, logistic regression and naive Bayes.

We consider the distributions modeled by logistic regression and by naive Bayes. We will see that the logistic regression directly models the conditional probability  $p(y|x)$  and that the estimates  $p(x|y)$  and  $p(y)$  designed by the naive Bayes can be used to predict a class given an observation.

We describe the maximum likelihood approaches used to estimate the parameters in these models, and we discuss whether the logistic regression and the naive Bayes are appropriate for all kinds of applications.

### 2.2.1 LOGISTIC REGRESSION

The logistic regression model aims to predict the posterior probability of a class  $y$ ,  $y \in \{1, \dots, K\}$  via linear functions of an observation  $x$ ,  $x \in \mathbb{R}^d$ . The model is based on the assumption that the conditional probability of a class, given an observation, is proportional to  $\exp(f_k(x))$ , with  $f_k(x) = \theta_k^T x$ , where  $k$  is a class  $k \in K$ , and  $\theta_k$  is a vector of parameters

associated with the class  $k$ . The functions

$$\begin{aligned}
 f_1(x) &= \log \frac{g(y=1|x)}{g(y=K|x)} = \theta_1^T x & \text{with} & & g(y=1|x) &= \frac{\exp(f_1(x))}{\sum_{j=1}^{K-1} \exp(f_j(x)) + 1} \\
 f_2(x) &= \log \frac{g(y=2|x)}{g(y=K|x)} = \theta_2^T x & \text{with} & & g(y=2|x) &= \frac{\exp(f_2(x))}{\sum_{j=1}^{K-1} \exp(f_j(x)) + 1} \\
 & & & \dots & & \\
 f_K(x) &= \log \frac{g(y=K|x)}{g(y=K|x)} = \theta_K^T x = 0 & \text{with} & & g(y=K|x) &= \frac{1}{\sum_{j=1}^{K-1} \exp(f_j(x)) + 1}
 \end{aligned}$$

are called logit transformations. Therefore, the multiclass or polytomous logistic regression is specified in terms of  $K - 1$  logit functions. Here, we used  $g(y = K|x)$  as denominator, however, this choice is arbitrary. Such a parameterization makes the model identifiable.

The estimation of the parameter vector  $\theta$  both, for the binary and polytomous logistic regressions is performed using maximum log-likelihood. Penalization methods, e.g., the  $L_2$  norm penalty term, aim to avoid overfitting of a model penalizing large fluctuations of the parameters to be estimated. The negated conditional log-likelihood of  $N$  observations  $X = (x_1, \dots, x_N)$  and their labels  $Y = (y_1, \dots, y_N)$ , penalized by the  $L_2$  penalty term, is defined as

$$\ell(Y|X; \theta) = - \sum_{i=1}^N \left\{ \sum_{k=1}^K \mathbb{1}\{y_i = k\} \theta_k^T x_i - \log \sum_{k=1}^K \exp \theta_k^T x_i \right\} - \frac{\rho}{2} \sum_{j=1}^{d \cdot (K-1)} \theta_j^2, \quad (2.1)$$

where  $\rho$  is a parameter to be adjusted, for instance, by cross-validation. Note that  $\theta_K = 0$ .

For a binary logistic regression,  $y \in \{0, 1\}$ , the conditional log-likelihood takes the form:

$$\ell(Y|X; \theta) = - \sum_{i=1}^N \left( y_i \theta^T x_i - \log (1 + \exp(\theta^T x_i)) \right) - \frac{\rho}{2} \sum_{j=1}^d \theta_j^2. \quad (2.2)$$

The logistic regression criterion is convex, since its Hessian matrix (the matrix of the second derivatives) is positive semi-definite. The logistic regression criterion, penalized with the  $L_2$  norm, is strictly convex, since the Hessian is guaranteed to be positive definite.

If the dimensionality of the problem  $d$  is reasonably small, so that it is feasible to store and to recompute the matrix of the second derivatives at every iteration, the Newton-Raphson method can be used to solve the problem. We use the Newton-Raphson method, considered below to estimate the parameters of logistic regression in Chapter 5.

## Newton-Raphson

The Newton-Raphson method (Nocedal and Wright, 2006) is a numerical method to minimize a function  $g(\theta)$ . At  $\theta = \tilde{\theta}$ ,  $g(\theta)$  can be approximated by the quadratic Taylor expansion of  $g(\theta)$ :

$$g(\theta) \approx g(\tilde{\theta}) + \nabla g(\tilde{\theta})^T (\theta - \tilde{\theta}) + \frac{1}{2} (\theta - \tilde{\theta})^T H(\tilde{\theta}) (\theta - \tilde{\theta}), \quad (2.3)$$

where  $\nabla g(\tilde{\theta})$  is the gradient of  $g(\tilde{\theta})$ , and  $H(\tilde{\theta})$  is the Hessian of  $g(\tilde{\theta})$ ; the approximation of  $g(\theta)$  is a quadratic function, which is minimized by solving

$$\nabla g(\tilde{\theta}) + H(\tilde{\theta})(\theta - \tilde{\theta}) = 0$$

this yields the Newton-Raphson step:

$$\theta = \tilde{\theta} - H(\tilde{\theta})^{-1} \nabla g(\tilde{\theta}). \quad (2.4)$$

The speed of convergence of  $\tilde{\theta}_t$  to the optimum of equation (2.1) is quadratic in the number of iterations  $t$ .

Since we apply the Newton method to the criteria of binary and polytomous logistic regressions, let us consider the expressions of the gradient and Hessian for the binary and polytomous logistic regressions. The first derivative of the negated unpenalized binary logistic regression log-likelihood function is

$$\begin{aligned} \nabla_{\theta} \ell(Y|X; \theta) &= - \sum_{i=1}^N \left( y_i - \frac{1}{1 + \exp(-\theta^T x_i)} \right) x_i \\ &= - \sum_{i=1}^N x_i (y_i - g(y = 1|x_i)). \end{aligned}$$

The second derivative of the binary negated unpenalized logistic regression log-likelihood function is

$$\begin{aligned} \nabla_{\theta}^2 \ell(Y|X; \theta) &= \sum_{i=1}^N x_i x_i^T g(y = 1|x_i)(1 - g(y = 1|x_i)) \\ &= \sum_{i=1}^N x_i x_i^T w(x_i), \end{aligned}$$

where  $w(x_i) = g(y = 1|x_i)(1 - g(y = 1|x_i))$ .

Therefore, the Newton-Raphson update for binary logistic regression taking the  $L_2$  penalty term into account is

$$\theta = \tilde{\theta} + \left( \sum_{i=1}^N x_i x_i^T w(x_i) + \rho \right)^{-1} \left( \sum_{i=1}^N x_i (y_i - g(y = 1|x_i)) - \rho \tilde{\theta} \right). \quad (2.5)$$

The expression for the negated unpenalized polytomous logistic regression gradient is

$$\nabla_{\theta_k} \ell(Y|X; \theta) = - \sum_{i=1}^N \sum_{k=1}^K (\mathbb{1}\{y_i = k\} - g(y = k|x_i)) x_i. \quad (2.6)$$

The expression for the Hessian is the following:

$$\nabla_{\theta_k \theta_l^T}^2 \ell(Y|X; \theta) = \begin{cases} \sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^K -g(y = k|x_i)g(y = l|x_i)x_i x_i^T & \text{if } k \neq l, \\ \sum_{i=1}^N \sum_{k=1}^K g(y = k|x_i)(1 - g(y = k|x_i))x_i x_i^T & \text{if } k = l. \end{cases}$$

The quadratic convergence of the Newton-Raphson method is its strong advantage. The necessity to compute the Hessian matrix at its each iteration makes the method not applicable for large dimensional problems. The solutions for large dimensional tasks are considered in Chapter 3.

## 2.2.2 NAIVE BAYES CLASSIFIER

To classify a test example  $x$ , naive Bayes chooses a label  $\hat{y}$  that maximizes the joint probability

$$\hat{y} = \arg \max_{y \in Y} p(x, y). \quad (2.7)$$

The applications we consider in the thesis are discrete data problems. The naive Bayes classifier modeling

$$p(x|y)p(y) \quad (2.8)$$

for discrete data uses counts to compute the maximum likelihood estimates  $p(x|y)$  and  $p(y)$ :

$$p(x|y) = \frac{\sum_{i=1}^N \mathbb{1}\{X_i = x, Y_i = y\}}{\sum_{i=1}^N \mathbb{1}\{Y_i = y\}}, \quad (2.9)$$

$$p(y) = \frac{\sum_{i=1}^N \mathbb{1}\{Y_i = y\}}{N}. \quad (2.10)$$

In the models where observations are  $d$  dimensional vectors, the naive Bayes model assumes that, given a class  $y$ , the components  $x_j$  of the observation  $x$  are independent:

$$p(y, x) = p(y) \prod_{j=1}^d p(x_j|y). \quad (2.11)$$

The logistic regression and the naive Bayes technique have been widely used in the machine learning community due to their simplicity and effectiveness. However, both models are not suited for applications where elaborated dependencies are to be taken into account.

We now consider graphical models which are a framework to visualize dependencies and structure, as well as to provide insights into training and inference of probabilistic models.

## 2.3 GRAPHICAL MODELS

---

Graphical models are a natural formalism to describe a structure and dependencies in a probabilistic model, since statistical models can be formulated in terms of graphs. The research on graphical models is very active nowadays, see, e.g., (Lauritzen, 1996, Jordan et al., 1999, Jordan, 1999, Wainwright and Jordan, 2003).

To provide some intuition, let us consider the logistic regression and the naive Bayes as graphical models. Logistic regression and naive Bayes models can be represented graphically, as it is done on the Figure 2.1:  $x = [x_1, x_2, x_3]$  is a vector of observations, and  $y$  its corresponding label. According to the standard notations, each node is a variable. Shadowed nodes are considered to be observed, and the transparent ones are hidden.

An arrow defines conditional dependence of an observation given a label, e.g.,  $p(x_1|y)$ ,  $p(x_2|y)$ ,  $p(x_3|y)$  on the Figure 2.1 on the left. As it was already mentioned, the idea of modeling  $p(y|x)$  directly is connected with discriminative models, e.g., with the logistic regression, drafted on the same figure on the right. The connection lines define conditional dependency of a label given all nodes that are included in a clique.

The naive Bayes model assumes that given a class  $y$ , the components  $x_j$  of the observation  $x$  are independent, as it is modeled by equation (2.11), and as it is shown on the left of Figure 2.1. The models assumption consists in conditional independence of all attributes.

In natural language processing application of logistic regression results in an approach called “bag of words” (Lewis, 1998), what means that the word order is completely ignored. It is a drastic simplification and can result in bad performance.



Figure 2.1: Graphical representations of the generative model naive Bayes (left) and discriminative model logistic regression (right).

It is not efficient to model sequential structure dependencies, i.e. dependencies between labels neither with the logistic regression, nor with the naive Bayes. Let us consider more complicated graphs that include sequential dependencies.

In the machine learning community, the graphical models are used to visualize the structure and the dependencies of the underlying distribution. All probabilistic labeling methods we investigate in this thesis are associated with some graphical structure, which is considered to be a part of their definitions. The graphical models are divided into directed and undirected models. Both directed and undirected graphs can represent either a generative or a discriminative underlying distribution. As we will see, it is the normalizing factor that determines whether the distribution is conditional or joint.

The semantics of directed and undirected models is different and their distinction lies in their factorization. Directed models are factorized as a product of local probability functions, i.e. functions which are related to local marginal distributions. Undirected models, on the contrary, factorize as a normalized product over all cliques of a graph, where the functions associated with the cliques do not have any probabilistic interpretation.

The concept of conditional independence is fundamental for graphical models.

**Definition 2.1.** *The nodes  $x_A$  and  $x_B$  are independent ( $x_A \perp x_B$ ) if  $p(x_A, x_B) = p(x_A)p(x_B)$ .*

**Definition 2.2.** *The nodes  $x_A$  and  $x_B$  are called conditionally independent given  $x_C$  ( $x_A \perp x_B | x_C$ ) if  $p(x_A, x_B | x_C) = p(x_A | x_C)p(x_B | x_C)$  or  $p(x_A | x_C, x_B) = p(x_A | x_C)$ .*

Hence, missing variables in the local conditional probability functions correspond to missing edges in the associated graph.

### 2.3.1 DIRECTED GRAPHS

Let  $G = (V, E)$  be a directed graph and  $\{\phi(x_i, x_{\pi_i}) : i \in V\}$  be a set of functions, where  $x_{\pi_i}$  are parents of  $x_i$ , then we can write a joint probability distribution

$$p(x_1, \dots, x_T) = \prod_{t=1}^T \phi_k(x_t, x_{\pi_t}), \quad (2.12)$$

where  $T$  is the length of a sequence, and  $\phi_k(x_t, x_{\pi_t})$  is a marginal probability of  $x_t$  given its parents. The choice of  $\{\phi_k(x_t, x_{\pi_t})\}_{t=1}^T$  defines the joint probability distribution that belongs to the family of joint probability distributions associated with a specific  $G$ .

Taking all the preceding context of  $x_t$  into consideration, and defining  $\phi_k(x_t, x_{\pi_t}) = p(x_t | x_1, \dots, x_{t-1})$ , we model

$$p(x_1, \dots, x_T) = \prod_{t=1}^T p(x_t | x_1, \dots, x_{t-1}). \quad (2.13)$$

An example of such a distribution is shown on Figure 2.2, and the factorization for this graph in particular is as follows:

$$p(x_1, x_2, x_3, x_4, x_5, x_6) = p(x_1) p(x_2 | x_1) p(x_3 | x_1, x_2) p(x_4 | x_1, x_2, x_3, x_4) \\ p(x_5 | x_1, x_2, x_3, x_4) p(x_6 | x_1, x_2, x_3, x_4, x_5).$$

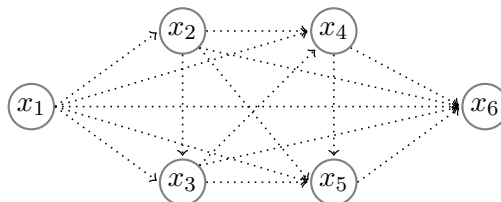


Figure 2.2: Example of a (rather complex) directed graphical model

An example of the graphical model, which is less complex than one presented in equation 2.13, is drafted in Figure 2.3, and the graph is factorized as

$$p(x_1, x_2, x_3, x_4, x_5, x_6) = p(x_1) p(x_2 | x_1) p(x_3 | x_1) p(x_4 | x_2) p(x_5 | x_3) p(x_6 | x_2, x_3).$$

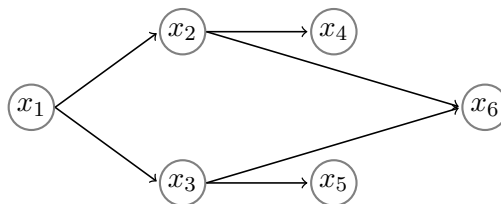


Figure 2.3: Example of a directed graphical model representing certain conditional independence assumptions

### 2.3.2 UNDIRECTED MODELS

Let  $G = (V, E)$  be an undirected graph, and  $V = X \cup Y$ ,  $\psi_c$  be a local function (or compatibility function), associated with a clique  $c \in C$ , where  $C$  is a set of all cliques;  $x$  is an assignment to  $X$ ,  $y$  to  $Y$ ;  $x_c$  is an assignment to a set  $c \in C \subset X$ ,  $y_c$  to a set  $c \in C \subset Y$ .

An example of undirected generative model are Markov random fields, defined in (Kendall and Snell, 1980) as

$$p(y, x) = \frac{1}{Z} \prod_{c \in C} \psi_c(y_c, x_c), \quad (2.14)$$

where  $Z = \sum_{y \in Y} \sum_{x \in X} \prod_{c \in C} \psi_c(y_c, x_c)$  is a normalization forcing the probability distribution to sum to one.

Note that, with the normalization factor  $Z = \sum_{y \in Y} \prod_{c \in C} \psi_c(y_c, x_c)$ , we get a discriminative model, that corresponds to an underlying conditional probability  $p(y|x)$ . Typically,

$$\psi_c(y_c, x_c) = \exp \left( \sum_{k=1}^K \theta_{ck} f_{ck}(y_c, x_c) \right), \quad (2.15)$$

where  $K$  is the number of features.

As an example of an undirected models representing certain conditional independencies, look at Figure 2.4. The nodes  $x_2$  and  $x_3$  separate  $x_1$  from  $x_4$  and from  $x_5$  respectively. Therefore, we say that the node  $x_4$  is independent from  $x_1$  given  $x_2$ , and the node  $x_5$  is independent from  $x_1$  given  $x_3$ .

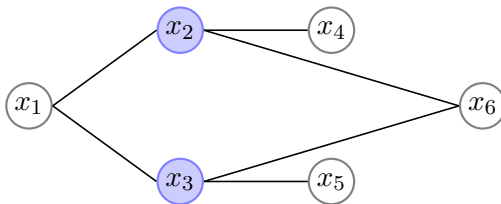


Figure 2.4: Example of an undirected graphical model, the nodes  $x_4$  and  $x_5$  are independent from  $x_1$  given  $x_2$ , and  $x_3$  respectively.

A joint or a conditional probability distribution in the context of graphical models is designed as a product of local graph functions. According to the definition of the local graph functions, there are several conditions not to be violated in the case of directed models. If we come back to equation (2.12), for example, we see that the right-hand side has to be non-negative, and the right-hand side sums to one over  $\{x_1, \dots, x_T\}$ . The properties of the exponential function guarantee that the local conditional functions are non-negative, since  $\exp(\cdot)$  is non-negative.

Now let us explore probabilistic graphical models for sequential prediction.



## 2.4 HIDDEN MARKOV MODELS AND MAXIMUM ENTROPY MARKOV MODELS

Hidden Markov model (HMM) and maximum entropy Markov model (MEMM) construct a generative-discriminative pair. HMM and MEMM achieve much better results in sequential labeling tasks than the logistic regression and naive Bayes, the generative-discriminative pair considered above.

In this section, we consider the structure modeling with HMM and MEMM, to be precise, first order Markovian type models. We investigate the dynamic programming procedure, since as we will see, the direct computation of the normalizing term is intractable.

Both hidden Markov models and maximum entropy Markov models have been successfully used for structured output prediction. Hidden Markov models have been widely applied to language structure predicting, first of all, to part of speech tagging and disambiguation tasks (DeRose, 1988, Elworthy, 1994, Kupiec, 1992). More recently, HMMs have been adapted for molecular biology problems, e.g., for gene prediction (Stanke and Waack, 2003); for bioinformatics in general see (Koski, 2001).

Maximum entropy Markov Models have been applied to various natural language processing tasks, e.g., (Blunsom, 2004) for semantic role labeling, and to protein secondary structure prediction in (Kim, 2001).

### 2.4.1 HIDDEN MARKOV MODELS

Hidden Markov models are not only a powerful approach to sequence labeling with strong theoretical foundations and good generalization performance. Inference in hidden Markov models relies on dynamic programming techniques that are also used in more recent structure prediction methods.

We start with a terminological note. Applications considered in this thesis are supervised learning tasks. In our case, in the training process, “visible” Markov models are constructed since the models we treat are mostly fully-observed, but, for the test (decoding), the labels are hidden. First order hidden Markov models are graphically sketched on the left on Figure 2.5.

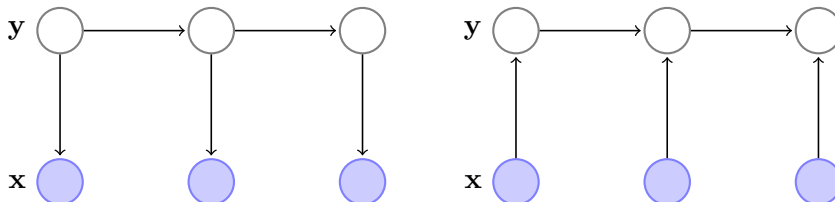


Figure 2.5: Graphical representation of hidden Markov models (left) and maximum entropy Markov models (right).

Hidden Markov models (Cappé et al., 2005), (Bilmes, 1998) have been and are one of

the most efficient generative methods appropriate for sequential data for years. HMMs construct a joint distribution over the states and observations. The states in hidden Markov models are considered to be hidden since the states (in the test procedure) are not given and have to be predicted. Here, we consider the transitions between two neighboring states, a current state depends only on its previous state, in other words, transitions follow a first order Markov process.

Formally, a hidden Markov model is defined by:

- a finite set of states  $Y$ ,
- a finite set of observations  $X$ ,
- a state transition matrix  $A$  that contains the transition probabilities  $a_{y,y'}$  from a state  $y'$  to the following one  $y$ ,
- an observation/transition matrix  $B$  containing the probability distribution  $b_{x,y}$  (the probabilities to emit  $x$  given  $y$ ),
- an initial state distribution  $q(y)$ .

Data generation under first order Markov models is described by the Algorithm 1.

---

**Algorithm 1** Sequence Generation in a First Order Markov Process

---

```

Start in state  $y_1$  with probability  $q(y_1)$ 
Emit an observation  $x_1$  with probability  $b_{x_1,y_1}$ 
for  $t = 2 \dots T$  do
  {for all positions in a sequence}
  Move from  $y_{t-1}$  to  $y_t$  with probability  $a_{y_t,y_{t-1}}$ 
  Emit an observation  $x_t$  with probability  $b_{x_t,y_t}$ 
end for

```

---

### Three Classical HMM Problems

Three classical problems of hidden Markov models and their solutions can be directly applied to maximum entropy Markov models, considered in Section 2.4.2 and to the model we study in this thesis, conditional random fields, examined in the next chapter.

There are “three classical problems” (Rabiner, 1989) concerning hidden Markov models, and correspondingly three standard solutions. To simplify notations, we let  $\theta = (A, B, q)$ .

1. Given a model, compute the probability of a sequence  $\mathbf{x}$

$$\begin{aligned}
 p(\mathbf{x}|\theta) &= \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}|\theta) = \sum_{\mathbf{y}} p(\mathbf{x}|\mathbf{y}, \theta)p(\mathbf{y}|\theta) \\
 &= \sum_{\mathbf{y}} q_{y_1} b_{x_1,y_1} a_{y_2,y_1} b_{x_2,y_2} \dots a_{y_T,y_{T-1}} b_{x_T,y_T}, \tag{2.16}
 \end{aligned}$$

where  $\mathcal{Y}$  is the set of all possible label sequences. Usually the direct computation of  $p(\mathbf{x}|\theta)$  is intractable, since it invokes  $2 \times T \times |Y|^T$  computations. At every position  $t =$

$1, \dots, T$  we have  $|Y|$  possible labels, and therefore  $|Y|^T$  possible output sequences. For every possible label sequence, one needs  $2 \times T$  computations, due to two types of parameters, state transition and observation emission probabilities.

The probability of a sequence can be computed with the forward algorithm. The joint probability of an observational subsequence  $(x_1 \dots x_t)$  reaching state  $y$  at time  $t$  is defined as

$$\alpha_t(y) = p(x_1, \dots, x_t, y_t = y | \theta).$$

The  $\alpha$ -pass or forward recursion takes the form

$$\begin{cases} \alpha_1(y) = q_y b_{x_1, y}, \forall y, \\ \alpha_t(y) = \sum_{y'} \alpha_{t-1}(y') a_{y, y'} b_{x_t, y}, \forall y, \end{cases}$$

and the probability of a sequence of observation is

$$p(\mathbf{x} | \theta) = \sum_y \alpha_T(y).$$

The complexity of the  $\alpha$ -pass is  $T \times |Y|^2$ . Why we have  $T \times |Y|^2$  computations, can be illustrated by Figure 2.6 on the left. On every position  $t$ ,  $t = 1, \dots, T$  we perform  $|Y|^2$  propagations, we propagate values from each state at a position  $t - 1$  to each state at a position  $t$ .

- Given observations and labels, train the model. The algorithm that is used to estimate parameters is the Baum-Welch algorithm (Baum et al., 1970). It makes use of a forward-backward procedure, and is a particular case of the expectation-maximization method. The conditional probability of an observation subsequence  $(x_{t+1}, \dots, x_T)$  given  $y_t = y$  is defined by

$$\beta_t(y) = p(x_{t+1}, \dots, x_T | y_t = y, \theta).$$

The  $\beta$ -pass, or backward algorithm, computes these quantities through a recursion

$$\begin{cases} \beta_T(y) = 1, \forall y, \\ \beta_t(y') = \sum_y \beta_{t+1}(y) a_{y, y'} b_{x_{t+1}, y}, \forall y', \end{cases}$$

and the marginal conditional probabilities are

$$p(y_t = y | \mathbf{x}, \theta) = \frac{\alpha_t(y) \beta_t(y)}{p(\mathbf{x} | \theta)}.$$

$$p(y_t = y', y_{t+1} = y | \mathbf{x}) = \frac{\alpha_t(y') b_{x_{t+1}, y} \beta_{t+1}(y) a_{y, y'}}{p(\mathbf{x} | \theta)}.$$

- Given a model and an observation sequence, find the optimal state sequence.

The Viterbi algorithm (Viterbi, 1967), drafted as Algorithm 2 describes the inference procedure: for every position  $t$  of a given sequence the optimal subpaths ending in all possible states  $Y$  are kept. Dynamic programming is used. The idea of it is to keep the probabilities of subpaths rather than recompute them several times. The algorithm is similar to the forward pass, except for summation is replaced by maximization. The values are stocked in tables of dimension  $|Y| \times T$ . Figure 2.6 on the left schematically visualizes the tables which stock the probabilities of subpatterns, and Figure 2.6 on the right provides an idea of the backtracking procedure used to reconstruct the best path.

---

**Algorithm 2** The Viterbi Algorithm
 

---

```

 $\delta_1(y) = q(y) \forall y$ 
for  $t = 2 \dots T$  do
  {for all positions in a sequence}
  {do Forward algorithm with max instead of summation}
   $\delta_t(y) = \max_{y'} \delta_{t-1}(y') a_{y,y'} b_{x_t,y} \forall y$ 
   $\iota_t(y) = \arg \max_{y'} \delta_{t-1}(y') a_{y,y'} b_{x_t,y} \forall y$ 
end for
{Backtracking}
 $\hat{y}_T = \arg \max_y \delta_T(y)$ 
for  $t = T - 1 : -1 : 1$  do
   $\hat{y}_t = \iota_{t+1}(\hat{y}_{t+1})$ 
end for

```

---

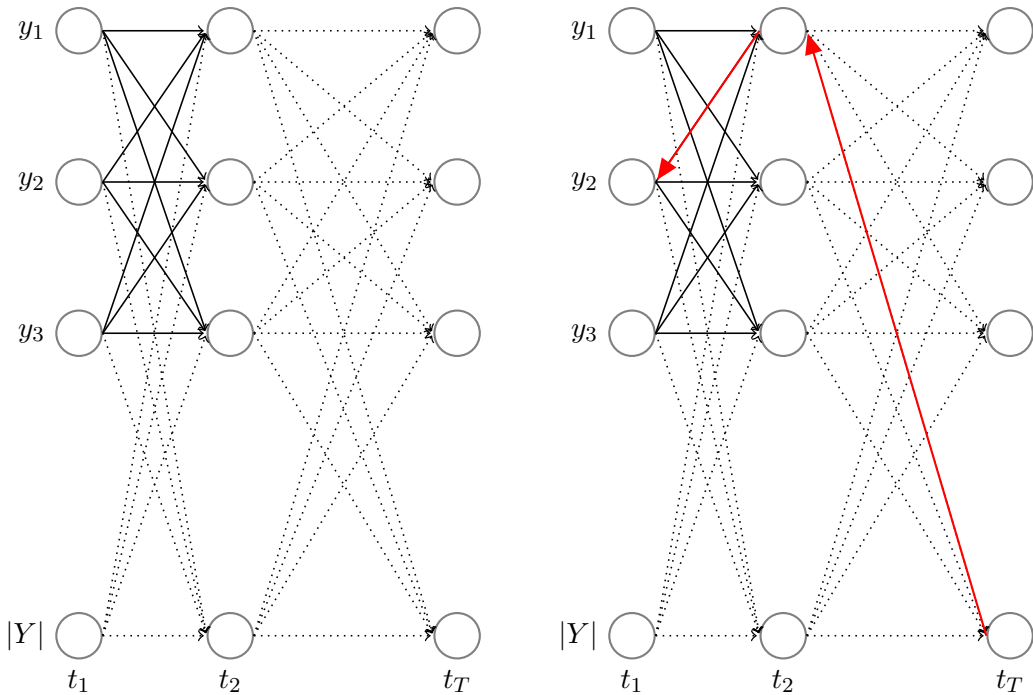


Figure 2.6: Tables. Partial results of  $\alpha$ -pass calculations (left) and backtracking procedure (right).

### 2.4.2 MAXIMUM ENTROPY MARKOV MODELS

The limitations of hidden Markov models motivated the development of the maximum entropy Markov models (MEMMs). The primary idea is to avoid modeling a joint distribution to solve a conditional problem.

The HMM transition and observation functions can be unified in a single function that stands for a weight of a current state given a previous state and a current observation. Such a reparameterization allows to take richer dependencies into account. This idea is applied to MEMMs.

The maximum entropy Markov model (McCallum et al., 2000) is a type of Markovian sequential model. It is a directed discriminative graphical model, and its graphical representation is drafted on the right of Figure 2.5. The three classical hidden Markov model problems, described above, are relevant for the maximum entropy Markov models and are solved straightforward with slightly modified forward-backward algorithm and the Viterbi algorithm.

In the MEMM model, the transition probability and emitting probability are replaced by a single dependency  $p(y|y', x)$ :

$$p(\mathbf{y}|\mathbf{x}) = p(y_1|x_1) \prod_{t=2}^T p(y_t|y_{t-1}, x_t),$$

each of these functions is represented by an exponential model

$$p(y|y', x) = \frac{1}{Z(y', x)} \exp \sum_k \theta_k f_k(y', y, x).$$

The model suffers from increased number of parameters compared to HMMs. However, splitting  $\sum_k \theta_k f_k(y', y, x)$  into  $\sum_m \theta_m f_m(y, x)$  and  $\sum_l \theta_l f_l(y', y)$  results in the number of parameters equivalent to first order HMMs. Its estimation and inference complexity is quadratic in the number of labels.

Maximum entropy Markov model made it possible to apply maximum entropy models (Berger et al., 1996, Rosenfeld, 1996) to sequence labeling tasks, however, the so-called label bias problem is usually associated with the model.

## The Label Bias Problem

Lafferty et al. (2001) mention that MEMMs, which perform the per-state normalization of probability distribution, can suffer from the so-called label bias problem. It has been also reported that the per-state normalization can lead to such a topology of a graph in which there are states with the only one outgoing state, or states with one highly probable transition (the problem of topology has been originally mentioned by Bottou (1991)).

As stated by Lafferty et al. (2001), the label bias problem reflects the situation when a previous state completely determines a next state. Klein and Manning (2002) estimated (on a POS tagging task) the parameters with the upward conditional Markov model, whose graphical representation is the same as of MEMM (see Figure 2.5 on the right) and transitions are normalized per-state. This model coincides with the MEMMs. However, Klein and Manning (2002) state that they did not observe the label bias problem but on the contrary, they noticed the observation bias problem, the situation where a current observation determines a label to be predicted ignoring a previous state.

We considered conditional random fields (see the next chapter) to be more perspective in comparison to MEMMs, since CRFs allow to model arbitrary dependencies and were reported to avoid the label bias problem. Therefore, in the context of the thesis, we did not perform experiments with any model based on the per-state normalization. Although we have never observed neither the label bias nor the observation bias problems in the experiments with CRFs, we are not convinced that CRFs is free from these explaining-away phenomena.

## 2.5 CONCLUSIONS

---

In this chapter, we provided a brief overview of statistical approaches that can be represented as graphical models. We have considered learning approaches for data without underlying structure, logistic regression and naive Bayes classifier, and for sequential data that take dependencies into account.

Probabilistic graphical models are widely used in machine learning, and are both a natural visualization of underlying probability distribution and a powerful inference framework. We considered briefly discriminative and generative learning families, including models for sequential prediction, namely hidden Markov models and maximum entropy Markov models. It was mentioned that discriminative models achieve in general a better generalizing performance than the generative ones (Ng and Jordan, 2002), however, the maximum entropy Markov model which is both, adopted for sequential data and models directly  $p(\mathbf{y}|\mathbf{x})$ , was reported to suffer from the label bias problem.

We devote the next chapter to the conditional random fields, an undirected discriminative model for structured output prediction.



---

## CHAPTER 3

# CONDITIONAL RANDOM FIELDS

---

### Contents

---

<b>3.1</b>	<b>Model Description . . . . .</b>	<b>24</b>
<b>3.2</b>	<b>Training and Decoding in Conditional Random Fields . . . . .</b>	<b>25</b>
3.2.1	Training in CRFs . . . . .	25
3.2.2	Decoding . . . . .	26
<b>3.3</b>	<b>Optimization Methods for Conditional Random Fields . . . . .</b>	<b>28</b>
3.3.1	Conjugate Gradient . . . . .	28
3.3.2	BFGS and L-BFGS . . . . .	29
3.3.3	Stochastic Gradient Descent . . . . .	31
<b>3.4</b>	<b>Applications and generalizations of CRFs . . . . .</b>	<b>32</b>
3.4.1	Application Domains of CRFs . . . . .	32
3.4.2	Generalizations and Alternative Estimation Methods . . . . .	34
<b>3.5</b>	<b>Performance of Conditional Random Fields . . . . .</b>	<b>37</b>
3.5.1	Performance of Conditional Random Fields on Nettealk Corpus . . . . .	38
3.5.2	Performance of Conditional Random Fields on CoNLL Data Sets . . . . .	41
<b>3.6</b>	<b>Conclusions . . . . .</b>	<b>45</b>

---

In the previous chapter, we considered two approaches to sequential labeling, HMMs and MEMMs. The discriminative alternative of HMM, maximum entropy Markov model, was reported to have some limitations, in particular, the label bias problem, caused by the per-state normalization.

The applications considered in this thesis are natural language tasks with sequential structure and complex dependencies. Therefore, a learning framework has to model sequential dependencies and take a rich set of features into account. Markov random fields, a generative approach, presented as equation (2.14), allow to construct arbitrary dependencies, however it is impossible to apply the dynamic programming to compute the normalizing factor. Even in a moderate size application the computation of the normalization is intractable.

Conditional random fields are a discriminative approach which models directly a conditional probability distribution  $p(\mathbf{y}|\mathbf{x})$ . In this chapter, we introduce the model and detail training and inference in linear-chain conditional random fields. We illustrate performance of conditional random fields on some standard natural language processing tasks.



### 3.1 MODEL DESCRIPTION

Conditional random fields (CRFs), introduced by Lafferty et al. (2001) and presented in details by Sutton and McCallum (2006) are based on the following discriminative probabilistic model

$$p_{\theta}(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\theta}(\mathbf{x})} \exp \left\{ \sum_{t=1}^T \sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t) \right\}, \quad (3.1)$$

where  $\mathbf{x} = (x_1, \dots, x_T)$  denotes an input sequence and  $\mathbf{y} = (y_1, \dots, y_T)$  is the output sequence, hereafter referred to as the sequence of labels;  $\{f_k\}_{1 \leq k \leq K}$  is an arbitrary set of feature functions and  $\{\theta_k\}_{1 \leq k \leq K}$  are the associated real-valued parameter values. By convention,  $y_0$  refers to a particular (always observed) label that indicates the beginning of the sequence. The CRF form considered in (3.1) is referred to as linear-chain CRF, although we stress that  $y_t$  and  $x_t$  could be composed not directly of the individual sequence tokens, but on sub-sequences (e.g., trigrams) or other localized characteristics. We will denote by  $Y, X$ , respectively, the sets in which  $y_t$  and  $x_t$  take their values. The normalization factor in (3.1) is defined by

$$Z_{\theta}(\mathbf{x}) = \sum_{\mathbf{y} \in Y^T} \exp \left\{ \sum_{t=1}^T \sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t) \right\}. \quad (3.2)$$

The graphical representation of a linear-chain conditional random fields is provided as Figure 3.1. As in the previous chapter, the shadowed nodes are observed and the transparent ones are hidden during the inference procedure.

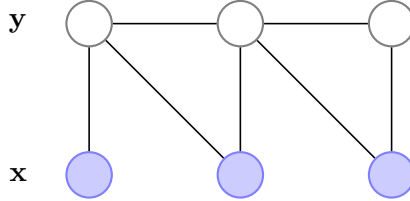


Figure 3.1: Graphical representation of linear-chain conditional random fields for a sequence of length 3.

One of possible feature choices is the combination of bigram  $\lambda_{y',y,x}$  and unigram  $\mu_{y,x}$  features:

$$\sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t) = \sum_{y', y \in Y^2, x \in X} \lambda_{y',y,x} \mathbb{1}\{y_{t-1} = y', y_t = y, x_t = x\} + \sum_{y \in Y, x \in X} \mu_{y,x} \mathbb{1}\{y_t = y, x_t = x\}, \quad (3.3)$$

where  $\mathbb{1}(\text{test}) = 1$ , if the variables are observed jointly and 0 otherwise. We can rewrite equation (3.3) as  $\mu_{y_t, x_t} + \lambda_{y_{t-1}, y_t, x_t}$ .

## 3.2 TRAINING AND DECODING IN CONDITIONAL RANDOM FIELDS

In this section, we consider the parameter estimation in CRFs. We perform minimization of the negated log-likelihood function. Training and inference in CRF are based on the forward-backward procedure, already considered for hidden Markov models in Section 2.4.1.

### 3.2.1 TRAINING IN CRFS

Given  $N$  independent labelled sequences  $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^N$ , the conditional maximum likelihood estimation is based on the minimization, with respect to  $\theta$ , of

$$\begin{aligned} \ell(\mathcal{D}; \theta) &= - \sum_{i=1}^N \log p_{\theta}(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}) \\ &= \sum_{i=1}^N \left\{ \log Z_{\theta}(\mathbf{x}^{(i)}) - \sum_{t=1}^{T_i} \sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t^{(i)}) \right\}, \end{aligned} \quad (3.4)$$

where  $T_i$  is the length of an observation  $\mathbf{x}^{(i)}$ .

Although  $\ell(\mathcal{D}; \theta)$  is a smooth convex function, it has to be optimized numerically.

The gradient of  $\ell(\mathcal{D}; \theta)$  is given by

$$\frac{\partial \ell(\theta)}{\partial \theta_k} = \sum_{i=1}^N \sum_{t=1}^{T_i} \mathbb{E}_{p_{\theta}(\mathbf{y} | \mathbf{x}^{(i)})} f_k(y_{t-1}, y_t, x_t^{(i)}) - \sum_{i=1}^N \sum_{t=1}^{T_i} f_k(y_{t-1}, y_t, x_t^{(i)}), \quad (3.5)$$

where  $\mathbb{E}_{p_{\theta}(\mathbf{y} | \mathbf{x}^{(i)})} f_k(y_{t-1}, y_t, x_t^{(i)})$  denotes the conditional expectation given the observation sequence. One can see that the gradient of the log-likelihood includes the empirical average of the global feature vector and its model expectation. It is not difficult to calculate the empirical average. The computation of the expectation in equation (3.5) implies to repeatedly compute the conditional expectation

$$\mathbb{E}_{p_{\theta}(\mathbf{y} | \mathbf{x}^{(i)})} f_k(y_{t-1}, y_t, x_t^{(i)}) = \sum_{(y', y) \in Y^2} f_k(y, y', x_t^{(i)}) p_{\theta}(y_{t-1} = y', y_t = y | \mathbf{x}^{(i)}). \quad (3.6)$$

for all input sequences  $\mathbf{x}^{(i)}$  and for all feature functions.

The solution is the same as for hidden Markov models: the forward-backward method. For every position of each training instance, we define the  $Y \times Y$  matrix

$$M_t(y_{t-1}, y_t, x_t^{(i)}) = \exp \left( \sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t^{(i)}) \right). \quad (3.7)$$

The probability of the label sequence  $\mathbf{y}$  given  $\mathbf{x}^{(i)}$  can be therefore rewritten as

$$p_{\theta}(\mathbf{y} | \mathbf{x}^{(i)}) = \frac{1}{Z_{\theta}(\mathbf{x}^{(i)})} \prod_{t=1}^{T_i} M_t(y_{t-1}, y_t, x_t^{(i)}). \quad (3.8)$$

The normalization factor  $Z_\theta(\mathbf{x}^{(i)})$  is nothing else than the sum

$$\begin{aligned} Z_\theta(\mathbf{x}^{(i)}) &= \sum_{\mathbf{y}} \prod_{t=1}^{T_i} \exp \left( \sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t^{(i)}) \right) \\ &= \sum_{\mathbf{y}} \prod_{t=1}^{T_i} M_t(y_{t-1}, y_t, x_t^{(i)}). \end{aligned}$$

The  $\alpha$  and  $\beta$  vectors are recursively defined as

$$\begin{cases} \alpha_1(y) = M_1(y', y, x_1^{(i)}), \\ \alpha_{t+1}(y) = \sum_{y'} \alpha_t(y') M_{t+1}(y', y, x_{t+1}^{(i)}), \end{cases} \quad (3.9)$$

$$\begin{cases} \beta_1(y) = \mathbf{1}, \\ \beta_t(y') = \sum_y \beta_{t+1}(y) M_{t+1}(y', y, x_{t+1}^{(i)}). \end{cases} \quad (3.10)$$

The normalization factor can be rewritten

$$Z_\theta(\mathbf{x}^{(i)}) = \sum_y \alpha_{T_i}(y). \quad (3.11)$$

The marginal probability is calculated by

$$p_\theta(y_{t-1} = y', y_t = y | \mathbf{x}_t^{(i)}) = \frac{\alpha_{t-1}(y') M_t(y', y, x_t^{(i)}) \beta_t(y)}{Z_\theta(\mathbf{x}^{(i)})}. \quad (3.12)$$

Notice that the forward and backward variables of the recursions (3.9) and (3.10) do not have any probabilistic interpretation, contrary to their analogues in hidden Markov models.

For parameter estimation, the log-likelihood is usually complemented with an additional regularization term so as to avoid overfitting, e.g. with the  $L_2$  norm. We redefine the objective functions as follows:

$$\ell(\mathcal{D}; \theta) = \ell(\mathcal{D}; \theta) + \frac{\|\theta\|^2}{2\sigma^2}.$$

The pseudo code for linear-chain conditional random fields training is presented as Algorithm 3. The complexity of the algorithm is  $T_i \times |Y|^2$ .

### 3.2.2 DECODING

The training of the CRF criterion results in an estimated vector of parameters which can be directly applied to predict a new previously unobserved sample.

Decoding in conditional random fields is done as in hidden Markov models, using Viterbi algorithm:

$$\begin{cases} \delta_1 = \theta_{y'=y_0, y=y_1, x=x_1}, \\ \delta_t(y) = \max_{y'} \{ \delta_{t-1}(y') \sum_{k=1}^K \theta_k f_k(y', y, x) \}. \end{cases} \quad (3.13)$$

---

**Algorithm 3** Training CRF

---

```
while Convergence criterion is not met do
  {Gradient and log-likelihood computations}
  for all sequences  $i$  do
    for all positions  $t = 1 \dots T_i$  do
      Update empirical average  $f_k(y_{t-1}, y_t, x_t^{(i)})$ ,  $\forall k$ 
      Compute  $\alpha_t(y)$  (eq. 3.9)
    end for
    Compute  $\ell(\theta; \mathcal{D}) + \frac{\|\theta\|^2}{2\sigma^2}$ 
    for all positions  $t = T_i \dots 1$  do
      Compute  $\beta_t(y')$  (eq. 3.10)
      Compute expectation  $p_\theta(y_{t-1}, y_t | x_t^{(i)})$  (eq. 3.12)
    end for
  end for
  Accumulate empirical average of feature vector and its model expectation for all
  sequences
  Add the penalty term  $\frac{\theta}{\sigma^2}$  to the gradient
  {Update parameter values}
  Perform an update step of numerical optimization
end while
```

---

The complexity of decoding in linear-chain CRFs is the same as for  $\alpha$ -pass, considered for HMMs in Chapter 2.

An alternative approach, symbol-by-symbol “maximum a posteriori” decoding based on the following decision rule

$$\hat{y}_t = \arg \max_{y_t \in Y} p(y_t | \mathbf{x}) \quad \forall t,$$

can achieve (Goel and Byrne, 2000) a better labeling quality than the Viterbi. However, the maximum a posteriori is more computationally expensive, since it requires to perform the Baum-Welch algorithm instead of the forward only.

Problem of Scaling

The values of  $\alpha$ s and  $\beta$ s tend to rather small values and risk to be zeroed in the case of long sequences. This problem and its solution exist for hidden Markov models, and it is possible to apply the same approach, called scaling, for conditional random fields. Each value of the  $\alpha_t$  vector is divided by the sum of  $\alpha_t$ .

Another option is to perform the forward-backward computations in the logarithmic domain, as it is described in (Sutton and McCallum, 2006) and as it is implemented in CRF++ (Kudo, 2005). The  $\alpha$ s and  $\beta$ s values are computed as follows:

$$\log \alpha_t(y) = \oplus_{y' \in \mathcal{Y}} (\log M_t(y', y, x_t^{(i)}) + \log \alpha_{t-1}(y')), \quad (3.14)$$

$$\log \beta_t(y') = \oplus_{y \in \mathcal{Y}} (\log M_{t+1}(y', y, x_{t+1}^{(i)}) + \log \beta_{t+1}(y)), \quad (3.15)$$

where the operator  $\oplus$  is defined as  $a \oplus b = \log(\exp(a) + \exp(b))$ .

### 3.3 OPTIMIZATION METHODS FOR CONDITIONAL RANDOM FIELDS

---

The conditional random fields criterion is convex and differentiable. However, the number of parameters to be estimated is usually too large and the choice of an optimization method should be done carefully. The estimation technique has to cope with large dimensionality and large data sets.

In the paper that introduces the notion of conditional random fields, in (Lafferty et al., 2001), generalized iterative scaling (Darroch and Ratcliff, 1972) and improved iterative scaling (Della Pietra et al., 1997) are used for parameter optimization. Both approaches are based on an iterative procedure

$$\theta_{i+1} = \theta_i + \delta\theta_i,$$

where  $\delta\theta_i$  is a value of update. Lafferty et al. (2001) provide details on the experiments carried out on the Penn Treebank POS tagging task. The improved iterative scaling needs about 2,000 iterations to converge. At the same time, the simple maximum Markov entropy model converges within 100 iterations. To speed the training up, the vector of parameters can be initialized with the optimized MEMMs parameter values, that results in 1,000 steps until convergence.

Iterative scaling is easy to implement and is computationally efficient for problems where the computations of a gradient and a likelihood function are expensive. It has been shown by Malouf (2002) that the performance of the scaling method is worse than one of first and second order optimization numerical methods for the maximum entropy criterion on several natural language processing problems.

A year after the introduction of CRFs, Wallach (2002) makes an attempt to extrapolate the results of (Malouf, 2002) to the conditional random fields and she shows that the conjugate gradient method is much faster. On the CoNLL 2000 data Wallach (2002) reports that the improved iterative scaling needs 150 iterations (188 seconds), the conjugate gradient method with the Fletcher-Reeves coefficient – 19 iterations (124 seconds), and the conjugate gradient with the Polack-Ribière coefficient – 27 iterations (176 seconds). Sha and Pereira (2003) reported that the iterative scaling converges slowly and never reaches the performance of the conjugate gradient method.

Below, we detail several numerical methods (Press et al., 1992) of the first and the second order that are considered to be the state-of-the art optimization approaches for conditional random fields in particular, and for log-linear models in general. All of the approaches considered below follow the general principle of gradient descent.

#### 3.3.1 CONJUGATE GRADIENT

The conjugate gradient method, introduced by Hestenes and Stiefel (1952), is an iterative procedure that is suited for numerical optimization of high dimensional problems such as CRFs training. The idea of the conjugate gradient method is to change the descent direction at every iteration in such a way that the new direction is conjugate to the previous one. A conjugate direction is a linear combination of a previous direction and of

a direction that is orthogonal to a previous one. The algorithm consists in the repetition of two main operations:

1. Compute the gradient at a point  $\theta_i$  and move in a direction that is conjugate to the previous one.
2. Perform a line search along the selected direction.

A new conjugate direction can be found in the following way:

$$d_{i+1} = d_i b_{i+1} + r_{i+1},$$

where  $d_{i+1}$  is the new direction,  $d_i$  is the previous direction,  $r_{i+1}$  is a value of the so-called residual,  $-\nabla_{\theta} \ell(\theta_i)$ , and  $b_{i+1}$  is a coefficient.

There are two common ways (Press et al., 1992) to estimate the coefficient  $b$ :

- Fletcher and Reeves method:

$$b_{i+1} = \frac{r_{i+1}^T r_{i+1}}{r_i^T r_i}. \quad (3.16)$$

- Polak and Ribière method:

$$b_{i+1} = \frac{r_{i+1}^T (r_{i+1} - r_i)}{r_i^T r_i}. \quad (3.17)$$

As mentioned above, Wallach (2002) showed that the Fletcher-Reeves method converges faster (for the CoNLL 2000 data set) than the method of Polak-Ribière.

---

**Algorithm 4** Conjugate Gradient

---

{Compute residual in  $\theta_0$ }

$d_0 = r_0 = -\nabla_{\theta} \ell(\theta_0)$

**while** Convergence criterion is not met, iterations  $i$  **do**

  {Perform line search and find  $\tau_i$  that minimizes}  $\ell(\theta_i + \tau_i d_i)$

$\theta_{i+1} = \theta_i + \tau_i d_i$

$r_{i+1} = -\nabla_{\theta} \ell(\theta_{i+1})$

  Compute  $b_{i+1}$  (Fletcher-Reeves (eq. 3.16) or Polak-Ribière (eq. 3.17))

  New search direction  $d_{i+1} = r_{i+1} + b_{i+1} d_i$

**end while**

---

### 3.3.2 BFGS AND L-BFGS

We have already presented the Newton-Raphson method and its application to the logistic regression in Section 2.2.1.

The Newton method demands to recompute the Hessian matrix and its inverse at each iteration. It is expensive to store the Hessian matrix, especially for high dimensional

applications. W. Davidon proposed to approximate the Hessian by successive gradient values. In other words, Quasi-Newton methods are based on the stored information (based on previous iterations) about the functions curvature.

The Quasi-Newton approaches decompose the Hessian matrix in such a way that it is not recomputed completely but only updated. Let  $H_i = \nabla_{\theta}^2 \ell(\theta_i)$ , then the matrix is decomposed as

$$H_{i+1} = H_i + H_i^u, \quad (3.18)$$

where  $H_i^u$  is the matrix update, and  $H_i$  is the matrix at the iteration  $i$ .

Fixing two points  $\theta_i$  and  $\theta_{i+1}$ , one can define

$$\begin{aligned} g_i &= \nabla_{\theta} \ell(\theta_i), \quad g_{i+1} = \nabla_{\theta} \ell(\theta_{i+1}), \\ p_i &= \theta_{i+1} - \theta_i, \quad q_i = g_{i+1} - g_i. \end{aligned}$$

Using the two-point difference formula of approximation

$$g_{i+1} - g_i \approx H(\theta_i)p_i, \quad (3.19)$$

which can be rewritten as

$$q_i = Hp_i, \quad (3.20)$$

from which one gets a condition that is called the Quasi-Newton condition:

$$H^{-1}q_j = p_j, \quad 0 \leq j \leq i. \quad (3.21)$$

However, it is rather the inverse of the Hessian that is used in Newton second-order methods. The same decomposition as in (3.18) can be performed with the Hessian inverse matrix as well. Let  $B = H^{-1}$ , then

$$B_{i+1} = B_i + B_i^u. \quad (3.22)$$

There does not exist a unique formula to compute the update matrix, but its general form is defined as

$$B_i^u = a\mathbf{u}\mathbf{u}^T + b\mathbf{v}\mathbf{v}^T, \quad (3.23)$$

where  $a, b$  are scalars,  $\mathbf{v}, \mathbf{u}$  are vectors.

There are two common formulas to compute the update matrix, Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS), that are based on the Quasi-Newton condition:

$$B_{i+1}q_j = p_j, \quad p_j = B_iq_j + B_i^uq_j, \quad 0 \leq j \leq i.$$

### Davidon-Fletcher-Powell (DFP) method

According to the Quasi-Newton conditions, one can write

$$p_j = B_iq_j + a\mathbf{u}\mathbf{u}^Tq_j + b\mathbf{v}\mathbf{v}^Tq_j. \quad (3.24)$$

Let us set  $\mathbf{u} = p_i$ ,  $\mathbf{v} = B_iq_i$ ,  $a\mathbf{u}^Tq_i = 1$  and  $b\mathbf{v}^Tq_i = -1$ . Hence, the Davidon-Fletcher-Powell equation has the form

$$B_{i+1} = B_i + \frac{p_i p_i^T}{p_i^T q_i} - \frac{B_i q_i q_i^T B_i}{q_i^T B_i q_i}. \quad (3.25)$$

However, most implementations of CRFs, e.g. CRF++ (Kudo, 2005), Mallet (McCallum, 2002) use the L-BFGS optimization method, introduced below, since it scales well to the large dimensional data.

### Broyden-Fletcher-Goldfarb-Shanno (BFGS) method and L-BFGS

Equations  $q_j = H_{i+1}p_j$  and  $B_{i+1}q_j = p_j$  have similar forms and the pairs  $q_j/p_j$ , as well as  $H/B$ , are interchangeable. The Broyden-Fletcher-Goldfarb-Shanno method defines the Hessian approximation as

$$H_{i+1} = H_i + \frac{q_i q_i^T}{q_i^T p_i} - \frac{H_i p_i p_i^T H_i}{p_i^T H_i p_i}. \quad (3.26)$$

Taking the inverse of the Hessian, we have

$$B_{i+1} = B_i + \frac{1 + q_i^T B_i q_i}{q_i^T p_i} \frac{p_i p_i^T}{p_i^T q_i} - \frac{p_i q_i^T B_i + B_i q_i p_i^T}{q_i^T p_i}. \quad (3.27)$$

The limited BFGS (L-BFGS) method was introduced by Nocedal (1980). The L-BFGS method is very close to the the BFGS, except for that only the  $M$  last corrections of the inverse Hessian (the differences of the variables and the differences of the gradient values) are stored in the memory. The complexity of BFGS is  $O(d^2)$ , at the same time the L-BFGS has complexity  $O(d \times M)$ , where  $d$  is the dimensionality of the problem.

The Quasi-Newton optimization procedure is presented as Algorithm 5.

---

#### Algorithm 5 Quasi-Newton Algorithm

---

Input  $\theta_0, B_0$   
**while** stopping criterion is not met, iteration  $i$  **do**  
     $S_i = -B_i g_i$   
    Perform line search to estimate  $\tau$  that minimizes  $\ell(\theta_i + \tau S_i)$   
     $\theta_{i+1} = \theta_i + \tau S_i$   
    Compute  $B_{i+1}^u$  (DFP eq. (3.25) or BFGS eq. (3.27))  
     $B_{i+1} = B_i + B_{i+1}^u$   
**end while**

---

### 3.3.3 STOCHASTIC GRADIENT DESCENT

Efficient processing of large data sets is one of the major challenges of machine learning nowadays. The stochastic gradient method (Spall, 2003, Bottou, 2004) takes one training instance per iteration, instead of considering all available points. So, if, for the batch gradient descent, the update takes a gradient (in case of a first-order method), or a gradient and a second derivative (in case of a second-order method), where the gradient is cumulated on all data  $\sum_{j=1}^N \nabla_{\theta} \ell_j(\theta_i)$ , where  $\nabla_{\theta} \ell_j(\theta_i)$  is the value of the gradient in  $\theta$  at an iteration  $i$  for an observation  $j$ , then the on-line gradient descent (first-order method) takes one instance  $j$  (one small batch) from the training set per iteration  $i$  and sets:

$$\theta_{i+1} = \theta_i + \tau_i \nabla_{\theta} \ell_j(\theta_i). \quad (3.28)$$



We use the implementation of stochastic gradient descent of Bottou (2007) in our experiments and discuss its performance in Chapter 7.

The following choice of  $\tau_i$  is a typical choice of the learning rate (see, e.g., (Collins et al., 2008)):

$$\tau_i = \frac{\tau_0}{1 + i/N}, \quad (3.29)$$

where  $\tau_0$  is a constant. However, the speed of convergence can be very poor and convergence is not guaranteed. Vishwanathan et al. (2006) proposes to accelerate the convergence of the stochastic gradient descent by choosing the step size using the second-order information.

It is necessary to mention the recent exponentiated gradient algorithm (Collins et al., 2008), that is an on-line approach with a convergence reported to be faster than ones of L-BGFS and conjugate gradient.

## 3.4 APPLICATIONS AND GENERALIZATIONS OF CRFs

---

Keeping in mind that an exhaustive state-of-the-art description of conditional random fields includes a number of applications and methods that are far away from our interests and are as well beyond of scope of the thesis, let us provide here recent ideas that refer to optimization, complexity reduction, or other original implementation issues on sequential data applications. Note that we come back to semi-supervised CRFs and sparse CRFs later, in the following chapters, and we do not discuss extensively these topics here.

### 3.4.1 APPLICATION DOMAINS OF CRFs

The number of applications of conditional random fields is very large. We briefly mention only some of them.

The primary field of applications for CRFs is sequence labeling. Usually, sequences have internal structure which is hardly detectable. Conditional random fields which can model arbitrary dependencies, can model the structure.

- *Structure of natural texts.* The initial application of CRFs, considered by Lafferty et al. (2001), concerns part-of-speech tagging (on the Penn TreeBank). First order HMMs, MEMMs, and linear-chain CRFs have been trained. The authors report that HMMs perform better than the MEMMs. CRFs in their turn outperform the HMMs. However note that there is not any drastic improvement in performance. The error rates reported are 5.69%, 6.37%, and 5.55% for HMMs, MEMMs, and CRFs respectively.

Attempts have been made to use the discriminative model to discover relations in natural texts (Culotta et al., 2006). CRFs were used to perform named-entity categorization (Watanabe et al., 2007), named-entity recognition in Wikipedia, to

introduce gazetteers in discriminative models (Smith and Osborne, 2006). Multiple variations of parsing with CRFs have been realized (see e.g., (Finkel et al., 2008)).

Conditional random fields have been applied to various languages, e.g. to Japanese to perform morphological analysis (Kudo et al., 2004), and Chinese, to detect new words and carry out segmentation (Peng et al., 2004), (Tseng et al., 2005), to Arabic to perform named-entities recognition (Benajiba and Rosso, 2008), etc.

- *Language discourse tasks.* Isotonic conditional random fields, introduced in (Mao and Lebanon, 2007) are used to predict polarity, negative or positive, of the opinions that are expressed in a text.

A sentiment is a function of words and it takes values in a finite ordered set  $(0, \leq)$ . For sentiment prediction it is important and natural to take the context of words into account. The CRFs criterion, presented as equation (3.1) is not appropriate for ordinal relations. The order can be imposed with the help of the following constraints. Let  $\mathcal{M}_1$  contain words associated with positive sentiments, and  $\mathcal{M}_2$  be associated with negative sentiments. Then imposing the constraints

$$\begin{aligned} y \leq y' &\Rightarrow \theta_{y,x} \leq \theta_{y',x}, \forall x \in \mathcal{M}_1, \\ y \leq y' &\Rightarrow \theta_{y,x} \geq \theta_{y',x}, \forall x \in \mathcal{M}_2, \end{aligned}$$

the ordinary CRF criterion can be applied to the local sentiment flow analysis.

Skip-chain CRFs (Sutton and McCallum, 2006) that model distant dependencies among labels have been applied for ranking utterances by importance of meetings (Galley, 2006). Every meeting has been analyzed given either its transcription provided by a human expert, or a result of an automatic speech recognition system.

These two applications, isotonic and skip-chain CRFs can be considered as an attempt to attack language discourse tasks.

- *Applications in molecular biology.* Conditional random fields are successfully used in molecular biology for gene prediction (Culotta et al., 2005) and segmentation of biological sequences (Liu et al., 2005).
- *Robotics.* The approach is used in robotics, for scan matching (Ramos et al., 2007), multi-agent reinforcement learning applied to the light control task (Zhang et al., 2007). Conditional random fields were tested for low-level vision (Tappen et al., 2007) and brain tumor segmentation (Lee et al., 2005).
- *Statistical machine translation* includes several but at least two phases: a) word alignment from bilingual corpus and b) inference to predict, in other words, to translate, a new text. Recently conditional random fields have been applied for word alignment for phrase-based statistical machine translation (Blunsom and Cohn, 2006). The results presented outperform the generative system GIZA++ described by Och and Ney (2003).

### 3.4.2 GENERALIZATIONS AND ALTERNATIVE ESTIMATION METHODS

Among the numerous extensions of conditional random fields we want to mention the following.

- *Bayesian conditional random fields* (BCRF) have been described by Qi et al. (2005a), and applied to diagram structure recognition by Qi et al. (2005b). The training method of BCRF is opposed to maximum likelihood training. Given the likelihood of data and the prior  $p_0(\theta)$ , the posterior distribution of the parameters

$$p(\theta|\mathbf{y}, \mathbf{x}) \propto p_0(\theta) \frac{1}{Z(\mathbf{x})} \prod_{k,t} \exp \theta_k f_k(y_{t-1}, y_t, x_t)$$

is optimized during training.

- *Semi-Markov CRFs*, that are inspired by segmentation problems, have been introduced by Sarawagi and Cohen (2004). Let us consider the example reproduced in Table 3.1. The observation is a sentence, the sequence of labels indicates for each word whether it is inside or outside an entity. The goal is to predict a segmentation. A segmentation is a triplet (start position, end position, label), e.g. in the example (2, 2, O) is decoded as “a phrase that starts at a position 2 and ends at position 2 is outside any entity”, and (8, 9, I) contains information that “words at the positions 8 and 9 form an entity”.

She	went	skiing	with	Claude	Frollo	in	Massif	Central.
↓	↓	↓	↓	↓	↓	↓	↓	↓
O	O	O	O	I	I	O	I	I
				⏟			⏟	
↓	↓	↓	↓	↓		↓	↓	
(1,1,O)	(2,2,O)	(3,3,O)	(4,4,O)	(5,6,I)		(7,7,O)	(8,9,I)	

Table 3.1: Example of named-entity segmentation

Formally, a segment  $s_t$  includes a start position  $b_t$ , an end position  $e_t$ , and a label  $y_t \in Y$ ,  $s_t = (b_t, e_t, y_t)$ . The feature functions in semi-Markov CRFs are segmentation feature functions

$$f_k(x_t, s_t) = f_k(y_{t-1}, y_t, x_t, b_t, e_t).$$

Hence, the criterion of the semi-Markov CRFs, often called semi-CRFs is defined as

$$p_\theta(\mathbf{s}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_{k,t} \theta_k f_k(s_{t-1}, s_t, x_t).$$

The distinction between the original CRFs and Semi-Markov models consists in that the latter model allows each segment  $s_t$  to last for an arbitrary number of time units. Transitions within each segment  $s_t$  are not necessarily Markovian. The approach provides flexibility in structure modeling in comparison to  $n$ -order CRFs. The  $n$ -order CRFs have a computational cost that is exponential in  $n$ ,  $O(|Y|^n)$ . The semi-Markov CRFs have the complexity that is linear in  $n$ . The power of expressiveness and the computational cost of  $n$ -order CRFs and the semi-Markov CRFs in which all segments have the same length  $n$  are the same.

- The optimization of the sequential CRFs criterion is usually used. However, since it is not easy to make the correct prediction for a whole sequence, the performance is often measured pointwise. An alternative for optimization of a sequential function is *training for maximum labelwise accuracy*. The pointwise (Altun et al., 2003) or labelwise approach (Gross et al., 2006) is supposed to optimize  $p_{\theta}(y_t|\mathbf{x})$  instead of  $p_{\theta}(\mathbf{y}|\mathbf{x})$ . The approach is theoretically attractive, since it addresses the risk minimization in a direct way. Among its significant disadvantages that prevent the proposed criterion to become as popular as the sequential function, are non-convexity and an increased time complexity.

In terms of performance, Altun et al. (2003) has reported that the sequential and pointwise criteria achieve the similar accuracy (experiments on POS tagging using the Penn TreeBank and CoNLL 2002 data set).

- Structure modeling in n-order CRFs is expensive. Pseudo-likelihood is an approximation of the likelihood function, where variables are conditioned on their neighbors (Besag, 1975). In the linear-chain CRFs it is equivalent to the per-state normalization, discussed in Chapter 2 for MEMMs. Piecewise estimation is a heuristic approach described in (Sutton and McCallum, 2005) and is equivalent to node-splitting. *Piecewise pseudo-likelihood* (Sutton and McCallum, 2007) that is a sum of local conditional probabilities is appealing for models with large cardinalities, since the piecewise pseudo-likelihood criterion conditions on fewer variables than the standard conditional random fields.

Let us consider the example of likelihood approximation by piecewise pseudo-likelihood. Figure 3.2 displays a dependency which can be modeled in CRFs. The complexity of such a model is proportional to  $|Y|^3$  and is not tractable in many applications. The node-splitting procedure of the graph can result in two cliques which are illustrated on Figure 3.3 and which lead to a model with squared complexity in the cardinality of  $Y$ .

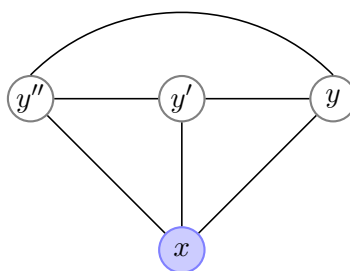


Figure 3.2: A clique modeling the dependency  $(y'', y', y, x)$ .



Figure 3.3: Example of node-splitting of the clique represented on Figure 3.2.

We investigate the node-splitting idea in our CRFs experiments in Section 3.5. We will see that splitting the expensive feature functions  $(y_t, x_{t-1}, x_t, x_{t+1})$  into combination of  $(y_t, x_{t-1})$ ,  $(y_t, x_t)$ , and  $(y_t, x_{t+1})$  does not degrade performance but drastically decreases the number of parameters.

- *Computational Savings Methods.* Many attempts have been made to reduce the complexity of the training and decoding routines. Cohn et al. (2005) proposes to represent labels as codewords in an error correcting code and to train one binary CRF for each bit of these codewords. In (Pal et al., 2006), the authors achieve significant computational savings by using beam-search during the forward-backward algorithm.
- CRFs are applied in cases where the goal is to model some structure. Since the underlying structure is not known, introduction of additional latent or observed layers can provide supplementary information. *Dynamic Conditional Random Fields* (DCRFs, (Sutton et al., 2004)) are a generalization of linear-chain conditional random fields. The DCRFs are motivated by the idea to introduce more complex interactions between labels, and to provide a possibility to perform training with multiple labels. Their graphical representation for a case of two types of labels is drafted as Figure 3.4 on the left.

As an example, let us imagine that we want to perform named-entity recognition. Our corpus contains, e.g., words, POS tags, and named entities. We know that POS tags are obtained by some tagging method (e.g., by cascading CRFs) and can be erroneous. We in our experiments (Section 3.5) consider POS tags to be additional observations. Another approach, namely DCRF, reflects the idea that POS tags are stochastic rather than observed and are considered to be supplementary labels.

Although the objective function of DCRFs is convex and L-BFGS optimization can be applied directly, the number of parameters is larger than in our experiments (Section 3.5), since there are two (or more) types of labels and the number of parameters is proportional to the sum of squared cardinalities of each type of labels. Sutton et al. (2004) propose to use approximations of the likelihood function to perform optimization of parameters.

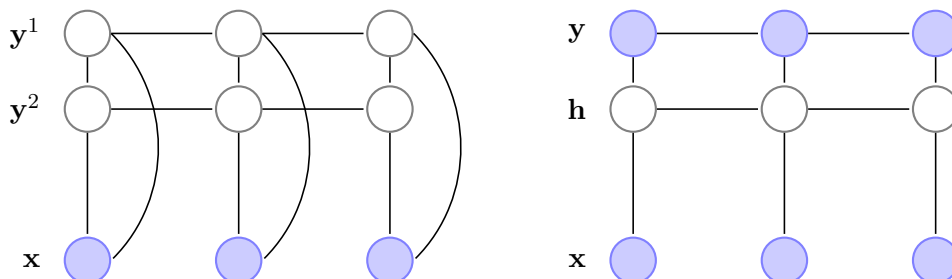


Figure 3.4: Graphical scheme of dynamic conditional random fields (left) and hidden conditional random fields (right).

- In *Hidden Conditional Random Fields* (Quattoni et al., 2007, Sung et al., 2007), intermediate hidden variables are used to model the underlying structure of the domain of observations (see graphical representation on the right of Figure 3.4). The joint distribution over the labels and hidden states given observations takes the

form

$$p(\mathbf{y}|\mathbf{x}; \theta) = \sum_{\mathbf{h}} p(\mathbf{y}, \mathbf{h}|\mathbf{x}; \theta) = \frac{\sum_{\mathbf{h}} \exp f(\mathbf{y}, \mathbf{h}, \mathbf{x}, \theta)}{\sum_{\mathbf{y}', \mathbf{h}} \exp f(\mathbf{y}', \mathbf{h}, \mathbf{x}, \theta)}.$$

The HCRF criterion is non-convex, therefore any optimization method may converge to a local optimum of the likelihood. HCRFs are useful for applications in which some intermediate structure is important and can be introduced via latent variables. In the natural language processing applications we consider in this thesis, it is not straightforward to design such a hidden layer.

- Annotation of trees is much less studied than annotation of sequences. CRFs have been generalized for *trees*, see, for instance, CRFs for XML document transformation (Jousse et al., 2006a,b), and tree-structured conditional random fields (Cohn and Blunsom, 2005). The inside-outside algorithm is used to compute the gradient.

Performance and computational efficiency of modified CRFs depend on domain of applications and on a particular task. Moreover, generalizations and alternative estimation approaches listed above are motivated by applications. So, semi-Markov CRFs is relevant for applications where labels are assigned to segments; piecewise-pseudo likelihood approach is used when the training is performed on very large CRFs; and to introduce hidden layers into CRFs as it is done in HCRFs, the knowledge of the structure of the hidden variables is required.

Although all CRFs extensions mentioned above deserve to be studied more deeply, the natural language applications we consider in the thesis do not necessarily argue in favor of such modifications. Therefore, in the next section, we apply the classical linear-chain CRFs to three corpora.

### 3.5 PERFORMANCE OF CONDITIONAL RANDOM FIELDS

---

The goal of this section is to demonstrate the potential of conditional random fields as a domain- and language-independent tool on several real world data sets. We also know that for CRFs which are able to take completely arbitrary dependencies into consideration, the modelling of dependencies is essential, both for the generalization performance and for complexity reasons. The regularisation value  $\sigma^2$  (in this section, we discuss the results of training with the  $L_2$  regularization term) is important and influences the performance, and is chosen by cross-validation.

In the following, features whose configurations are never observed during training are called negative examples. We will show that negative examples are much more numerous than the positive, i.e. observed ones, and are important to achieve a good generalization on a test data set.

There are several implementations of conditional random fields, e.g. CRFSuite (Okazaki, 2007), Sunita Sarawagi's CRF Package (Sarawagi and Cohen, 2004)<sup>1</sup>, MALLET (McCallum, 2002). Results reported in this section are obtained with CRF++<sup>2</sup> (Kudo, 2005).

---

<sup>1</sup><http://crf.sourceforge.net/>

<sup>2</sup><http://crfpp.sourceforge.net/>

Feature functions  $\{f_k\}_{k=1}^K$  take binary values, “+” is used to denote the superposition of different types of features. As mentioned above,  $(y_t, x_t)$  is a feature of a label  $y$  and an observation  $x$  at a position  $t$ ; the feature is extracted for all configurations, that is, for all labels and all observed  $x$ . If a data set contains more than one type of observations, e.g., words and part-of-speech tags and if we intend to extract the same dependencies for each observation type, our feature set takes the form of  $(y_t, x_t^1) + (y_t, x_t^2)$ , where  $x^1$  is associated with words and  $x^2$  with part of speech tags.

### 3.5.1 PERFORMANCE OF CONDITIONAL RANDOM FIELDS ON NETTALK CORPUS

In this section, we present the performance of CRFs on the Nettetalk corpus, a word phonetization task. The data contain words extracted from an English dictionary and their phonetic transcriptions. The size of the letter alphabet is 26, and the number of phonemes is 53. A brief description of the Nettetalk corpus is provided in Appendix B. We use nine parts (each part contains 1,628 instances) of the Nettetalk corpus for training, and one for testing performance.

The performance is measured in terms of the error rate on the testing set of  $N$  observations and their labels:

$$\frac{1}{N} \sum_{i=1}^N \mathbb{1}\{\hat{y}_i \neq y_i\},$$

where  $\hat{y}_i$  is a predicted label and  $y_i$  is provided by an expert.

#### Choice of Features

We carried out a number of experiments on the Nettetalk corpus, starting from naive features such as  $(y_t, x_t)$  that result in a high error rate up to elaborated redundant dependencies with millions of features and an acceptable performance.

Table 3.2 provides error rates and the number of features involved for different dependency patterns.

It is not surprising that longer and richer dependencies perform better than the simple ones. E.g.,  $(y_{t-1}, y_t, x_t)$  and the combination of  $(y_{t-1}, y_t, x_t)$  and  $(y_t, x_t)$  reach a significantly better accuracy than  $(y_t, x_t)$  alone. However, we have observed several less straightforward effects. Long patterns of observations such as  $(y_t, x_{t-2}, x_{t-1}, x_t, x_{t+1}, x_{t+2})$  do not generalize well to data and are prone to overfitting. Notice, e.g. that the combination of features which extracts about 17 millions of features overfits more than one with 3 millions of features (the last and the next to last results in Table 3.2). The redundant short dependencies play a role of smoothing. We notice that the feature  $(y_t, x_t)$  provides important smoothing.

The experiments demonstrate that longer dependencies can be modeled not only as indivisible observed patterns as the feature  $(y_t, x_{t-2}, x_{t-1}, x_t, x_{t+1}, x_{t+2})$ . A trial to split this configuration, which extracts 2 millions features, into  $(y_t, x_{t-2}) + (y_t, x_{t-1}) + (y_t, x_t) + (y_t, x_{t+1}) + (y_t, x_{t+2})$  with 7,000 parameters, results in the significant improvement in

performance (11.65% versus 20.5% on the test set). It is obvious that redundant features play an important role, and that the contribution of longer features is not possible without shorter redundant features.

Note that in the configurations with bigram features  $(y_{t-1}, y_t, x_t)$ , it is important that the implementation of CRFs takes care of the case  $t = 1$ . The CRF++ tool ignores the feature  $(y_0, y_1, x_1)$  and the error rate with the single feature  $(y_{t-1}, y_t, x_t)$  is 27.2% on the training data, and 27.8% on the test set. Taking the feature into account as a special case leads to 14.8% error on the testing set. However such situations of first (or last) positions in a sequence are not very critical. It is sufficient to add the unigram  $(y_t, x_t)$  features to reintroduce the missed dependency to the model.

### Qualitative Error Analysis

Even with a large number of features, we could not achieve a better performance than approximately 7%. We tried to analyze the errors in order to understand where they come from. Qualitative analysis of errors made on the Nettetalk data set is shown in Table 3.3 for two choices of features, although the results are rather similar. The most frequent confusions reflect the phonetic ambiguity. The most frequent error is to mix up long and short sounds, e.g., [æ] (avid - [æ v ɪ d] (long sound)) is confused with [ə] (creditable - [k r e d ɪ t ə b l] (short sound)). Another source of ambiguity are letters that are associated with several phonemes, e.g., “er” at the end of words can be pronounced differently (confusion [ə:] (adventurer - [ə d v e n t ʃ ə r ə:]) - [ə] (defender - [d i f e n d ə])).<sup>3</sup>

The errors are directly connected with the estimated parameter vectors. Let us consider (observation, label) pairs that suffer at most from the false predictions. Given a model and an observation sequence,  $score(\mathbf{y})$  is the value of log-likelihood of  $\mathbf{y}$  given  $\mathbf{x}$ . We compare the scores computed using estimated parameters of a correct sequence  $\mathbf{y}_{\text{original}}$  and a labeled sequence  $\mathbf{y}_{\text{labeled}}$ . Approximately 65% of words are predicted with errors. In our experiments, there are 1, 100 words (the whole testing set contains 1, 628 sequences) for which  $score(\mathbf{y}_{\text{labeled}}) - score(\mathbf{y}_{\text{original}}) \neq 0$ , and these words are labeled wrongly. However, the majority of erroneously labeled sequences have only one or two confusions:

more than 2 confusions  $\implies$  355 words,  
 more than 5 confusions  $\implies$  80 words,  
 more than 8 confusions  $\implies$  23 words,  
 more than 10 confusions  $\implies$  11 words.

There are only few words whose predictions are almost completely wrong. Table 3.4 lists the most problematic words (the most problematic in the sense of maximal difference  $score(\mathbf{y}_{\text{labeled}}) - score(\mathbf{y}_{\text{correct}}) > 8$ ) for structured prediction from the Nettetalk corpus with their correct labels and corresponding decoded labels.

There is not any very clear correspondence between the pattern frequency and its estimated parameter value. Performing training with a single type of features  $(y_{t-1}, y_t, x_t)$ , we get  $54 * 53 * 26 = 74, 412$  parameters, but only 1, 976 triplets are really observed in the training procedure. Therefore, a great number of dependencies are the so-called negative

<sup>3</sup>See Appendix B for the description of the phoneme set.



Feature(s)	Number of features extracted with CRF++	Train	Test
$(y_t, x_t)^*$	1,378	38.7%	38.7%
$(y_{t-1}, y_t, x_t)^*$	74,412	17.9%	18.5%
$(y_t, x_t) + (y_{t-1}, y_t, x_t)^*$	75,790	13.9%	14.8%
$(y_t, x_{t-1}, x_t, x_{t+1})$	252,408	12.5%	13.9%
$(y_t, x_{t-1}, x_t, x_{t+1}) +$ $(y_t, x_t)$	258,640	13.8%	13.9%
$(y_t, x_{t-1}, x_t, x_{t+1}) +$ $(y_{t-1}, y_t, x_t)$ $(y_t, x_t)$	331,674	10.5%	11.4%
$(y_t, y_{t-1}, x_{t-1}, x_t, x_{t+1})$	12,125,216	7.6%	9.6%
$(y_{t-1}, y_t, x_{t-1}, x_t, x_{t+1}) +$ $(y_t, x_t)$	12,976,149	7.4%	9.3%
$(y_{t-1}, y_t, x_{t-1}, x_t, x_{t+1}) +$ $(y_{t-1}, y_t, x_t)$ $(y_t, x_t)$	13,049,183	6.8%	8.5%
$(y_t, x_{t-2}, x_{t-1}, x_t, x_{t+1}, x_{t+2})$	1,916,200	3.1%	20.6%
$(y_t, x_{t-2}, x_{t-1}, x_t, x_{t+1}, x_{t+2}) +$ $(y_t, x_{t-1}, x_t, x_{t+1}) +$ $(y_t, x_t)$	2,169,960	5.3%	9.6%
$(y_t, x_{t-2}) + (y_t, x_{t-1}) + (y_t, x_t) +$ $(y_t, x_{t+1}) + (y_t, x_{t+2})$	7,072	10.7%	11.7%
$(y_t, x_{t-2}, x_{t-1}, x_t, x_{t+1}, x_{t+2}) +$ $(y_t, x_{t-2}, x_{t-1}, x_{t+1}, x_{t+2}) +$ $(y_{t-1}, y_t, x_{t-1}, x_t, x_{t+1}) +$ $(y_{t-1}, y_t, x_{t-1}, x_{t+1})$	17,318,756	3.4%	7.8%
$(y_t, x_{t-2}, x_{t-1}, x_{t+1}, x_{t+2}) +$ $(y_{t-1}, y_t, x_{t-1}, x_{t+1}) +$ $(y_t, x_t) + (y_{t-1}, y_t, x_t)$	2,984,436	4.0%	7.3%

Table 3.2: Different patterns on Nettetalk corpus, estimation carried with CRF++ except for features marked with a star (our Matlab implementation).

Confusion $\hat{y}_t - y_t$	Confusion counts	Confusion $\hat{y}_t - y_t$	Confusion counts
[æ] - [ə]	104 (max)	[æ] - [ə]	90 (max)
[ai] - [I]	84	[ai] - [I]	85
[.] - [ə]	81	[ɔ] - [ə]	69
[e] - [.]	80	[I] - [e]	61
[ɔ] - [ə]	69	[ə] - [æ]	51
[ə] - [.]	64	[.] - [ə]	44
[z] - [s]	46	[ə] - [ɔ]	43
[i] - [I]	45	[i] - [I]	42
[I] - [e]	45	[z] - [s]	39
[ɔ:] - [ə]	44	[ə] - [e]	39
[a:] - [ə]	43	[ə] - [.]	39
[ə] - [æ]	43	[ɔ:] - [ə]	38
[ə] - [ɔ]	43	[a:] - [ə]	38
[e] - [I]	41	[e] - [I]	35
[I] - [.]	39	[e] - [ə]	34
[e] - [ə]	39	[ə] - [ei]	34
[əʊ] - [ɔ]	35	[ə:] - [ə]	33
[ə:] - [ə]	35	[æ] - [ei]	32
[ə] - [ei]	35	[.] - [e]	31
[æ] - [ei]	32	[əʊ] - [ɔ]	30

Table 3.3: The most frequent confusions (IPA) made by conditional random fields on Nettetalk corpus. Left: training carried out with the feature  $(y_{t-1}, y_t, x_t)$ . Right: training carried out with the feature combination  $(y_{t-1}, y_t, x_t) + (y_t, x_t)$ .

examples that appear to be important as well. We tried to understand which features are the most and least important and whether their frequencies play any role for the parameter estimation. Considering the most frequent label transitions for each letter and comparing them to the maximal values of parameters associated with each letter, we notice that for 14 letters the most frequent state transition coincides with the maximal estimated value. The very negative values of parameters appear as a counterpart to the frequent dependencies and are configurations, where marginal frequency of a label or an observation is high but their joint frequency is not, e.g., if a value of  $x_t$  is frequently observed in the corpus but never or rare with particular value of  $y_{t-1}$  and  $y_t$ .

### 3.5.2 PERFORMANCE OF CONDITIONAL RANDOM FIELDS ON CoNLL DATA SETS

CoNLL 2000 and CoNLL 2003 data sets are connected with the analysis of structure of natural language. As we will see from the results, CRFs can achieve a reasonable performance based on a rather simple choice of features and without making any use of linguistic sources. The CoNLL 2000 challenge is devoted to the prediction of groups of words that are syntactically correlated. The goal of the CoNLL 2003 challenge is to predict named entities. Brief descriptions of the corpora are given in the Appendix C.

$\mathbf{x}$	$\mathbf{y}_{\text{correct}}$	$\mathbf{y}_{\text{corr. score}}$	$\mathbf{y}_{\text{labeled}}$	$\mathbf{y}_{\text{lab. score}}$
ofttimes	[ɔ f t t aɪ m z]	63.29	[ɔ f t I m e s]	71.76
curvature	[k ə: v ə tʃ ʊə]	71.54	[k ə: v eɪ t ə:]	80.14
meteorological	[m i: t i ə r ə l ɔ dʒ I k ʊ l]	117.27	[m e t ə r ɔ l ɔ dʒ I k ʊ l]	125.30
beautify	[b yʊ t I f aɪ]	61.47	[b i: yʊ t I f aɪ]	69.82
cabaret	[k æ b ə r eɪ]	52.45	[k ə b a: e t]	63.72
exhortation	[e k s ɔ: t eɪ f ə n]	81.69	[I g z ə t eɪ f ə n]	96.65
rarely	[r e ə l i]	28.47	[r a: ə l I]	58.68
boatswain	[b əʊ s ə n]	71.53	[b əʊ t s w eɪ n]	84.76
glowworm	[g l ə ʊ w ə: m]	66.50	[g l a u w ə: m]	76.38
chemise	[ʃ e m i: z]	58.76	[tʃ e m I s]	67.52
anywhere	[e n i w e ə]	63.21	[ə n i w ə]	71.27
cyclist	[s aɪ k l I s t]	60.26	[s I k l I s t]	68.37
magazine	[m æ g ə z i: n]	63.64	[m æ g eɪ z I n]	74.37
elaboration	[I l ə b ə r eɪ f ə n]	90.11	[I l ə b ɔ r eɪ f ə n]	99.38
austerity	[ɔ s t e r I t i]	73.31	[ə s t ə r I t I]	82.56
rye	[r aɪ]	16.32	[r i]	32.54
acre	[eɪ k ə]	26.13	[æ k r I]	34.78
fiance	[f i a n s eɪ]	37.97	[f I a n s]	55.12
aforesaid	[ə f ɔ: s eɪ d]	66.38	[ə f ɔ e s eɪ d]	77.38
regime	[r eɪ dʒ i m]	39.35	[r I dʒ i m]	58.11
because	[b i k ə z]	54.24	[b e k ə s]	63.63

Table 3.4: Words with their correct and predicted labels for that  $score(\mathbf{y}_{\text{labeled}}) - score(\mathbf{y}_{\text{correct}}) > 8$ .

Both CoNLL00 and CoNLL03 corpora are multiobservational sets. We let  $x^1$  be associated with words,  $x^2$  – with part of speech tags, and  $x^3$  – with syntactic chunks. It is necessary to say that, as any kind of data, the CoNLL 2000 and CoNLL 2003 sets have some peculiarities, but they have some common aspects. Note, e.g., that in Table 3.8 the combination of unigram features  $(y_t, x_t^1) + (y_t, x_t^2) + (y_t, x_t^3)$  performs better than the corresponding combination of bigram features  $(y_{t-1}, y_t, x_t^1) + (y_{t-1}, y_t, x_t^2) + (y_{t-1}, y_t, x_t^3)$ . Therefore, it happens that the simple dependencies generalize much better to the test data than more elaborated features. Table 3.5 demonstrates another aspect of the CoNLL data: different informational intensity of different types of observations. In other words, it may happen, as it happens for CoNLL 2000, that performance on part-of-speech tags only is better than on words only.

The performances of the CoNLL challenges are usually evaluated using precision, recall, and F-measure rather than accuracy (Davis and Goadrich, 2006). To understand the measures, let us consider a binary problem, with 2 possible labels  $\{-1, +1\}$ . As a result of a classification, one gets a confusion matrix (Table 3.6) that contains the numbers of true/false classified data with respect to the given classes  $\{-1, +1\}$ .

	True label +1	True label -1
Classified as +1	True Positive	False Positive
Classified as -1	False Negative	True Negative

Table 3.6: Classification confusion matrix

Feature	CoNLL 2000		CoNLL 2003		
	8,936 training sequences		14,987 training sequences		
	Train error	Test error	Train error	Test A error	Test B error
$(y_t, x_t = \text{word})$ (CoNLL00 – 420, 684 feat.) (CoNLL03 – 188, 992 feat.)	16.2	24.66	1.7	10.3	14.6
$(y_t, x_t = \text{POS})$ (CoNLL00 – 968 feat.) (CoNLL03 – 368 feat.)	22.6	22.7	14.1	13.5	15.2
$(y_t, x_t = \text{SCT})$ (CoNLL03 – 136 feat.)	N/A	N/A	16.6	16.7	17.9

Table 3.5: Performance of naive features

Feature	Train Error	Test Error
$(y_{t-1}, y_t, x_t^1, x_t^2)$	9.0	17.2
$(y_t, x_t^1) + (y_t, x_t^2)$	15.2	17.7
$(y_{t-1}, y_t) + (y_t, x_t^1) + (y_t, x_t^2)$	4.9	5.9
$(y_{t-1}, y_t, x_t^1) + (y_{t-1}, y_t, x_t^2)$	4.9	6.7
$(y_t, x_t^1) + (y_t, x_t^2)$	3.0	5.6
$(y_{t-1}, y_t, x_t^1) + (y_{t-1}, y_t, x_t^2)$		

Table 3.7: Performance on CoNLL 2000, CRF++ ( $\sigma^2 = 1$ )

$$\text{Recall} = \frac{TP}{TP + FN}, \quad \text{Precision} = \frac{TP}{TP + FP}, \quad F = \frac{2 \text{ Precision Recall}}{\text{Recall} + \text{Precision}}, \quad (3.30)$$

where  $TP$  are true positives,  $FN$  – false negatives, and  $FP$  – false positives.

### Performance on CoNLL 2000

For the CoNLL 2000 task (text chunking, problem described in Appendix C), 211,727 observation/label pairs are used for training, and 47,377 pairs for testing. The baseline<sup>4</sup> proposed by the challenge is obtained by predicting the chunk tag based on its joint frequency with a corresponding part-of-speech tag. The baseline result is the following: precision – 72.58%, recall – 82.14%, F – 77.07. The best result reported on the data set is by Zhang et al. (2001): precision – 94.29%, recall – 94.01%, F – 94.13. The results of the conditional random fields experiments with the feature combination  $(y_t, x_t^1) + (y_{t-1}, y_t, x_t^1) + (y_t, x_t^2) + (y_{t-1}, y_t, x_t^2)$  reaches accuracy of 93.24%, precision of 89.65%, recall – 89.26%, and F-measure – 89.46. The number of parameters is 9, 266, 269.

Table 3.7 displays the performance for different models. We notice that it is preferable both computationally and in terms of generalization to the test set to split the dependencies. The combination of unigram features with a simple bigram bias term  $(y_{t-1}, y_t) + (y_t, x_t^1) + (y_t, x_t^2)$  has less parameters and performs better than  $(y_{t-1}, y_t, x_t^1) + (y_{t-1}, y_t, x_t^2)$ .

<sup>4</sup><http://www.cnts.ua.ac.be/conll2000/chunking/>

Feature	Train Error	Test Error	
		Test A	Test B
$(y_{t-1}, y_t, x_t^1, x_t^2, x_t^3)$	1.4	8.2	12.9
$(y_t, x_t^1) + (y_t, x_t^2) + (y_t, x_t^3)$	1.6	3.9	5.8
$(y_t, x_t^1) + (y_t, x_t^2) + (y_t, x_t^3) + (y_{t-1}, y_t)$	0.5	3.6	5.4
$(y_{t-1}, y_t, x_t^1) + (y_{t-1}, y_t, x_t^2) + (y_{t-1}, y_t, x_t^3) + (y_{t-1}, y_t)$	1.4	4.8	7.5
$(y_t, x_t^1) + (y_{t-1}, y_t, x_t^1)$ $(y_t, x_t^2) + (y_{t-1}, y_t, x_t^2)$ $(y_t, x_t^3) + (y_{t-1}, y_t, x_t^3)$	0.1	3.0	5.2

Table 3.8: Performance on CoNLL 2003, CRF++ ( $\sigma^2 = 50$ )

### Performance on CoNLL 2003

The CoNLL 2003 challenge is a named entity recognition task, briefly described in Appendix C. The baseline<sup>5</sup> for CoNLL 2003 is obtained by labelling unambiguous named entities observed in the training set. The baseline values of performance are as follows: precision – 71.91%, recall – 50.90%, and F –  $59.61 \pm 1.2$ . The best result of the challenge on the data set is presented in (Florian et al., 2003): precision – 88.99%, recall – 88.54%, and F –  $88.76 \pm 0.7$ . The performance of conditional random fields with the feature combination  $(y_t, x_t^1) + (y_{t-1}, y_t, x_t^1) + (y_t, x_t^2) + (y_{t-1}, y_t, x_t^2) + (y_t, x_t^3) + (y_{t-1}, y_t, x_t^3)$  is the following: accuracy – 96.96%, precision – 85.42%, recall – 80.78, and F-measure – 83.04.

A number of linguistic but language independent features are often used (see e.g., (Carreras et al., 2002, Carreras and Màrquez, 2003)), among them forms of words, binary flags with respect to whether a word is capitalized, whether only the first letter is capitalized. The nature of characters is taken also often into consideration (digits, alphanumeric, roman-number, punctuation, single-character patterns, etc.). The so-called predefined classes of items can provide additional information: whether a current word is a functional word, whether it is an URL, etc. Prefixes and suffixes (up to 3 – 4 characters) are often extracted from observed words, as well as flags indicating start/end of a word/phrase/sentence. An additional feature (used e.g. by Zhang and Johnson (2003)) is to convert all words either to lower-case, or to upper-case, not to loose the information that the same word can either start a sentence or be situated on a position  $t$ .

Conditional random fields with interdependent numerous features have been applied by McCallum and Li (2003) to the CoNLL 2003 English data. 8 lexicons entered by hand, such as days and months, 15 lexicons obtained from web sites (countries, publicly-traded companies, surnames, stop-words, and universities), and 25 lexicons obtained by WebListing (including people names, organizations, NGOs, and nationalities) have been used. The results reported by McCallum and Li (2003) on test B: precision – 84.52%, recall – 83.55%, F – 84.04.

<sup>5</sup><http://www.cnts.ua.ac.be/conll2003/ner/>

## 3.6 CONCLUSIONS

---

In this chapter, we introduced the conditional random field model. Conditional random fields and their extensions are among the state-of-the-art approaches for structured prediction tasks. Conditional random fields allow to model arbitrary dependencies. At the same time, even for the linear-chain conditional random fields the complexity is quadratic in the number of labels. The problems of feature selection arise from the capability of CRFs to model arbitrary dependencies. We consider the problem of feature selection in Chapters 6 and 7.

We tested the CRFs performance on standard data sets. It is important to underline that conditional random fields are a powerful learning approach. One achieves a baseline performance even with a modest feature choice, and with a more elaborated feature set CRFs are close to state-of-the-art performance without inducing knowledge about the domain of an application.



## Part II

# Semi-Supervised Discriminative Estimator





---

## CHAPTER 4

# SEMI-SUPERVISED LEARNING OF DISCRIMINATIVE MODELS

---

### Contents

---

<b>4.1</b>	<b>Brief Overview of Semi-Supervised Learning Methods . . . . .</b>	<b>50</b>
4.1.1	Four Assumptions Proposed for Semi-Supervised Learning . . . . .	52
4.1.2	Categories of Algorithms . . . . .	53
4.1.3	Semi-Supervised CRFs . . . . .	57
<b>4.2</b>	<b>Marginal Probability in Discriminative Models . . . . .</b>	<b>59</b>
4.2.1	Asymptotically Optimal Semi-Supervised Estimation . . . . .	60
4.2.2	Covariate Shift . . . . .	65
<b>4.3</b>	<b>Application to Binary Logistic Regression . . . . .</b>	<b>67</b>
<b>4.4</b>	<b>Conclusions . . . . .</b>	<b>68</b>

---

In most real-world pattern classification problems (e.g., for text, image or audio data), unannotated data are plentiful and can be collected at almost no cost, whereas labeled data are comparatively rarer, and more costly to gather. Semi-supervised learning has drawn attention of the machine learning community, since it is a sensible question to find ways to exploit the unlabeled data in order to improve the performance of supervised training procedures.

We consider semi-supervised probabilistic classifiers, in which observations and their labels are modeled as random variables. Semi-supervised approaches can be applied to real-world classification tasks which either do or do not take underlying structure into consideration. In previous chapters we discussed the advantages of the discriminative models over generative ones. Following the conclusions we made, our goal is to make use of unlabeled data in discriminative models. As we will see, it is not so straightforward as for generative models.

We define the problem and consider the framework of semi-supervised learning in general and in discriminative models in particular. We propose, in Section 4.2 a semi-supervised estimator that is shown to be asymptotically optimal. Experiments on synthetic and real world data will be reported in Chapter 5.

## 4.1 BRIEF OVERVIEW OF SEMI-SUPERVISED LEARNING METHODS

---

In this section, we discuss the attempts that have been made to combine the supervised and unsupervised learning, as well as we provide the formalization of various semi-supervised learning approaches.

Given  $X = (x_1, \dots, x_n)$  which are  $n$  i.i.d. (independent identically distributed) points ( $x_i \in \mathcal{X}$ ), unsupervised learning aims to reveal the structure of data, to find some similarities between data points and to split the input into several subsets, usually called clusters.

Supervised learning is based on the observed i.i.d. pairs (observation, label),  $(x_i, y_i)$ , and the goal is to find a mapping from  $X$  to  $Y$ . Usually, one supposes that the pairs  $(x_i, y_i)$  are sampled from a probability distribution on  $\mathcal{X} \times \mathcal{Y}$ , as described in Chapter 2. There are two main families of algorithms for supervised learning: generative and discriminative.

Semi-supervised learning assumes that there are some labeled data, and some unlabeled data. So, if we have  $X_l = (x_1, \dots, x_l)$ ,  $Y_l = (y_1, \dots, y_l)$ , observations with their labels and  $X_u = (x_{l+1}, \dots, x_{l+u})$ , some unlabeled observations, the problem is called semi-supervised learning. We refer to  $\mathcal{D}_u = X_u$  as the unlabeled data, and to  $\mathcal{D}_l = (X_l, Y_l)$  as the labeled data. The problem of semi-supervised learning was taken into consideration later than the supervised and unsupervised learning frameworks, and the first formulation of the semi-supervised problem as it is now accepted is made by Merz et al. (1992).

The reason why the semi-supervised approach became topical is that labeled data are expensive and limited. Whereas unlabeled data are cheap and plentiful. The supervised algorithms achieve a satisfactory performance, but their performance depends directly on the amount of input data. Could it be possible to improve the performance somehow with the unlabeled data? Although intuitively it should be so, some negative results are reported (Cohen et al., 2004), and it is still an open question, when and why unlabeled data are useful, and when their introduction into a model degrades performance.

There exists several forms of semi-supervised learning. In most cases it is considered as a supervised learning with additional information. Semi-supervised learning can be also considered as unsupervised learning guided by constraints whether unlabeled data points should have or should not have these of those labels. Recently, (Daumé III, 2009) proposed to introduce a differentiation between semi-supervised and semi-unsupervised learning. Usually we are in one of two following situations. We either have a lot of unlabeled instances and we hope to improve the performance by introducing some labeled points, or on the contrary, we perform the supervised learning and try to make use of unlabeled data. Semi-unsupervised learning should be based on a lot of unlabeled and little labeled points, semi-supervised learning should take little unlabeled and a lot of labeled data. It is suggested that for the case of numerous unlabeled data, i.e. for a semi-unsupervised learning, it is more natural to use a generative model, and for the case of little unlabeled data, i.e. for semi-supervised learning, to apply a discriminative model. In this thesis, we do not follow the terminology proposed by Daumé III (2009), and we refer to all methods which make use of unlabeled data as the semi-supervised approaches. However, we focus on the second scenario where there are plenty of unlabeled examples which are used (hopefully) to improve the performance of supervised classification.

The semi-supervised setting reflects in some sense the real-world better than the supervised and unsupervised approaches. Some data points are already classified, and for the rest it is up to us to deduce where observations belong to. Many proposals have been made in the recent years to devise effective semi-supervised training schemes (see (Chapelle et al., 2006) for an up-to-date panorama). Zhu (2005b) mentions five semi-supervised methods which are used more often than others: expectation-maximization with generative models, self-training, co-training, transductive support vector machines, and graph-based methods. We consider expectation-maximization applied to generative models and provide some intuition on graph-based methods in Section 4.1.2. Below we describe transductive learning, self-training, and co-training which are application independent state-of-the art semi-supervised approaches.

## Transduction Versus Semi-Supervised Learning

The idea of transductive learning is to transfer the information from labeled instances to testing points directly. Transductive learning is considered to be a simpler task than the inductive learning which consists in finding the dependencies, i.e. a function, between observations and labels. An exciting discussion ((Chapelle et al., 2006), Chapter 25) is devoted to common, if there are any, aspects of the semi-supervised and transductive algorithms. The transductive learning can be formulated as a semi-supervised learning task, since a transductive approach always uses the information of the test data points. Although the discussion on the similarities between the semi-supervised and transductive learning invokes more questions, sometimes philosophical, than solutions, one of the conclusions is that both a semi-supervised approach and a transductive one use the marginal probability of observations. The interesting opinion is that in an asymptotic case, when we have infinitely many unlabeled points, the semi-supervised and the transductive approaches should perform the same thing: induction that somehow uses knowledge of the marginal probability of observations. The weighted semi-supervised estimator, proposed in Section 4.2 is based on similar ideas.

## Self-Training

Self-training is mentioned for the first time in (Scudder, 1965). It has many names. The same learning principle is sometimes called self-learning, self-labeling, decision-directed learning, and bootstrapping. The idea lies in usage of one's own predictions. The algorithm starts on labeled data and in each iterative step a part of unlabeled data, the most confident points (e.g., instances with maximal conditional probabilities of a class given an observation), is labeled according to a current decision rule. Self-training is a wrapper method and the learner has to be chosen. As a result, the method depends on a supervised underlying method. If margin maximization methods are used, then the decision boundary is pushed away from the unlabeled data; and for a number of optimization methods the behavior of self-training is not determined (Chapelle et al., 2006). In spite of its disadvantages, self-training was successfully applied to several real-world problems, e.g., to the word sense disambiguation problem in natural language processing (Yarowsky, 1995).

## Co-Training

Co-training is another approach which can be classified as a semi-supervised learning method. It was introduced by Blum and Mitchell (1998) and exploits the training of several classifiers, each of which is trained on different types of features, in other words, on different “views” of the objects to be classified. These “views” are independent, given a classifier. Unlabeled data are also split into “views”. The constraints impose that labels for all “views” of an observation should be the same. Therefore, each classifier labels data and simultaneously teaches another classifier. For instance, the possible “views” of feature split for a document categorization task can be all words of a document on the one hand and all hyperlinks of the same document on the other hand (Denis et al., 2003). Co-training is based on two important assumptions. The feature split into “views” has to be possible, and each “view” has to be sufficient to train a classifier. The notion of co-training is used for cases with two “views”. If more “views” are invoked, the approach is called multiview learning. We do not apply neither co-training nor multiview training to the tasks considered in this thesis, since it is not obvious how to model the “views”, e.g., in the phonetisation task (Nettalk corpus), previously considered in Section 3.5, it is hardly possible to design two representations of a letter (or a group of letters).

### 4.1.1 FOUR ASSUMPTIONS PROPOSED FOR SEMI-SUPERVISED LEARNING

Discussions around the utility of unlabeled data have been going on since the problem of semi-supervised learning has been formalized. Chapelle et al. (2006) recently proposed four assumptions which state when semi-supervised learning can work.

1. Smoothness assumption. If two points  $x_1, x_2$  in a high-density region are close to each other, then so should be the corresponding outputs  $y_1, y_2$ .
2. Cluster assumption. If points are in a same cluster, they are likely to be of the same class.

Stronger interpretation of the cluster assumption has been formalized by Rigollet (2007) for the binary case. Let  $y$  be a label,  $y \in \{0, 1\}$ ,  $\eta(x) = p(y = 1|x)$ , the conditional probability of a class given an observation,  $C_j, j = 1, 2, \dots$ , be a collection of clusters such that  $C_j \subset \mathcal{X}$ . Then the cluster assumption means that the function  $x \in \mathcal{X} \rightarrow \mathbb{1}\{\eta(x) \geq 1/2\}$  takes a constant value on each of the  $C_j$ .

3. Low-density separation. The decision boundary lies in a low-density region.
4. Manifold assumption. The data lie on a low-dimensional manifold.

The assumptions are clear but they rely more on intuition than on theoretic foundations. So, to our knowledge, the only attempt to formalize the semi-supervised assumptions was done by Rigollet (2007) and concerns the cluster assumption. Although the types of algorithms listed below can implement some assumptions mentioned above, there is not any direct correspondence between the assumptions and methods.

A semi-supervised approach does not necessarily implement all these assumptions. Our semi-supervised criterion proposed in Section 4.2.1 was not initially based on any assumption. As we will see, an interesting observation about the proposed method is that it is most efficient when the Bayes error is very small which correlates well with the intuition underlying most semi-supervised approaches that unlabeled data is most useful if one can assume that the classes are “well-separated”. The notion of “well-separated” classes is common to the smoothness, cluster, and low-density separation assumptions.

### 4.1.2 CATEGORIES OF ALGORITHMS

The decades of semi-supervised learning have been fruitful. Usually (see (Chapelle et al., 2006, Zhu, 2005a)), semi-supervised learning algorithms are divided into four categories listed below. This classification of the semi-supervised approaches provides a brief overview over existing semi-supervised learning methods, but as we will see, it is not complete.

Investigating the categories described below, we consider the difficulties of integrating the unlabeled data into discriminative models and the possible solutions of the problem.

#### Generative Models

Probabilistic generative models fare easily with the use of unlabeled data, usually through Expectation-Maximization (Dempster et al. (1977)). They are the oldest ones among the semi-supervised approaches and are explicitly described in (Seeger, 2002). In a generative framework the log-likelihood of the labeled data is given by

$$\ell_G(\theta) = \sum_{i=1}^{|\mathcal{D}_l|} \log \left\{ p(x_i, y_i | \theta) \right\} = \sum_{i=1}^{|\mathcal{D}_l|} \log \left\{ p(x_i | y_i, \lambda) p(y_i | \pi) \right\}, \quad (4.1)$$

where the parameter  $\theta = (\lambda, \pi)$ . Unlabeled data can be encoded directly and the joint log-likelihood of labeled data  $D_l$  and unlabeled data  $D_u$  is as follows:

$$\ell_G(\theta) = \sum_{i=1}^{|\mathcal{D}_l|} \log \left\{ p(y_i | \pi) p(x_i | y_i, \lambda) \right\} + \sum_{i=|\mathcal{D}_l|+1}^{|\mathcal{D}_l|+|\mathcal{D}_u|} \log \sum_{y \in \mathcal{Y}} \left\{ p(y | \pi) p(x_i | y, \lambda) \right\}. \quad (4.2)$$

The class posteriors  $p(y|x)$  are influenced by both estimated parameters,  $\lambda$  and  $\pi$ . One can notice that the labels  $y$  associated with the unlabeled data can be seen as latent variables. The expectation-maximization algorithm used for optimization is an iterative approach that converges to a local maximum of the log-likelihood function. The expectation-maximization procedure is drafted as Algorithm 6. It has been successfully applied by, e.g., (Mérialdo, 1993), (Nigam et al., 2000) and (Klein and Manning, 2004), to text classification problems with both labeled and unlabeled data.

In contrast, in discriminative models, the class posteriors are modeled directly, hence, one can see that the discriminative model’s likelihood

$$L_D(\theta) = \prod_{i=1}^{|\mathcal{D}_l|} p(y_i | x_i, \theta) \quad (4.3)$$

---

**Algorithm 6** Expectation-Maximization Algorithm
 

---

```

while Until convergence criterion is not met do
  {Expectation step, for every  $x_i$ ,}
  Compute  $p(y|x_i)$ 
  if  $x_i$  is unlabeled then
     $p(y|x_i) \propto p(y|\pi)p(x_i|y, \lambda)$ 
  else
     $p(y|x_i) = \mathbb{1}\{y = y_i\}$ 
  end if
  {Maximize with respect to  $\lambda$  and  $\pi$ }
   $\lambda^{(t+1)}, \pi^{(t+1)} = \arg \max_{\lambda^{(t)}, \pi^{(t)}} \ell_G(\mathcal{D}_l, \mathcal{D}_u, \lambda^{(t)}, \pi^{(t)})$ 
end while

```

---



Figure 4.1: On the left: generative framework; on the right: discriminative framework.

does not take  $D_u$  into consideration and therefore  $D_u$  does not change the posterior belief.

Minka (2005) and Seeger (2002) argue from the Bayesian point of view that in a discriminative model (on the right of Figure 4.1)

$$p(y, x, \theta, \theta') = p(\theta)p(\theta') \prod_{i=1}^N p(y_i|x_i, \theta)p(x_i|\theta'). \quad (4.4)$$

the posterior  $p(\theta|x, y)$  does not depend on the nature of the marginal  $p(x|\theta')$ . In contrast, assuming that  $\theta = \theta'$  gives a generative model (on the left of Figure 4.1).

*Hybrid Models.* The major intuition behind hybrid models is that generative and discriminative models can be mutually complementary.

Bouchard and Triggs (2004) and Holub and Perona (2005) consider hybrid models that are based on a convex combination of a discriminative model likelihood  $L_D(\theta)$  and a generative model likelihood  $L_G(\theta)$

$$\alpha \log L_D(\theta) + (1 - \alpha) \log L_G(\theta),$$

where  $0 \leq \alpha \leq 1$  is a trade-off between two models.

Minka (2005) explores another avenue, further developed in (Lasserre et al., 2006). As we have already mentioned, the case where  $\theta$  and  $\theta'$  in equation (4.4) are unrelated corresponds to the purely discriminative model, where unlabeled data are of no help; taking  $\theta = \theta'$  results in the traditional generative model; introducing via their Bayesian prior distribution dependencies between  $(\theta, \theta')$  allows to build a full range of hybrid models.

The hybrid models are reported to achieve better accuracy than a generative and a discriminative models separately. However, the main drawback of hybrid of generative and discriminative methods is the increased number of parameters to be estimated. Usually, it is doubled, since there is a set of parameters associated with a generative model, and another set, associated with a discriminative model. Lasserre et al. (2006) reports that the optimization in hybrid models can be carried out with the conjugate gradient or expectation-maximization methods.

## Low-Density Separation Methods

They include, first of all, margin maximizing approaches, such as support vector machines (Chapelle and Zien, 2005), transductive support vector machines, and entropy minimization approaches. So, transductive support vector machines use  $p(x)$ , estimated on unlabeled data, to avoid setting the separator  $p(y|x)$  in the high-density regions. The information regularization approach (Szummer and Jaakkola, 2002) is based on a similar idea that labels can not vary very much in the regions where  $p(x)$  is high.

Criterion of Grandvalet and Bengio. The criterion of Grandvalet and Bengio (2004) is particularly important for us, since it integrates unlabeled data into a probabilistic discriminative model. The criterion is based on the idea that classes should be well-separated. This idea was applied to the mixture models integrating unlabeled data (O’Neill, 1978). Later, Castelli and Cover (1996) concluded that information content of unlabeled data decreases as classes overlap. In (Grandvalet and Bengio, 2004), it is Shannon’s conditional entropy over unlabeled data

$$H(y|x) = - \sum_{i=1}^{|\mathcal{D}_u|} \sum_{y \in \mathcal{Y}} p(y|x_i) \log p(y|x_i)$$

that is used as a measure of class overlap. Grandvalet and Bengio (2004) minimize the following semi-supervised criterion embedding an entropy regularization term ( $\rho_{BG}$  is used to tune the strength of the regularizer)

$$\begin{aligned} \ell(\theta) &= - \sum_{i=1}^{|\mathcal{D}_l|} \log p(y_i|x_i; \theta) + \rho_{BG} H(y|x) \\ &= - \sum_{i=1}^{|\mathcal{D}_l|} \log p(y_i|x_i; \theta) + \rho_{BG} \sum_{i=|\mathcal{D}_l|+1}^{|\mathcal{D}_l|+|\mathcal{D}_u|} \sum_{y \in \mathcal{Y}} p(y|x_i; \theta) \log p(y|x_i; \theta). \end{aligned} \quad (4.5)$$

The criterion of Grandvalet and Bengio is significant, since it makes an attempt to introduce unlabeled data into discriminative models. However, the entropy term yields a non-convex criterion, hence one expects local minima.

## Graph-Based Methods

The cluster assumption is also used in graph-based methods, which exploit the intuition that unlabeled data points should receive the same label as their labeled neighbors: in (Zhu and Ghahramani, 2002), a neighborhood graph is used to iteratively propagate labels from labeled to unlabeled data points until convergence.



Graph-based methods are an active area of semi-supervised learning. Most of them are based on the graph Laplacian, which is matrix representation of a graph. Let  $g = (V, E)$  be a graph, and  $w(e)$  - the weight of an edge  $e$ , that is a measure of similarity between nodes. If an edge is missing, two nodes are considered to be independent. A weighted adjacency matrix, which describes the graph, is defined as:

$$\mathbf{W}_{ij} = \begin{cases} w(e), & e = (i, j) \in E, \\ 0, & e = (i, j) \notin E. \end{cases} \quad (4.6)$$

An example of a graph-based approach are transductive algorithms already discussed in Section 4.1. They use the smoothness assumption to label the test points.

### Change of Representation

Many approaches use unlabeled data to induce new representation or new features. These methods are based mainly on the two following steps (e.g., approaches described in (Sha and Saul, 2005) and (Zhu et al., 2005)):

1. An unsupervised step on all data (labels are ignored), that may lead to construction of a new metric or kernel to perform a projection to a low-dimensional space.
2. Ignore the unlabeled data and perform a supervised learning, using the new representation.

The list of considered semi-supervised approaches is not complete, and the algorithms incorporate mostly information about unlabeled observations  $X$ . Additional information regarding labels  $Y$  can be introduced as well. It can be even the case that both marginal probability distributions,  $p(x)$  and  $p(y)$  are provided. However, it is not obvious how to integrate this knowledge into a model. The cases where both distributions are known are discussed below.

### Knowledge of Class Proportions

It was suggested that not only marginal probability of observations can be important. The class proportion knowledge can be used as constraints as well (Joachims, 1999). In some specific applications, some prior knowledge on the distribution of the labels  $Y$  may be available, e.g, one can make an assumption that in a natural text about 50% of capitalized lexical items are named entities. Recently, Mann and McCallum (2007b) introduced class proportions into a regularizer, and the criterion under consideration takes the form:

$$\min_{\theta} - \sum_{i=1}^{|\mathcal{D}_l|} \log \ell_{\theta}(y_i|x_i) + \rho KL(\hat{p}||\hat{p}_{\theta}), \quad (4.7)$$

where  $\hat{p}$  is a distribution of class proportions provided by a human expert,  $\hat{p}_{\theta}$  is a distribution of class proportions associated with the model parameterized by  $\theta$ , computed on unlabeled data

$$\frac{1}{|\mathcal{D}_u|} \sum_{i=|\mathcal{D}_l|+1}^{|\mathcal{D}_l|+|\mathcal{D}_u|} p_{\theta}(Y_i = y)$$

and  $KL$  is the Kullback-Leibler divergence. According to Mann and McCallum (2007b), among the advantages of criteria embedding the class proportions marginals are simplicity, scalability, and robustness.

The previous ideas are connected to estimation under marginal constraints, which have been addressed in the 1960-s, e.g. by Ireland and Kullback (1968) for the case of the contingency tables.

Ireland and Kullback (1968) describe one of possible ways to estimate the joint probability  $p_{ij}$  when the marginals  $p_{i\cdot}$  and  $p_{\cdot j}$  are known, with

$$p_{i\cdot} = \sum_j p_{ij} \quad p_{\cdot j} = \sum_i p_{ij}.$$

The criterion to be optimized is the Kullback-Leibler divergence between the model probabilities and the empirical distribution arising from the data:

$$\sum_{i,j} p_{ij} \log \frac{p_{ij}}{N_{ij}}, \quad (4.8)$$

where  $N_{ij}$  are entries in cells of the contingency table.

Algorithm 7 summarizes the iterative procedure proposed by Ireland and Kullback (1968) to optimize the proposed criterion. The parameters  $a_i$  and  $b_j$  are unknown and are estimated, and  $N = \sum_{i,j} N_{ij}$ . Although the approach is considered for a case of a two-dimensional table, it can be generalized for tables of higher dimensions.

---

**Algorithm 7** Algorithm of C.T. Ireland and S. Kullback

---

INPUT:  $p_{i\cdot}, p_{\cdot j}, N, N_{ij}$

OUTPUT:  $p_{ij}, a_i, b_j$  (values  $a_i$  and  $b_j$  themselves are not of primary importance)

$b_j = 1$

$a_i = p_{i\cdot}N/N_{i\cdot}, p_{ij} = a_i b_j N_{ij}/N$

**while** some stopping criterion/a is/are not met **do**

$b_j = p_{\cdot j}N/(\sum_i a_i N_{ij}), p_{ij} = a_i b_j N_{ij}/N$

$a_i = p_{i\cdot}N/(\sum_j b_j N_{ij}), p_{ij} = a_i b_j N_{ij}/N$

**end while**

---

### 4.1.3 SEMI-SUPERVISED CRFS

Semi-supervised learning has also been applied to structured output prediction tasks. Altun et al. (2005) and Brefeld and Scheffer (2006) describe a maximum margin semi-supervised learning approaches for structured output prediction; Jiao et al. (2006), Mann and McCallum (2008), and Mann and McCallum (2007a) discuss semi-supervised learning for conditional random fields.

Jiao et al. (2006) applied the minimum entropy regularization approach of Grandvalet and Bengio, already mentioned as equation (4.5), for conditional random fields:

$$-\sum_{i=1}^{|\mathcal{D}_l|} \log p_{\theta}(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) + \frac{\|\theta\|^2}{2\sigma^2} - \rho_{BG} \sum_{i=|\mathcal{D}_l|+1}^{|\mathcal{D}_l|+|\mathcal{D}_u|} \sum_{\mathbf{y}} p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)}) \log p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)}). \quad (4.9)$$

The direct computation of the gradient of the entropy term requires  $O(T^2|Y|^3)$  operations in comparison to  $O(T|Y|^2)$  of a standard forward-backward procedure. Mann and McCallum (2007a) proposed an efficient way (complexity of a standard forward-backward algorithm) to compute the gradient of the criterion presented in (4.9).

A hybrid semi-supervised model is proposed in (Suzuki et al., 2007). The model combines discriminative and generative models, the parameters  $\Gamma = \{\{\gamma_i\}_{i=1}^I, \{\gamma_j\}_{j=I+1}^{I+J}\}$  are associated with  $I$  generative and  $J$  discriminative models. Unlabeled data are introduced into the generative models, as we discussed above in Section 4.1.2. The following criterion

$$p(\mathbf{y}|\mathbf{x}, \Lambda, \Theta, \Gamma) \propto \prod_i p_i^D(\mathbf{y}|\mathbf{x}, \lambda_i)^{\gamma_i} \prod_j p_j^G(\mathbf{x}, \mathbf{y}, \theta_j)^{\gamma_j} \quad (4.10)$$

contains three sets of parameters to be estimated,  $\Gamma$ ,  $\Lambda$ , and  $\Theta$ . The values of  $\Lambda$  are estimated on labeled data. An iterative optimization procedure run until convergence is used to adjust  $\Gamma$  (parameters of hybrid models) and parameters  $\Theta$  associated with discriminative components.

Suzuki and Isozaki (2008) introduce a semi-supervised approach that is simpler than the one proposed in (Suzuki et al., 2007), since there are only two parameter vectors to be estimated. The parameter vector  $\Lambda$  is estimated on labeled data using a discriminative model, and  $\Theta$  on unlabeled data, using a generative approach.

Results reported on CoNLL 2003 and CoNLL 2000 data sets achieve state-of-the art performance. (See Chapter 3 for the state-of-the art and baselines values of performance.) The following results are provided as F-score. On CoNLL 2003 corpus, the semi-supervised CRF and the hybrid models reach respectively 84.4 and 87.2. On the CoNLL 2000 data set, the semi-supervised CRF achieves 93.87, and the hybrid model 94.3.

Daumé III (2009) called the approach discussed in (Suzuki et al., 2007) a great step forward in hybrid models, since it combines models that take underlying structure into account, namely hidden Markov models and conditional random fields. The approach of Suzuki and Isozaki (2008) has been recently applied to parsing problems by Suzuki et al. (2009).

One of the recent works on semi-supervised learning applied to natural language processing is a trial to add incomplete annotations (Tsuboi et al., 2008). Ambiguous annotations are considered as candidate labels, and parameters are estimated by marginalizing out the unknown labels. The method is a particular case of hidden conditional random fields, introduced in (Quattoni et al., 2004) and mentioned in Chapter 3.

The idea to introduce the knowledge of labels proportions, the method called “expectation regularization”, proposed in (Mann and McCallum, 2007b) for maximum entropy models, has been generalized in (Mann and McCallum, 2008) for structured output prediction, using linear-chain CRFs. The approach was called generalized expectation. It was supposed that not only fully labeled instances can be used but labeled features as well. The proposed criterion

$$-\sum_{i=1}^N \log p_{\theta}(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) + \frac{\|\theta\|^2}{2\sigma^2} + \rho KL(\hat{p}||\tilde{p}_{\theta})$$

uses the values of  $\hat{p}$  provided by an expert.

The K-similar conditional random fields (Chen et al., 2008) method relies on an assumption that a word can be labeled using knowledge of labels of similar words. Similarity can be measured using standard coefficients: inner product, cosine coefficient, Dice coefficient, and Jaccard coefficient. The unlabeled data are used to compute the similarity between words. The criterion

$$p_{\theta}(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\theta}(\mathbf{x})} \exp \left\{ \sum_{\mathbf{x}' \in S(\mathbf{x})} (w\theta)^T F(\mathbf{y}, \mathbf{x}') \right\} \quad (4.11)$$

where  $S(\mathbf{x})$  is a set of  $k$ -similar words of  $\mathbf{x}$  and  $w$  are corresponding similarity weights, incorporates the similarity features in the linear-chain CRFs.

Although all described above semi-supervised methods for sequential labeling are reported to be efficient, they are either application dependent (such as K-similar CRFs and generalized expectation), or violate the convexity (as the minimum entropy regularization approach), or suffer from an increased complexity (such as the hybrid semi-supervised method of Suzuki *et al.*).

In the following, we propose a semi-supervised estimator, that is based on the introduction of the marginal probability of observations  $p(x)$  into a discriminative model. The approach is application independent, the criterion is convex and therefore the first- and second-order numerical optimization methods can be applied directly. Using  $p(x)$  as weights does not change the model's complexity.

## 4.2 MARGINAL PROBABILITY IN DISCRIMINATIVE MODELS

---

As we have seen, it is easy to introduce unlabeled data into generative models. It is however an extensively documented fact that discriminative models perform better than generative models for classification tasks (Ng and Jordan, 2002, Liang and Jordan, 2008). Integrating unlabeled data into discriminative models is a much more challenging issue. Put in probabilistic terms, when learning to predict an output  $y$  from an observation  $x$ , a discriminative model attempts to fit  $p(y|x; \theta)$ , where  $\theta$  denotes the parameter. The role to be played by any available prior knowledge about the marginal probability  $p(x)$  in this context is not obvious.

In general, as we discussed in previous section, the most common approach is to make the unknown parameter vector  $\theta$  depend on the unlabeled data, either directly or indirectly. One way to achieve this goal is to use the unlabeled data to enforce constraints on the shape of  $p(y|x)$ : the cluster assumption, for instance, stipulates that the decision boundary should be located in low density regions. This approach, as any attempt to distort the supervised training criterion with supplementary terms faces two risks:

- to turn a well-behaved convex optimization problem into a non-convex one, fraught with local optima, thus making the results highly dependent of a proper initialization;
- to lose the asymptotic consistency property of the usual (conditional maximum likelihood) estimator.

As a result, these methods are not guaranteed to improve over a trivial baseline which would only use the available annotated data. They furthermore require a fine tuning of the various optimization parameters as in (Mann and McCallum, 2007b).

We try to challenge the view that unlabeled data cannot help purely discriminative models by exhibiting a semi-supervised estimator of the parameter  $\theta$  which is asymptotically optimal and, in some situations, preferable to the usual maximum (conditional) likelihood estimator. To this aim, we make the simplifying assumption that the marginal  $p(x)$  is fully known, which is true in the limit of infinitely many unlabeled data.

#### 4.2.1 ASYMPTOTICALLY OPTIMAL SEMI-SUPERVISED ESTIMATION

Let  $g(y|x; \theta)$  denote the conditional probability density function (pdf) corresponding to a discriminative probabilistic model parametrized by  $\theta \in \Theta$ . The case when  $\eta(x) \neq g(y|x; \theta_*)$  is referred to as misspecification. In the following, we will always assume that the class variable  $Y$  takes its values in a finite set,  $\mathcal{Y}$ , with a special interest for the binary case where  $\mathcal{Y} = \{0, 1\}$ . We will further assume that the input (or explanatory) variable  $X$  also takes its values in a finite set  $\mathcal{X}$ , which may be arbitrary large. Such an assumption is made to simplify the mathematical framework. At the same time, the assumption coincides with the settings of real-world applications.

The training procedure has access to a set of  $n$  i.i.d. labeled observations,  $(X_i, Y_i)_{1 \leq i \leq n}$ , as well as to a potentially unlimited number of unlabeled observations, where the quantity of unlabeled data is so large that we can consider that the marginal probability of  $X$  is fully known.

Finally, for a function  $f : \mathbb{R}^p \mapsto \mathbb{R}$ , we denote by  $\nabla_z f(z_*)$  the  $p \times 1$  gradient vector and by  $\nabla_{z^\top} \nabla_z f(z_*)$  the  $p \times p$  Hessian matrix in  $z_*$ . When  $f : \mathbb{R}^p \mapsto \mathbb{R}^r$ , the notation  $\nabla_{z^\top} f(z_*)$  will be used to denote the  $r \times p$  Jacobian matrix in  $z_*$ .

#### Connection with Stratified Sampling

We first consider the case where the “model” of interest is very basic and simply consists in estimating the complete joint probability of  $X$  and  $Y$ , which is denoted by  $\pi(x, y)$ . We will also denote by  $\eta(y|x)$  and  $q(x)$ , respectively, the conditional and the marginal probabilities associated with  $\pi$ . Although this case is not directly of interest for statistical learning, it highlights the role played by the knowledge of the marginal  $q$  in semi-supervised learning.

It is well known that the maximum-likelihood estimator of  $\pi(x, y)$  defined by

$$\hat{\pi}_n(x, y) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{X_i = x, Y_i = y\} \quad (4.12)$$

is asymptotically efficient with asymptotic variance  $v(x, y) = \pi(x, y)(1 - \pi(x, y))$  (assuming that  $0 < \pi(x, y) < 1$ ).

Assume now that we are given  $q(x)$ , the marginal distribution of  $X$ , and that  $0 < q(x) < 1$ . It is easily checked that the maximum-likelihood estimator of  $\pi(x, y)$  subject to

the marginal constraint that  $\sum_{y \in \mathcal{Y}} \pi(x, y) = q(x)$  is given by

$$\hat{\pi}_n^s(x, y) = \frac{\sum_{i=1}^n \mathbb{1}\{X_i = x, Y_i = y\}}{\sum_{i=1}^n \mathbb{1}\{X_i = x\}} q(x), \quad (4.13)$$

where the superscript  $s$  stands for “semi-supervised” and the ratio is recognized as the maximum-likelihood estimate of the conditional probability  $\eta(y|x)$ . As  $\hat{\pi}_n^s(x, y)$  is a ratio of two simple estimators, its asymptotic variance can be computed using the  $\delta$ -method, yielding

$$v^s(x, y) = \pi(x, y)(1 - \pi(x, y)/q(x)).$$

As  $0 < \pi(x, y) \leq q(x) < 1$ ,  $v^s(x, y)$  is less than  $v(x, y)$ . Hence, in general the semi-supervised estimator  $\hat{\pi}_n^s(x, y)$  and  $\hat{\pi}_n(x, y)$  are not asymptotically equivalent, and  $\hat{\pi}_n^s(x, y)$  is preferable. More precisely,  $v^s(x, y)/v(x, y) = (1 - \pi(x, y)/q(x))/(1 - \pi(x, y))$  which tends to zero as  $\pi(x, y)$  gets closer to  $q(x)$ . In other words, the performance of  $\hat{\pi}_n^s(x, y)$  is all the more appreciable, compared to that of  $\hat{\pi}_n(x, y)$ , that  $y$  is a frequent label for  $x$ . In this case, knowledge of the marginal  $q(x)$  makes it possible to obtain a precise estimate of  $\hat{\pi}_n^s(x, y) \approx q(x)$  even with a very limited number of observations of  $x$ .

The classical statistical use of this result consists in estimating marginal probabilities  $p(y)$  according to

$$\hat{p}_n^s(y) = \sum_x \hat{\pi}_n^s(x, y).$$

To determine the asymptotic variance of the stratified estimator  $\hat{p}_n^s(y)$ , it is first easily shown that  $\hat{\pi}_n^s(x_1, y)$  and  $\hat{\pi}_n^s(x_2, y)$  are asymptotically uncorrelated when  $x_1 \neq x_2$  and then by rewriting  $v^s(x, y)$  as  $q(x)\eta(y|x)(1 - \eta(y|x))$  one obtains the classic formula

$$\sum_x q(x)\eta(y|x)(1 - \eta(y|x)) = \mathbb{E}_q(V_\eta[\mathbb{1}\{Y = y\}|X]),$$

which is indeed smaller than  $V_\pi[\mathbb{1}\{Y = y\}]$  for the un-stratified estimator

$$\hat{p}_n(y) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{Y_i = y\}$$

due to the well-known Rao-Blackwell variance decomposition

$$V_\pi[f(Y)] = \mathbb{E}_q(V_\eta[f(Y)|X]) + V_q(\mathbb{E}_\eta[f(Y)|X]). \quad (4.14)$$

## Estimation in General Discriminative Models

We now consider the extension of the previous simple observation to the case of a general discriminative probabilistic model; the main difference being the fact that a given parametric model  $\{g(y|x; \theta)\}_{\theta \in \Theta}$  will generally not be able to fit exactly the actual conditional distribution  $\eta(y|x)$  of the data. As in the fully-specified case above, it is nonetheless possible to exhibit a semi-supervised estimator which is asymptotically optimal and preferable to the usual conditional maximum likelihood estimator defined by

$$\hat{\theta}_n = \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n \ell(Y_i|X_i; \theta) \quad (4.15)$$

where  $\ell(y|x; \theta) = -\log g(y|x; \theta)$  denotes the inverse of the conditional log-likelihood function.

Under the (classical) assumptions of Theorem 4.1 below,  $\frac{1}{n} \sum_{i=1}^n \ell(Y_i|X_i; \theta)$  tends, uniformly in  $\theta$ , to  $E_\pi[\ell(Y|X; \theta)]$  and thus the limiting value of  $\hat{\theta}_n$  is given by

$$\theta_\star = \arg \min_{\theta \in \Theta} E_\pi[\ell(Y|X; \theta)] \quad (4.16)$$

The maximum likelihood estimator in (4.15) may also be interpreted as

$$\hat{\theta}_n = \arg \min_{\theta \in \Theta} E_{\hat{\pi}_n}[\ell(Y|X; \theta)]$$

where

$$\hat{\pi}_n(x, y) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{X_i = x, Y_i = y\}$$

denotes the empirical measure associated with the sample  $(X_i, Y_i)_{1 \leq i \leq n}$ , which also coincides with the maximum likelihood estimate of  $\pi(x, y)$  defined in (4.12).

If we now assume that the marginal  $q(x)$  is available, we know that  $\hat{\pi}_n(x, y)$  is dominated (asymptotically) by the estimator  $\hat{\pi}_n^s(x, y)$  defined in (4.13), which we here particularize to

$$\hat{\pi}_n^s(x, y) = \begin{cases} \frac{\sum_{i=1}^n \mathbb{1}\{X_i=x, Y_i=y\}}{\sum_{i=1}^n \mathbb{1}\{X_i=x\}} q(x) & \text{if } \sum_{i=1}^n \mathbb{1}\{X_i = x\} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.17)$$

By analogy with the construction used in the absence of information on  $q$ , we now define the corresponding semi-supervised estimator as  $\hat{\theta}_n^s = \arg \min_{\theta \in \Theta} E_{\hat{\pi}_n^s}[\ell(Y|X; \theta)]$ , where the notation  $E_{\hat{\pi}_n^s}[f(Y, x)] = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \hat{\pi}_n^s(x, y) f(x, y)$  is used somewhat loosely here as it may happen that, for finite  $n$ ,  $\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \hat{\pi}_n^s(x, y) < 1$ , although  $\hat{\pi}_n^s(x, y)$  sums to one with probability one, for sufficiently large  $n$ . It is easily checked that  $\hat{\theta}_n^s$  may also be rewritten as

$$\hat{\theta}_n^s = \arg \min_{\theta \in \Theta} \sum_{i=1}^n \frac{q(X_i)}{\sum_{j=1}^n \mathbb{1}\{X_j = X_i\}} \ell(Y_i|X_i; \theta) \quad (4.18)$$

Eq. (4.18) is a weighted version of (4.15) where the weight given to observations that share the same input  $x$  is common and reflects our prior knowledge on the marginal  $q(x)$ .

**Theorem 4.1.** *Let the joint probability of  $X$  and  $Y$  factorize as  $\pi(x, y) = \eta(y|x)q(x)$ , where  $q$  is known, and define the following matrices*

$$H(\theta_\star) = E_q (V_\eta [\nabla_\theta \ell(Y|X; \theta_\star)|X]) \quad (4.19)$$

$$I(\theta_\star) = E_\pi \left[ \nabla_\theta \ell(Y|X; \theta_\star) \{ \nabla_\theta \ell(Y|X; \theta_\star) \}^T \right] \quad (4.20)$$

$$J(\theta_\star) = E_\pi [\nabla_{\theta^T} \nabla_\theta \ell(Y|X; \theta_\star)] \quad (4.21)$$

*Assume that (1)  $\mathcal{X}$  and  $\mathcal{Y}$  are finite sets; (2)  $\pi(x, y) > 0$  for all  $(x, y) \in \mathcal{X} \times \mathcal{Y}$ ; (3) for all  $(x, y) \in \mathcal{X} \times \mathcal{Y}$ ,  $\ell(y|x; \theta)$  is bounded on  $\Theta$ ; (4)  $\theta_\star$  is the unique minimizer of  $E_\pi[\ell(Y|X; \theta)]$  on  $\Theta$ ; (5) for all  $(x, y) \in \mathcal{X} \times \mathcal{Y}$ ,  $\ell(y|x; \theta)$  is twice continuously differentiable on  $\Theta$ ; (6) the matrices  $H(\theta_\star)$  and  $J(\theta_\star)$  are non singular.*

Then,  $\hat{\theta}_n$  and  $\hat{\theta}_n^s$  are consistent and asymptotically normal estimators of  $\theta_*$ , which satisfy

$$\sqrt{n} \left( \hat{\theta}_n - \theta_* \right) \xrightarrow{L} \mathcal{N} \left( 0, J^{-1}(\theta_*) I(\theta_*) J^{-1}(\theta_*) \right) \quad (4.22)$$

$$\sqrt{n} \left( \hat{\theta}_n^s - \theta_* \right) \xrightarrow{L} \mathcal{N} \left( 0, J^{-1}(\theta_*) H(\theta_*) J^{-1}(\theta_*) \right) \quad (4.23)$$

Furthermore,  $\hat{\theta}_n^s$  is asymptotically efficient.

*Proof.* First note that (4.22) is the well-known result that pertains to the behavior of the maximum likelihood estimator in misspecified models – see, for instance, (White, 1982) or Lemma 1 of (Shimodaira, 2000).

Now, the fact that  $\hat{\theta}_n^s = \arg \min_{\theta \in \Theta} \mathbb{E}_{\hat{\pi}_n^s} [\ell(Y|X; \theta)]$  implicitly defines the semi-supervised estimator  $\hat{\theta}_n^s$  as a function of the maximum-likelihood estimator of the conditional probabilities

$$\hat{\eta}_n(y|x) = \frac{\sum_{i=1}^n \mathbb{1}\{X_i = x, Y_i = y\}}{\sum_{i=1}^n \mathbb{1}\{X_i = x\}}$$

In our setting, the conditional probability  $\eta$  may be represented by a finite dimensional vector block defined by  $\boldsymbol{\eta} = (\boldsymbol{\eta}(x_1), \dots, \boldsymbol{\eta}(x_d))^\top$ , where  $\boldsymbol{\eta}(x_i) = (\eta(y_1|x_i), \dots, \eta(y_k|x_i))^\top$ ,  $\{x_1, \dots, x_d\}$  denote the elements of  $\mathcal{X}$ , and,  $\{y_0, \dots, y_k\}$  denote the elements of  $\mathcal{Y}$ . As usual in polytomous regression models, we omit one of the possible values of  $Y$  (by convention,  $y_0$ ) due to the constraint that  $\sum_{y \in \mathcal{Y}} \eta(y|x) = 1$ , for all  $x \in \mathcal{X}$ . The estimator  $\hat{\boldsymbol{\eta}}_n$  is defined similarly with  $\hat{\eta}_n(y|x)$  substituted for  $\eta_n(y|x)$ .  $\hat{\boldsymbol{\eta}}_n$  is the maximum likelihood estimator of  $\boldsymbol{\eta}$  and it is asymptotically efficient with asymptotic covariance matrix given by  $K^{-1}(\boldsymbol{\eta})$ , the inverse of the Fisher information matrix for  $\boldsymbol{\eta}$ , block-defined by

$$K^{-1}(\boldsymbol{\eta}) = \text{diag} \left( K^{-1}(x_1; \boldsymbol{\eta}), \dots, K^{-1}(x_d; \boldsymbol{\eta}) \right)$$

where

$$K^{-1}(x_i; \boldsymbol{\eta}) = q(x_i)^{-1} \left\{ \text{diag} \left( \boldsymbol{\eta}(x_i) \right) - \boldsymbol{\eta}(x_i) \boldsymbol{\eta}^\top(x_i) \right\} \quad (4.24)$$

To obtain the asymptotic behavior of the semi-supervised estimator  $\hat{\theta}_n^s$ , remark that  $\hat{\theta}_n^s$  is obtained as a function  $\psi$  of  $\hat{\boldsymbol{\eta}}_n$ , where  $\psi$  is implicitly defined by the optimality equation  $s(\boldsymbol{\eta}, \psi(\boldsymbol{\eta})) = 0$  where  $s$  is the (negative of the) score function defined by

$$s(\boldsymbol{\eta}, \theta) = \nabla_{\theta} \mathbb{E}_{\pi} [\nabla_{\theta} \ell(Y|X; \theta)] = \sum_{x \in \mathcal{X}} q(x) \sum_{y \in \mathcal{Y}} \eta(y|x) \nabla_{\theta} \ell(y|x; \theta) \quad (4.25)$$

Because  $\theta_* = \psi(\boldsymbol{\eta})$  and  $\hat{\theta}_n^s = \psi(\hat{\boldsymbol{\eta}}_n)$ ,  $\hat{\theta}_n^s$  is an asymptotically efficient estimator of  $\theta_*$  with asymptotic covariance matrix given by  $\nabla_{\boldsymbol{\eta}^\top} \psi(\boldsymbol{\eta}) K^{-1}(\boldsymbol{\eta}) \left\{ \nabla_{\boldsymbol{\eta}^\top} \psi(\boldsymbol{\eta}) \right\}^\top$ . The Jacobian matrix  $\nabla_{\boldsymbol{\eta}^\top} \psi(\boldsymbol{\eta})$  may be evaluated thanks to the implicit function theorem as

$$\nabla_{\boldsymbol{\eta}^\top} \psi(\boldsymbol{\eta}) = \left\{ \nabla_{\theta^\top} s(\boldsymbol{\eta}, \theta_*) \right\}^{-1} \nabla_{\boldsymbol{\eta}^\top} s(\boldsymbol{\eta}, \theta_*)$$

From the definition of the score function in (4.25), it is obvious that  $\nabla_{\theta^\top} s(\boldsymbol{\eta}, \theta_*) = J(\theta_*)$ . In order to calculate  $\nabla_{\boldsymbol{\eta}^\top} s(\boldsymbol{\eta}, \theta_*)$ , we differentiate the rightmost expression in (4.25) using the fact that  $\eta(y_0|x) = 1 - \sum_{y \neq y_0} \eta(y|x)$  to obtain

$$\frac{\partial s(\boldsymbol{\eta}, \theta)}{\partial \eta(x|y)} = q(x) \left[ \nabla_{\theta} \ell(y|x; \theta) - \nabla_{\theta} \ell(y_0|x; \theta) \right]$$



Thus  $\nabla_{\boldsymbol{\eta}^\top} s(\boldsymbol{\eta}, \boldsymbol{\theta}_\star)$  is obtained as the concatenation of the  $d$  ( $k \times 1$ ) vectors  $\nabla_{\boldsymbol{\eta}(x_i)^\top} s(\boldsymbol{\eta}, \boldsymbol{\theta}_\star)$  (for  $i = 1, \dots, d$ ) where

$$\begin{aligned} \nabla_{\boldsymbol{\eta}(x_i)^\top} s(\boldsymbol{\eta}, \boldsymbol{\theta}_\star) = \\ (q(x_i) [\nabla_{\theta} \ell(y_1|x_i; \boldsymbol{\theta}_\star) - \nabla_{\theta} \ell(y_0|x_i; \boldsymbol{\theta}_\star)], \dots, q(x_i) [\nabla_{\theta} \ell(y_k|x_i; \boldsymbol{\theta}_\star) - \nabla_{\theta} \ell(y_0|x_i; \boldsymbol{\theta}_\star)])^\top \end{aligned} \quad (4.26)$$

The asymptotic covariance matrix of  $\hat{\boldsymbol{\theta}}_n^s$  is given by

$$J^{-1}(\boldsymbol{\theta}_\star) \left[ \sum_{x \in \mathcal{X}} \nabla_{\boldsymbol{\eta}(x)^\top} s(\boldsymbol{\eta}, \boldsymbol{\theta}_\star) K^{-1}(x; \boldsymbol{\eta}) \left\{ \nabla_{\boldsymbol{\eta}(x)^\top} s(\boldsymbol{\eta}, \boldsymbol{\theta}_\star) \right\}^\top \right] J^{-1}(\boldsymbol{\theta}_\star)$$

Tedious but straightforward calculations, using (4.24), (4.26) and the fact that  $\eta(y_0|x) = 1 - \sum_{y \neq y_0} \eta(y|x)$  show that

$$\begin{aligned} \nabla_{\boldsymbol{\eta}(x)^\top} s(\boldsymbol{\eta}, \boldsymbol{\theta}_\star) K^{-1}(x; \boldsymbol{\eta}) \left\{ \nabla_{\boldsymbol{\eta}(x)^\top} s(\boldsymbol{\eta}, \boldsymbol{\theta}_\star) \right\}^\top = \sum_{y \in \mathcal{Y}} \nabla_{\theta} \ell(y|x; \boldsymbol{\theta}_\star) \left\{ \nabla_{\theta} \ell(y|x; \boldsymbol{\theta}_\star) \right\}^\top \eta(y|x) \\ - \left( \sum_{y \in \mathcal{Y}} \nabla_{\theta} \ell(y|x; \boldsymbol{\theta}_\star) \eta(y|x) \right) \left( \sum_{y \in \mathcal{Y}} \nabla_{\theta} \ell(y|x; \boldsymbol{\theta}_\star) \eta(y|x) \right)^\top, \end{aligned}$$

which concludes the proof.  $\square$

Theorem 4.1 asserts that the asymptotic covariance matrix associated with  $\hat{\boldsymbol{\theta}}_n^s$  is optimal. Understanding the relations between  $H(\boldsymbol{\theta}_\star)$  and  $I(\boldsymbol{\theta}_\star)$  is thus important to assess the asymptotic performance achievable by any semi-supervised training method which assumes prior knowledge of  $q(x)$ . The multivariate generalization of the Rao-Blackwell variance decomposition (4.14) shows that

$$I(\boldsymbol{\theta}_\star) - H(\boldsymbol{\theta}_\star) = V_q(\mathbb{E}_\eta [\nabla_{\theta} \ell(Y|X; \boldsymbol{\theta}_\star)|X])$$

As a result, the difference between both estimators will mostly depend on whether

$\mathbb{E}_\eta [\nabla_{\theta} \ell(Y|X; \boldsymbol{\theta}_\star)|X = x]$  varies significantly or not around 0 as a function of  $x$ , given that, by definition,  $\boldsymbol{\theta}_\star$  is such that  $\mathbb{E}_q(\mathbb{E}_\eta [\nabla_{\theta} \ell(Y|X; \boldsymbol{\theta}_\star)|X]) = 0$ .

Note that in the particular case where the model is well-specified, in the sense that  $\boldsymbol{\theta}_\star$  is such that  $g(y|x; \boldsymbol{\theta}_\star) = \eta(y|x)$  for all  $(x, y) \in \mathcal{X} \times \mathcal{Y}$ , not only is  $\mathbb{E}_q(\mathbb{E}_\eta [\nabla_{\theta} \ell(Y|X; \boldsymbol{\theta}_\star)|X])$  null but one indeed has the stronger result that for all  $x \in \mathcal{X}$ ,  $\mathbb{E}_\eta [\nabla_{\theta} \ell(Y|X; \boldsymbol{\theta}_\star)|X = x] = 0$ . This is the only case for which  $H(\boldsymbol{\theta}_\star) = I(\boldsymbol{\theta}_\star)$ , and hence, where both estimators are asymptotically equivalent; it is also well known that in this case  $J(\boldsymbol{\theta}_\star) = I(\boldsymbol{\theta}_\star)$  so that all asymptotic covariance matrices coincide with the usual expression of the inverse of the Fisher information matrix for  $\theta$ . Theorem 4.1 gives formal support to the intuition that it is impossible to improve over the classic maximum likelihood estimator for large  $n$ 's when the model is well-specified, even when the marginal  $q$  is known.

The results of Theorem 4.1 are stated in terms of parameter estimation which is usually not the primary interest for statistical learning tasks. Due to the non-differentiability of the 0–1 loss, it is not directly possible to derive results pertaining to the error probability from Theorem 4.1. One may however state the following result in terms of the logarithmic risk, in which the negated log-likelihood  $\ell(y|x; \boldsymbol{\theta})$  is interpreted as a loss function.

**Corollary 4.2.** *In addition to the assumptions of Theorem 4.1, assume that  $\ell(y|x; \theta)$  has bounded second derivative on  $\Theta$ . Then, the logarithmic risk admits the following asymptotic equivalent:*

$$\mathbb{E}_{\pi^{\otimes n}} \{ \mathbb{E}_{\pi} [\ell(Y|X; \hat{\theta}_n)] \} = \mathbb{E}_{\pi} [\ell(Y|X; \theta_{\star})] + \frac{1}{2n} \text{trace} \{ I(\theta_{\star}) J^{-1}(\theta_{\star}) \} + o\left(\frac{1}{n}\right)$$

where  $\mathbb{E}_{\pi^{\otimes n}}$  denotes the expectation with respect to the training data  $(X_i, Y_i)_{1 \leq i \leq n}$ ; for the semi-supervised estimator  $\hat{\theta}_n^s$ , the first order term is given by  $\frac{1}{2n} \text{trace} \{ H(\theta_{\star}) J^{-1}(\theta_{\star}) \}$ .

*Proof.* Corollary 4.2 is based on the classical asymptotic expansion of  $\mathbb{E}_{\pi} [\ell(Y|X; \hat{\theta}_n)] - \mathbb{E}_{\pi} [\ell(Y|X; \theta_{\star})]$  as  $\frac{1}{2}(\hat{\theta}_n - \theta_{\star})^T J(\theta_{\star})(\hat{\theta}_n - \theta_{\star}) + o_p(\frac{1}{n})$ , see, for instance, (Bach, 2006).  $\square$

## 4.2.2 COVARIATE SHIFT

Usually machine learning approaches make a drastic simplification, assuming that training and test samples are drawn from the same distribution. This assumption does not hold in practice and the cases of differing training and test distributions are being studied, e.g., by Bickel et al. (2007) and Sugiyama et al. (2007). The reason of different distributions can be the so-called sample selection bias problem (see e.g. (Cortes et al., 2008)). The sample selection bias problem implies that training points are drawn from the test distribution but some of instances are not available during the training procedure.

The simplest model of covariate shift consists in assuming that  $q_0(x)$  is determined by a sampling scheme and  $q_1(x)$  is determined by a population. The complete joint probabilities of training and testing distributions are  $\pi_0(x, y) = q_0(x)g(y|x)$  and  $\pi_1(x, y) = q_1(x)g(y|x)$ , and in the following the expectations  $\mathbb{E}_0$  and  $\mathbb{E}_1$  are taken with respect to  $\pi_0(x, y)$  and  $\pi_1(x, y)$  respectively. Interestingly the weighting approaches used in this setting, e.g., in (Shimodaira, 2000) have some similarities with the proposed semi-supervised estimator.

In the absence of covariate shift:

$$\lim_{n \rightarrow \infty} \frac{q_1(x_i)}{n^{-1} \sum_{j=1}^n \mathbb{1}\{x_j = x_i\}} \longrightarrow 1,$$

with a covariate shift, we have:

$$\lim_{n \rightarrow \infty} \frac{q_1(x_i)}{n^{-1} \sum_{j=1}^n \mathbb{1}\{x_j = x_i\}} \longrightarrow \frac{q_1(x_i)}{q_0(x_i)}.$$

Considering the logistic risk criterion based on test sample marginal

$$C(\theta) = - \sum_{x \in X} q_1(x) \sum_{y \in Y} \eta(y|x) \log \ell(y|x; \theta), \quad (4.27)$$

Shimodaira (2000) introduced the weighted estimator

$$\begin{aligned} \hat{\pi}_n^w &= \frac{1}{n} \sum_{i=1}^n \frac{q_1(X_i)}{q_0(X_i)} \mathbb{1}\{X_i = x, Y_i = y\} \\ &= \frac{1}{n} \sum_{i=1}^n w(X_i) \mathbb{1}\{X_i = x, Y_i = y\} \end{aligned}$$

and proved that  $w(x) = q_1(x)/q_0(x)$  is the optimal weight if  $n$  is sufficiently large, i.e. asymptotically. The key idea lies in the importance sampling identity:

$$\begin{aligned} \mathbb{E}_0\left[\frac{q_1(x)}{q_0(x)} \log \ell(y|x; \theta)\right] &= \sum_{x \in X, y \in Y} q_0(x) \eta(y|x) \frac{q_1(x)}{q_0(x)} \log \ell(y|x; \theta) \\ &= \mathbb{E}_1[\log \ell(Y|X; \theta)]. \end{aligned}$$

The form of the semi-supervised estimator in (4.18) shows that  $\hat{\theta}_n^s$  will be consistent also in the presence of covariate shift, whereas the logistic regression estimates can only be consistent in this case if we assume that the model is well-specified (Shimodaira, 2000). In the presence of covariate shift however, the expressions of the asymptotic covariance matrices are given by the following proposition.

**Proposition 4.3.** *Assuming that the training distribution  $\pi_0(x, y) = q_0(x)\eta(y|x)$  satisfies the assumptions of Theorem 4.1 and that  $q_1(x)/q_0(x) > 0$ , the semi supervised estimator used with  $q(x) = q_1(x)$  converges to  $\theta_{1,\star} = \arg \min_{\theta \in \Theta} \mathbb{E}_{\pi_1}[\ell(Y|X; \theta)]$  with asymptotic variance given by  $J_1^{-1}(\theta_{1,\star})H_{0,1}(\theta_{1,\star})J_1^{-1}(\theta_{1,\star})$  where*

$$J_1(\theta_{1,\star}) = \mathbb{E}_{\pi_1} [\nabla_{\theta^T} \nabla_{\theta} \ell(Y|X; \theta_{1,\star})] \quad (4.28)$$

$$H_{0,1}(\theta_{1,\star}) = \mathbb{E}_{q_1} \left[ \frac{q_1}{q_0}(X) \mathbb{V}_{\eta} (\nabla_{\theta} \ell(Y|X; \theta_{1,\star})|X) \right] \quad (4.29)$$

By comparison, the weighted estimator  $\hat{\theta}_n^w = \arg \min_{\theta \in \Theta} \sum_{i=1}^n \frac{q_1}{q_0}(X_i) \ell(Y_i|X_i; \theta)$ , which, in addition, assumes knowledge of  $q_0$  has a larger asymptotic variance given by

$$J_1^{-1}(\theta_{1,\star})I_{0,1}(\theta_{1,\star})J_1^{-1}(\theta_{1,\star}),$$

where

$$I_{0,1}(\theta_{1,\star}) = \mathbb{E}_{\pi_1} \left[ \frac{q_1}{q_0}(X) \nabla_{\theta} \ell(Y|X; \theta_{1,\star}) \{ \nabla_{\theta} \ell(Y|X; \theta_{1,\star}) \}^T \right] \quad (4.30)$$

*Proof.* Asymptotic normality of the weighted ML estimator is proved in Lemma 1 of (Shimodaira, 2000). For the semi-supervised estimator, the only change that is needed to the proof of Theorem 4.1 above is to replace (4.24) by

$$K^{-1}(x_i; \boldsymbol{\eta}) = q_0(x_i)^{-1} \{ \text{diag}(\boldsymbol{\eta}(x_i)) - \boldsymbol{\eta}(x_i)\boldsymbol{\eta}^T(x_i) \}$$

as the training observations are distributed under  $\pi_0$ . Then  $\theta_{1,\star}$  is now the minimizer of  $\mathbb{E}_{\pi_1}[\ell(Y|X; \theta)]$  and thus  $s(\boldsymbol{\eta}, \theta)$  in (4.25) must now be defined as

$$s(\boldsymbol{\eta}, \theta) = \sum_{x \in \mathcal{X}} q_1(x) \sum_{y \in \mathcal{Y}} \eta(y|x) \nabla_{\theta} \ell(y|x; \theta)$$

The rest of the proof is unchanged which gives the expressions of  $J_1$  and  $H_{0,1}$  in (4.28) and (4.29), respectively.  $\square$

### 4.3 APPLICATION TO BINARY LOGISTIC REGRESSION

---

To gain further insights into the results summarized in Theorem 4.1 and Proposition 4.3, we consider the example of the logistic regression model with binary labels  $Y$  and input variables  $X$  in  $\mathbb{R}^p$ ; the parameter  $\theta$  is thus  $p$ -dimensional. In this model, the negative log-likelihood function is given by  $\ell(y|x; \theta) = -y\theta^T x + \log(1 + e^{\theta^T x})$ <sup>1</sup>. Thus, the estimation equation which implicitly defines the value of the optimal fit  $\theta_*$  as the value for which  $E_\pi [\nabla_\theta \ell(Y|X; \theta_*)] = 0$  may be rewritten as

$$E_q [X (g(1|X; \theta_*) - \eta(1|X))] = 0 \quad (4.31)$$

Similar direct calculations yield

$$H = E_q [\eta(1|X)(1 - \eta(1|X))XX^T] \quad (4.32)$$

$$I(\theta_*) = E_q [\{\eta(1|X)(1 - \eta(1|X)) + (\eta(1|X) - g(1|X; \theta_*))^2\}XX^T] \quad (4.33)$$

$$J(\theta_*) = E_q [g(1|X; \theta_*)\{1 - g(1|X; \theta_*)\}XX^T] \quad (4.34)$$

$J(\theta_*)$  is the Fisher information matrix traditionally found in logistic regression. Interestingly,  $H$  is recognized as the Fisher information matrix for  $\theta_*$  corresponding to the fully supervised logistic regression model in the well-specified case (i.e. assuming that  $g(y|x; \theta_*) = \eta(y|x)$ ), although we made no such assumption here. Note that, as a consequence, it does not depend on the fitted model and, in particular, on the parameter value  $\theta_*$ .

For the logistic regression, the difference

$$I(\theta_*) - H = E_q [\{\eta(1|X) - g(1|X; \theta_*)\}^2 XX^T]$$

is clearly a term that is all the more significant that the fit achievable by the model is poor. The second important factor that can lead to substantial differences between the asymptotic performances of  $\hat{\theta}_n$  and  $\hat{\theta}_n^s$  is revealed by the following observation: for a given distribution  $\pi$ , the largest (in a matrix sense) achievable value for  $I(\theta_*)$  is given by

$$I(\theta_*) = E_q [\max\{\eta(1|X), 1 - \eta(1|X)\}XX^T]$$

whereas  $H$  in (4.32) may be rewritten as

$$H = E_q [\max\{\eta(1|X), 1 - \eta(1|X)\} \min\{\eta(1|X), 1 - \eta(1|X)\}XX^T]$$

Hence, the difference between  $I(\theta_*)$  and  $H$  can only become very significant in cases where  $\min\{\eta(1|X = x), 1 - \eta(1|X = x)\}$  is small, that is, when the probability of incorrect decision is small, for some values of  $x$ . The overall effect will be all the more significant that this situation happens for many values of  $x$ , or, in other words, that the Bayes error associated with  $\pi$  is small.

---

<sup>1</sup>Or  $\log(1 + e^{-\theta^T yx})$  when the labels are coded as  $\{-1, 1\}$  rather than  $\{0, 1\}$ .

In the presence of the covariate shift, Proposition 4.3 gives the following expressions

$$\begin{aligned} I_{0,1}(\theta_{1,\star}) &= \mathbb{E}_{q_1} \left[ \frac{q_1(X)}{q_0(X)} (\eta(1|X)(1 - \eta(1|X)) + (\eta(1|X) - g(1|X; \theta_{1,\star}))^2) X^T X \right] \\ H_{0,1} &= \mathbb{E}_{q_1} \left[ \frac{q_1(X)}{q_0(X)} (\eta(1|X)(1 - \eta(1|X))) X^T X \right] \\ J_1(\theta_{1,\star}) &= \mathbb{E}_{q_1} [(g(1|X; \theta_{1,\star})(1 - g(1|X; \theta_{1,\star})) X^T X)] \end{aligned}$$

Note that the standard (unweighted) logistic regression estimator is not directly comparable to the other estimators in this case as it converge to  $\theta_{0,\star} = \arg \min_{\theta \in \Theta} \mathbb{E}_{\pi_0} [\ell(Y|X; \theta)]$  rather than to  $\theta_{1,\star}$ . Its asymptotic covariance matrix is defined by  $J_0^{-1}(\theta_{0,\star}) I_0(\theta_{0,\star}) J_0^{-1}(\theta_{0,\star})$ , where

$$\begin{aligned} I_0(\theta_{0,\star}) &= \mathbb{E}_{q_0} \left[ (\eta(1|X)(1 - \eta(1|X)) + (\eta(1|X) - g(1|X; \theta_{0,\star}))^2) X^T X \right], \\ J_0(\theta_{0,\star}) &= \mathbb{E}_{q_0} [g(1|X; \theta_{0,\star})(1 - g(1|X; \theta_{0,\star})) X^T X]. \end{aligned}$$

Of course, if the model is assumed to be well-specified, then  $\theta_{1,\star} = \theta_{0,\star}$  and all estimators can now be compared with the unweighted logistic regression being preferable to the weighted logistic regression and equivalent to the semi-supervised estimator.

In Appendix A we provide the expressions of the asymptotic matrices for the polytomous logistic regression.

## 4.4 CONCLUSIONS

---

We have considered the problem of semi-supervised learning in general and tried to address the problem of semi-supervised learning in the discriminative framework by introducing the marginal  $p(x)$  into the model using an asymptotic perspective. We do not use any prior idea on what type of information is provided by the unlabeled data. The result of Theorem 4.1 provides both proper theoretical support for the claim that the unlabeled data does not matter asymptotically when the model is well-specified and a better understanding of the cases where the unlabeled data does matter. In particular, it confirms the intuition that unlabeled data is most useful when the Bayes error is small. In addition to the asymptotic results, in the next chapter we carry out experiments on an artificial data set, make an attempt to apply the semi-supervised criterion to the real world data, and discuss a number of empirical findings pertaining to logistic regression.

---

## CHAPTER 5

# SEMI-SUPERVISED LEARNING EXPERIMENTS

---

### Contents

---

<b>5.1</b>	<b>A Small Scale Experiment</b>	<b>69</b>
5.1.1	Multinomial Model and Artificial Data	70
5.1.2	Experiments with the Criterion of Bengio-Grandvalet	71
5.1.3	Performance of the Proposed Semi-Supervised Estimator	71
<b>5.2</b>	<b>Text Classification Experiments</b>	<b>74</b>
<b>5.3</b>	<b>Conclusions</b>	<b>75</b>

---

In this chapter, we perform experiments testing the newly introduced asymptotically optimal semi-supervised estimator on a synthetic set (in Section 5.1) and real data set (in Section 5.2). We compare its performance to one of the standard logistic regression, as well as discuss our main observations.

### 5.1 A SMALL SCALE EXPERIMENT

---

We consider here experiments on artificial data, which correspond to the case of binary logistic regression discussed in previous chapter. We focus on a small-scale problem, where it is possible to exactly compute error probabilities and risks so as to completely bypass the empirical evaluation of trained classifiers. This setting makes it possible to obtain an accurate assessment of the performance, as the only source of Monte Carlo error lies in the random selection of the training corpus.

We simulate data in such a way that we can perform experiments with both well-specified and misspecified models. It is well-known that one can simulate data from well-specified logistic models by resorting to a mixture of multinomial distributions.

We consider the case where each observation consists of a vector of  $p = 10$  positive counts which sums to  $d = 3$ . Hence the logistic regression parameter  $\theta$  is ten-dimensional and the set  $\mathcal{X}$  of possible count vectors contains exactly 220 different vectors. We describe

the model in details just below.

### 5.1.1 MULTINOMIAL MODEL AND ARTIFICIAL DATA

The probability mass function of the multinomial distribution is defined as:

$$f(x_1, \dots, x_p; d; \beta_1, \dots, \beta_p) = \begin{cases} \frac{d!}{x_1! \dots x_p!} \beta_1^{x_1} \dots \beta_p^{x_p}, & \text{if } \sum_{i=1}^p x_i = d \\ 0, & \text{otherwise} \end{cases} \quad (5.1)$$

where  $d$  is the number of trials,  $p$  is the number of possible outcomes (each of which has some probability  $\beta_i$ ,  $i = 1, \dots, p$ ),  $\sum_{i=1}^p \beta_i = 1$ ;  $x_i$  is the number of times outcome  $i$  has been seen within  $d$  trials.

The intuition behind our simulated data is as follows. We generate an artificial “document” data set, where each “document” belongs either to class 0 or to class 1. Each “document” contains  $d = 3$  “words”, and the “language” includes  $p = 10$  “words”. According to the formula of combinations with repetitions, the number of possible “documents” is  $\frac{(d+p-1)!}{(p-1)!d!} = 220$ .

Denote by  $\alpha_1$  the prior probability of class 1, and by  $\beta_0$  and  $\beta_1$  the vectors of multinomial parameters. Count vectors  $X$  generated from the mixture of multinomials have marginal probabilities  $q(x) = \alpha_1 \text{mult}(x; \beta_1) + (1 - \alpha_1) \text{mult}(x; \beta_0)$  and conditional probabilities  $P(Y = 1 | X = x) = \{1 + \exp[-(\log \beta_1 - \log \beta_0)^T x + \log \frac{\alpha_1}{1 - \alpha_1}]\}^{-1}$ , where the log is to be understood componentwise. In the following, we take  $\alpha_1 = 0.5$ , i.e., balanced classes, so as to avoid the bias term, that is,  $\log \frac{\alpha_1}{1 - \alpha_1} = 0$ . In order to generate misspecified scenarios, we simply flipped the labels of a few (to be precise, three in the following experiments)  $x$ 's taken among the most likely ones. This label flipping transformation leaves the Bayes error unchanged to that of the underlying unperturbed logistic model but the performance achievable by logistic regression is of course reduced.

#### Evaluation Parameters

Since we simulate our data, and all the parameters are known, we can easily control the Bayes error of the problem, the probability of error, and the logistic loss. The Bayes error is defined as

$$\sum_{x \in X} \min\{\eta(x), 1 - \eta(x)\} q(x). \quad (5.2)$$

We calculate the probability of error in our binary case as follows:

$$\begin{aligned} E(\mathbb{1}\{y = 0, \hat{g}_\theta(x) = 1\} + \mathbb{1}\{y = 1, \hat{g}_\theta(x) = 0\}) = \\ \sum_{x \in X} q(x) (\mathbb{1}\{y = 0, \hat{g}_\theta(x) = 1\} + \mathbb{1}\{y = 1, \hat{g}_\theta(x) = 0\}). \end{aligned}$$

The logarithmic loss for a binary problem takes the form

$$\sum_{x \in X} q(x) \{ \eta(x) \log \ell(y = 1 | x; \theta) + (1 - \eta(x)) \log(1 - \ell(y = 0 | x; \theta)) \}. \quad (5.3)$$

### 5.1.2 EXPERIMENTS WITH THE CRITERION OF BENGIO-GRANDVALET

In the Bengio–Grandvalet criterion already presented as equation (4.5) we replace empirical average over unlabeled  $x_i$ s by expectation computed under  $q$ . We aim at minimizing the negative log-likelihood:

$$\sum_{i=1}^n \sum_{y \in \mathcal{Y}} -\ell(y|X_i; \theta) \mathbb{1}\{Y_i = y\} + \rho_{\text{BG}} \sum_{x \in \mathcal{X}} \left( \sum_{y \in \mathcal{Y}} -\ell(y|x; \theta) L(y|x; \theta) \right) q(x). \quad (5.4)$$

We applied the criterion of Bengio–Grandvalet to the artificial data. In our experiments, the optimal value of  $\rho_{\text{BG}}$  that controls the impact of the unlabeled data, has been chosen by cross validation and equals  $\rho_{\text{BG}} = 0.001$ . If  $\rho_{\text{BG}}$  is large ( $> 0.01$ ), that is we let unlabeled data influence the estimation significantly, the performance tends to drop. The criterion of Bengio–Grandvalet is not convex, therefore there are local minima and the necessity to choose well the initial point for  $\theta$ .

Several experiments on the simulated data lead to the following conclusions:

- The Bengio–Grandvalet method is sensitive to the parameter initialization. So, if we provide it with initial values that are close to  $\hat{\theta}_{ML}$ , its convergence is faster and the optimized values of parameters are more appropriate than if we initialize it randomly.
- The value  $\rho_{\text{BG}}$  has to be well chosen, otherwise the generalization performance is significantly worse than that of logistic regression.
- We noticed that the method has difficulties (stability problems) in the interval of very small  $n$  values ( $n = 10, 20, 30$ ).

In our experiments, the use of the entropy regularization did not warrant improved results, even in cases where the Bayes error was particularly low.

### 5.1.3 PERFORMANCE OF THE PROPOSED SEMI-SUPERVISED ESTIMATOR

In this section, we provide the comparative performance on the synthetic data of the asymptotically optimal estimator, described in Chapter 4. Figures 5.1 and 5.2 correspond to a case where the underlying unperturbed logistic model has a Bayes error of 1.7% and the probability of error associated with the best fitting logistic model is of 9.4%. Remember that in these figures, the only source of randomness is due to the choice of the training sample, which is repeated 1000 times independently for each size of the training sample, from  $n = 10$  to  $n = 5000$  observations.

As logistic regression is very sensitive to the use of regularization for small sample sizes (here, when  $n$  is less than one thousand), both (4.15) and (4.18) were regularized by adding a  $L_2$  penalty term of the form  $\rho_n \|\theta\|_2^2$ , where  $\rho_n$  has been calibrated independently for each value of  $n$ . This being said, the optimal regularization parameter was always



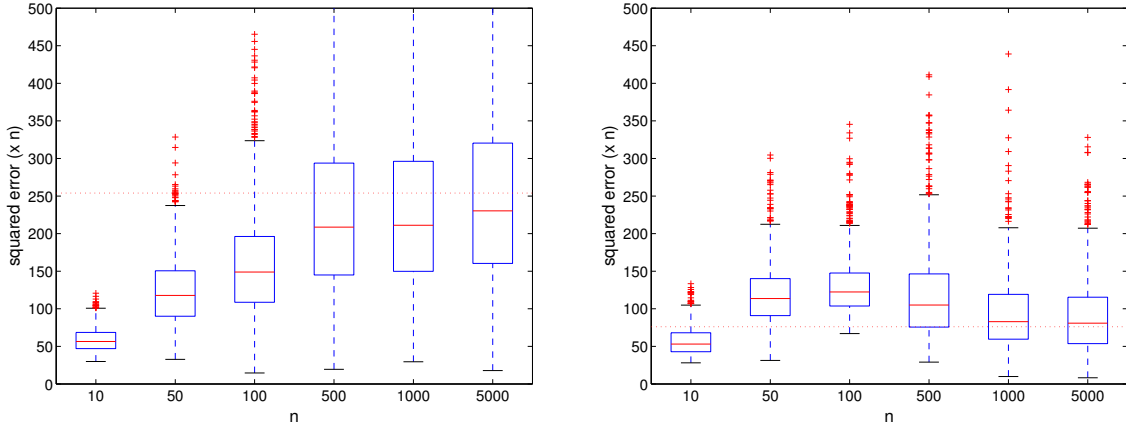


Figure 5.1: Boxplots of the scaled squared parameter estimation error as a function of the number of observations. Left: for logistic regression,  $n\|\hat{\theta}_n - \theta_\star\|^2$ ; right: for the semi-supervised estimator,  $n\|\hat{\theta}_n^s - \theta_\star\|^2$ .

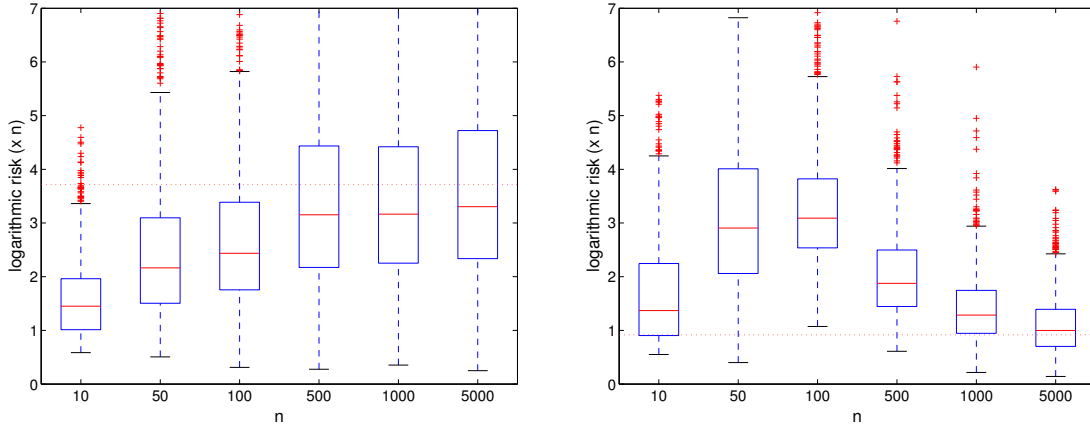


Figure 5.2: Boxplots of the scaled excess logarithmic risk as a function of the number of observations. Left: for logistic regression,  $n(\mathbb{E}_\pi[\ell(Y|X; \hat{\theta}_n)] - \mathbb{E}_\pi[\ell(Y|X; \theta_\star)])$ ; right: for the semi-supervised estimator,  $n(\mathbb{E}_\pi[\ell(Y|X; \hat{\theta}_n^s)] - \mathbb{E}_\pi[\ell(Y|X; \theta_\star)])$ .

found to be within a factor 2 of  $\rho_n = 1/n$  for (4.15) and  $\rho_n = \frac{1}{n} \sum_{\{x: \sum_{i=1}^n \mathbb{1}_{\{X_i=x\}} > 0\}} q(x)$  for (4.18). The effect of regularization is also negligible for the two rightmost boxplots in each graph (i.e., when  $n$  is greater than 1000). On Figures 5.1 and 5.2, the superimposed horizontal dashed lines correspond to the theoretical averages computed from Theorem 4.1 and Corollary 4.2, respectively.

Notice that the squared error and the logarithmic risk are scaled by  $n$ , since both values decrease at speed  $1/n$ .

When  $n$  is larger than one thousand, Figures 5.1 and 5.2 perfectly correlate with the theory which predicts some advantage for the semi-supervised estimator as we are considering a case where the Bayes error is small and the model misspecification is significant. For large values of  $n$ , the semi-supervised estimator not only achieves better average performance but also does so more constantly, with a reduced variability. For smaller values of  $n$ , the picture is more contrasted, particularly when  $n$  ranges from 50 to 100 where the

semi-supervised estimator may perform comparatively worse than the logistic regression. In this example, in terms of the probability of error, the semi-supervised estimator performs marginally better than logistic regression when  $n = 10$  and  $n = 5000$  (although the difference is bound to be very small in the latter case) and somewhat worse in between.

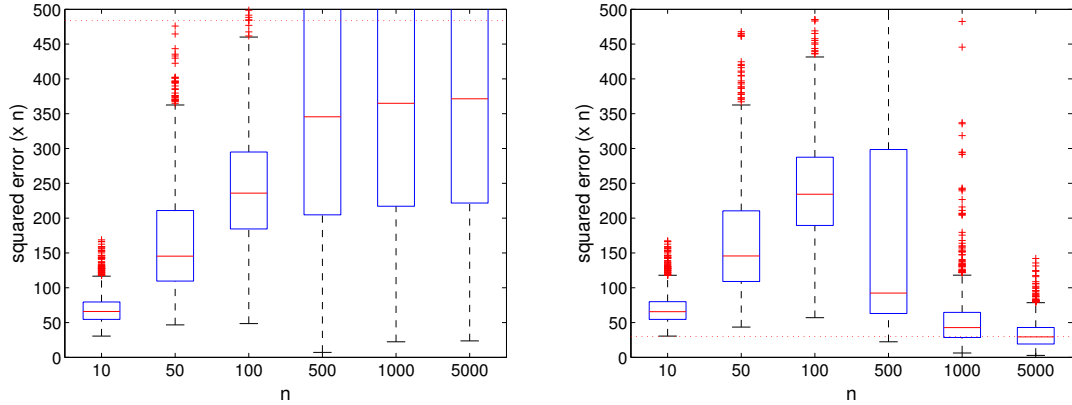


Figure 5.3: Boxplots of the scaled squared parameter estimation error as a function of the number of observations for the case with the covariate shift. Left: for the logistic regression,  $n\|\hat{\theta}_n - \theta_\star\|^2$ ; right: for the semi-supervised estimator,  $n\|\hat{\theta}_n^s - \theta_\star\|^2$ .

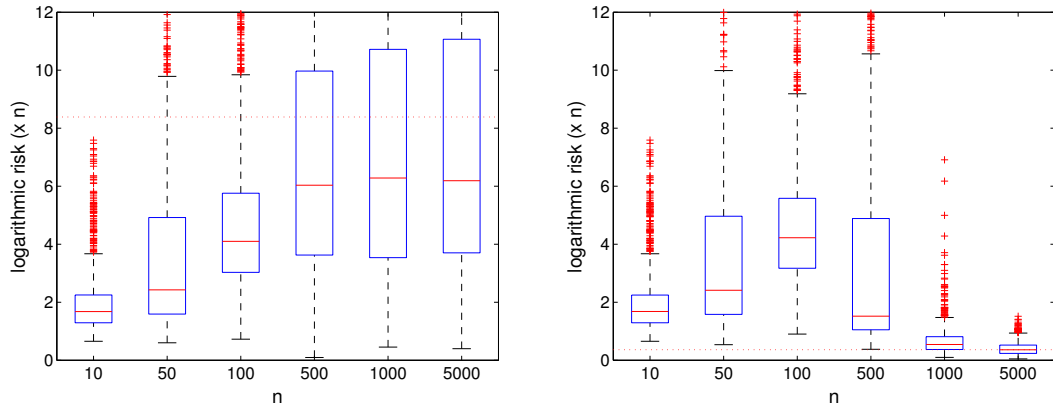


Figure 5.4: Boxplots of the scaled excess logarithmic risk as a function of the number of observations for the case with a covariate shift. Left: for the logistic regression,  $n(\mathbb{E}_\pi[\ell(Y|X; \hat{\theta}_n)] - \mathbb{E}_\pi[\ell(Y|X; \theta_\star)])$ ; right: for the semi-supervised estimator,  $n(\mathbb{E}_\pi[\ell(Y|X; \hat{\theta}_n^s)] - \mathbb{E}_\pi[\ell(Y|X; \theta_\star)])$ .

Figures 5.3 and 5.4 correspond to the case of a covariate shift. In the experiments,  $q_0(x)$  that is the training distribution, is uniform,  $q_0(x) = 1/220, \forall x$ . The test distribution  $q_1(x)$  is the same as described in Section 5.1.1. The dashed horizontal line corresponds to the theoretical values computed according to Theorem 4.1 and Proposition 4.3.

As expected, the difference between both approaches for large values of  $n$  decreases for scenarios with larger error probabilities. In those scenarios, the semi-supervised estimator performs worse than logistic regression for smaller values of  $n$  and equivalently for large values of  $n$ . A finding of interest is the fact that for well-specified models (i.e., with data generated from a multinomial mixture model) with low Bayes error, the semi-supervised approach does perform better than logistic regression, for small values of  $n$ . This effect

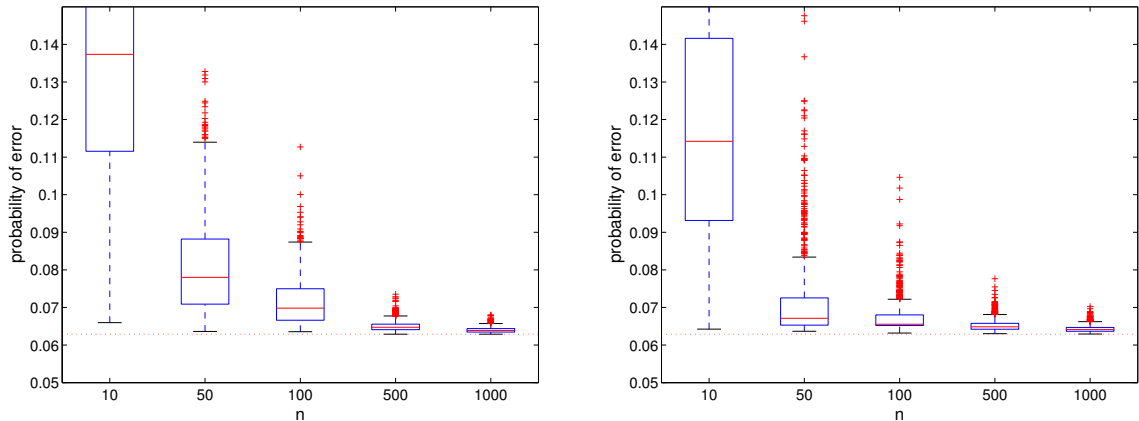


Figure 5.5: Boxplots of the probability of error as a function of the number of observations for a well-specified model. Left: for the logistic regression; right: for the semi-supervised estimator.

can be significant even when considering the probability of error of the trained classifiers, as exemplified on Figure 5.5 in a case where the Bayes error is 6.3%. This observation is promising and deserves further investigation as the analysis of Section 4.2.1 only explains the behavior observed for large values of  $n$ , which in the case of well-specified models results in the two approaches being equivalent.

## 5.2 TEXT CLASSIFICATION EXPERIMENTS

---

To evaluate our methodology on a more realistic test bed, we have used a simple binary classification task, consisting in classifying mails as spam or ham based on their textual content (SpamAssassin corpus), and the word phonetisation task (Nettalk corpus).

The corpus used is the SpamAssassin corpus (Mason, 2002), which contains approximately 6,000 documents. The error rate on the test data is approximately 3% using the standard logistic regression. Adapting our technique to real-world data requires to provide an estimate for the marginal  $q(x)$ . The space of  $\mathcal{X}$  is too large (about 1,500 words in the corpus dictionary) to estimate  $q(x)$  as it is done for the simulated data. It was then carried out by performing a discrete quantification of the data vectors as follows. We first use unsupervised clustering techniques to partition the available unlabeled collection of documents in  $k$  clusters. More specifically, we used a mixture of multinomial model as in (Nigam et al., 2000, Rigouste et al., 2007) with  $k = 10$  components. We then simply adapt (4.18) by replacing  $q(X_i)$  by the empirical frequency of the cluster to which  $X_i$  belongs, likewise the denominator  $\sum_{j=1}^n \mathbb{1}\{X_j = X_i\}$  is replaced by the number of training documents belonging to the same cluster as  $X_i$ . We believe that this methodology is very general and makes the proposed approach applicable to a large variety of data. In effect, observations belonging to clusters which are underrepresented in the training corpus have higher relative weights, while the converse is true for observations belonging to overrepresented clusters. Note that, at this stage, no attempts have been made at tuning the number  $k$  of clusters, although the intuition suggests that it would probably be reasonable

to increase  $k$  (slowly) with  $n$ .

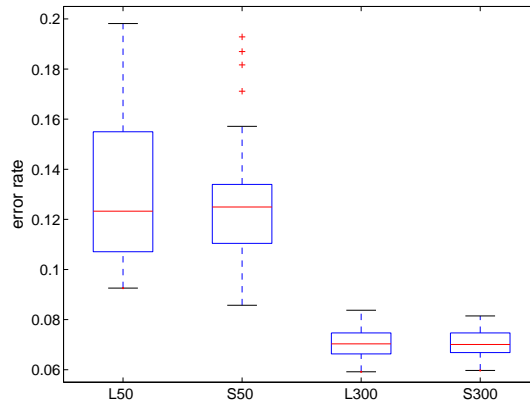


Figure 5.6: Boxplots of the error rates for, L50: logistic regression with  $n = 50$ ; S50: semi-supervised estimator with  $n = 50$ ; L300 and S300, idem with  $n = 300$ .

We tested the method with  $n = 50$  and  $n = 300$  randomly chosen training documents, the remaining mails serving as the test set; each trial gave rise to 50 Monte Carlo replications. For each value of  $n$ , the best regularization parameter was determined experimentally both for the usual logistic regression and the semi-supervised estimator. Each document is here represented as a count vector of dimension 1,500. The resulting error rates are plotted as boxplots on Figure 5.6. Although the difference between both methods is certainly not very significant in this preliminary experiment, we note that, as in the simple case of Section 5.1, the semi-supervised estimator provides a more stable performance when  $n$  is small.

### 5.3 CONCLUSIONS

---

We carried out a number of experiments with the criterion proposed in the previous chapter. The advantage of the proposed method is that it does not compromise the simplicity of the maximum likelihood approach because the weighted semi-supervised criterion stays convex. In addition, one could incorporate prior knowledge as used in other semi-supervised approaches: for instance the “cluster assumption” can be implemented by modifying (4.17) so as to incorporate a Bayesian prior that connects conditional probabilities for neighboring values of the input vector. In Section 5.2, we suggested a means by which the method can be extended to larger scales problem, including applications in which the feature vector is either continuous or has a more complex structure.

On the real data set the performance of the semi-supervised criterion is close to the performance of the standard logistic regression and the difference is hardly distinguishable. We explain it as follows. The asymptotic advantage of the semi-supervised approach can be observed only when considering the scaled excess logarithmic risk or the scaled squared error. The computations of the excess risk and the squared error involve the knowledge of the true distribution. In the case of any real data we do not know the optimal parameter values and thus neither the excess logarithmic risk, nor the squared

error are available. More generally, the fact that the observed differences are mostly significant in the asymptotic regime suggests that the approach has a limited potential for typical semi-supervised settings in machine learning applications.

The extension of the proposed approach to the case of sequence labeling with conditional random fields is still an open issue.

The experiments have illustrated another open problem, that is the theoretical analysis of the behavior of the proposed criterion when  $n$  is small, which cannot be deduced from the asymptotic analysis presented here.

## Part III

# $L_1$ Norm Based Model Selection in Discriminative Models



---

## CHAPTER 6

# SPARSITY AND MODEL SELECTION IN DISCRIMINATIVE MODELS

---

### Contents

---

<b>6.1 Empirical Study: Sparsity in Conditional Random Fields . . .</b>	<b>80</b>
6.1.1 How Many Features Can Be Eliminated? . . . . .	80
6.1.2 Most Influential Features . . . . .	83
<b>6.2 Brief Overview of Feature Selection Techniques . . . . .</b>	<b>85</b>
6.2.1 Naive Model Selection Methods for CRFs . . . . .	85
6.2.2 Heuristic Approaches Applied to CRFs . . . . .	86
6.2.3 Penalty Terms Including the $L_1$ Norm . . . . .	86
<b>6.3 Numerical Optimization of Criteria Including the <math>L_1</math> Norm . .</b>	<b>88</b>
6.3.1 Orthant-Wise Limited-Memory Quasi-Newton . . . . .	89
6.3.2 Coordinate-Wise Descent . . . . .	90
<b>6.4 Conclusions . . . . .</b>	<b>92</b>

---

Conditional random fields, considered in Chapter 3, constitute a popular and effective approach for supervised structure learning tasks involving the mapping between complex objects such as strings and trees. An important property of CRFs is their ability to cope with large and redundant feature sets and to integrate some form of structural dependency between output labels.

The dependencies in conditional random fields are extracted according to pre-defined patterns. The number of parameters to be estimated can be very large. Do we need all of them? Is there any sparsity of the model and, if yes, can we exploit it to speed up the training and inference procedures?

In this chapter, we illustrate on real world applications in the domain of natural language processing that sparsity patterns do exist and we can hope to obtain a model that is sparse and interpretable in the sense that irrelevant features have zero values. We start with some simple heuristic methods which result in sparse models but whose performance is worse than the accuracy of state-of-the-art approaches based on  $L_1$  penalization. We investigate the combination of the  $L_1$  and  $L_2$  norms known as elastic net.



## 6.1 EMPIRICAL STUDY: SPARSITY IN CONDITIONAL RANDOM FIELDS

---

In this section, we illustrate the natural sparsity of the data sets already considered in Chapter 3. We perform experiments on Nettetalk, CoNLL 2000 and CoNLL 2003 data and therefore motivate the need to perform model selection.

In the following, we use CRFs that involve two types of feature functions, unigram  $\mu_{y,x}$  and bigram  $\lambda_{y',y,x}$  which we index as follows for the Nettetalk corpus:

$$\sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t) = \sum_{y \in Y, x \in X} \mu_{y,x} \mathbb{1}\{y_t = y, x_t = x\} + \sum_{(y',y) \in Y^2, x \in X} \lambda_{y',y,x} \mathbb{1}\{y_{t-1} = y', y_t = y, x_t = x\}, \quad (6.1)$$

where  $X = \{\text{letters}\}$  and as follows for the CoNLL data sets

$$\sum_{k=1}^K \theta_k f_k(y_{t-1}, y_t, x_t) = \sum_{X \in \mathcal{X}} \left( \sum_{y \in Y, x \in X} \mu_{y,x} \mathbb{1}\{y_t = y, x_t = x\} + \sum_{(y',y) \in Y^2, x \in X} \lambda_{y',y,x} \mathbb{1}\{y_{t-1} = y', y_t = y, x_t = x\} \right), \quad (6.2)$$

where

$$\mathcal{X} = \{\text{words, POS tags}\}$$

for the CoNLL 2000 corpus, and

$$\mathcal{X} = \{\text{words, POS tags, syntactic chunks}\}$$

for CoNLL 2003 set.

### 6.1.1 HOW MANY FEATURES CAN BE ELIMINATED?

We carry out training of the  $L_2$ -penalized CRF criterion with the feature set described above. For each corpus, the regularization parameter is chosen by cross validation. Tables 6.1, 6.2, and 6.3 illustrate the sparsity for CoNLL 2000, CoNLL 2003, and Nettetalk data respectively.

#### CoNLL Data Sets

For CoNLL 2000 and CoNLL 2003 we take all types of observations into account, that is words and part of speech tags for CoNLL 2000, and words, their part of speech tags, and syntactic tags for CoNLL 2003. We know that parameters estimated with the  $L_2$  penalty term are never sparse, however, a large number of values is close to zero. After parameter

Interval feat. set to 0	Nb. of active features	Nb. of active features %	Accuracy	Precision	Recall	F
$\emptyset$	9,266,269	100	94.43	91.34	90.98	91.16
$[-0.25 \ 0.25]$	26,986	0.29	94.42	91.34	91.02	91.18
$[-0.5 \ 0.5]$	12,127	0.13	94.35	91.11	90.83	90.97
$[-0.75 \ 0.75]$	5,457	0.06	93.82	90.46	89.61	90.03
$[-1 \ 1]$	2,079	0.02	93.02	89.14	87.93	88.53

Table 6.1: Empirical study of sparsity patterns (CoNLL 2000 Corpus, English) . Dependencies  $\lambda_{y',y,x^j}$ ,  $\mu_{y,x^j}$ ,  $j \in \{1,2\}$ .

estimation, we consequently set to zero a number of parameters whose estimated values lie in the interval centered at zero.

Tables 6.1 and 6.2 illustrate that for the CoNLL 2000 and CoNLL 2003 data, we can set 99.99% of feature parameters to zero and still achieve baseline performance (see Section 3.5.2 for the values of baseline performance), and about 90% of parameters of the initial full model can be deleted without degrading performance. We noticed empirically that after training of the  $L_2$ -penalized criterion the positive feature values correspond to observed patterns. The never observed features (negative examples) correspond to negative values.

## Nettalk Corpus

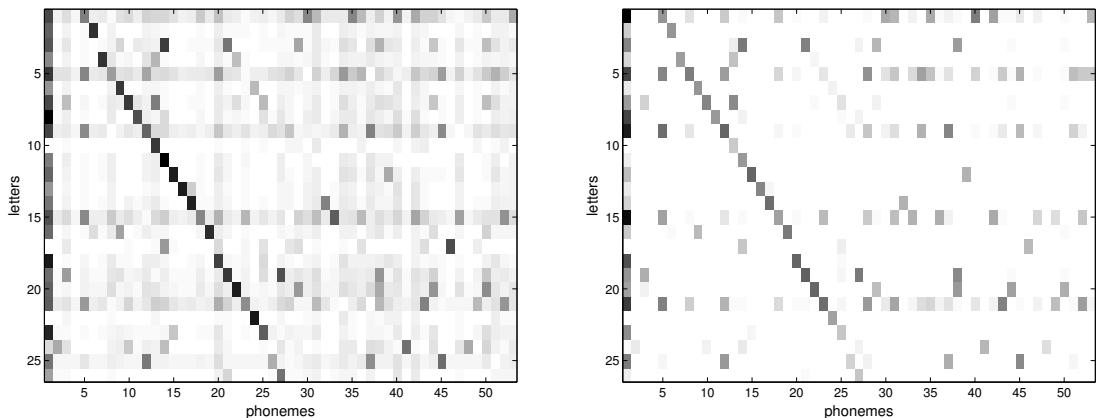


Figure 6.1:  $L_1$  norm of the parameters estimated with standard  $L_2$ -regularized maximum likelihood for the Nettalk task. Left:  $|\mu_{y,x}|$  for the 53 phonemes  $y$  and 26 letters  $x$ . Right:  $\sum_{y'} |\lambda_{y',y,x}|$  for the 53 phonemes  $y$  and 26 letters  $x$ .

Figure 6.1 displays the sparsity of the parameter vectors (Nettalk data) obtained with the  $L_2$ -regularized maximum likelihood approach. Sparsity is especially striking in the case of the bigram parameters  $\lambda_{y',y,x}$  which are, by far, the most numerous ( $53^2 \times 26$ ). Another observation is that this sparsity pattern is quite correlated to the corresponding value of  $|\mu_{y,x}|$ : in other words, most sequential dependencies  $\lambda_{y',y,x}$  are only significant when the associated marginal factor  $\mu_{y,x}$  is. This suggests to take a closer look at the internal structure of the feature set.

Interval	Number of Active Feat.	Number of Active Feat. in %	Test A						Test B		
			Acc.	Prec.	Recall	FBI	Acc.	Prec.	Recall	FBI	
$\emptyset$	1,611,832	100	96.96	85.42	80.78	83.04	94.69	75.17	71.23	73.15	
$[-0.1 \ 0.1]$	97,790	6.07	96.95	85.36	80.72	82.97	94.69	75.19	71.28	73.18	
$[-0.25 \ 0.25]$	62,563	3.88	96.81	84.65	79.94	82.23	94.55	74.84	70.71	72.72	
$[-0.5 \ 0.5]$	38,520	2.39	96.67	84.03	79.06	81.47	94.45	74.75	70.21	72.41	
$[-0.75 \ 0.75]$	26,038	1.62	96.5	83.49	78.11	80.71	94.22	74.64	68.70	71.54	
$[-1 \ 1]$	19,244	1.19	96.18	86.11	75.24	80.31	94.0	78.65	65.67	71.58	

Table 6.2: Empirical study of sparsity patterns (CoNLL 2003 Corpus, English). Dependencies  $\lambda_{y^i, x^j}$ ,  $\mu_{y, x^j}$ ,  $j \in \{1, 2, 3\}$ .

Interval feat. set to 0	Number of active features	Number of active features in %	Error
$\emptyset$	75,790	100	13.98%
$[-0.25 \ 0.25]$	5,446	7.2	14.02%
$[-0.5 \ 0.5]$	4,005	5.3	14.11%
$[-1 \ 1]$	2,585	3.4	14.91%
$[-2 \ 2]$	1,288	1.7	17.08%
$\leq 0$	1,697	2.2	17.49%

Table 6.3: Empirical study of sparsity patterns (Nettalk). Dependencies  $\lambda_{y',y,x}$ ,  $\mu_{y,x}$ .

In the Nettalk experiments, we tried to eliminate all negative features, and estimate the error rate keeping positive features only (see the last result in the Table 6.3). We came to the conclusion that never observed configurations, the so-called negative instances are equally important as observed, the so-called positive examples. The performance degradation is significant, with an error rate of 17.49%, compared to the initial error rate of 13.98% when using all features.

## 6.1.2 MOST INFLUENTIAL FEATURES

What kind of dependencies are still active after our naive screening? Since the dependencies  $\mu_{y,x}$  and  $\lambda_{y',y,x}$  are redundant and in some sense hierarchical, one can imagine two scenarios. The bigram feature  $\lambda_{y',y,x}$  is there only if the corresponding unigram  $\mu_{y,x}$  feature is active. The second intuition is based on their redundancy. It is not necessary that both of them survive. If  $\lambda_{y',y,x}$  is active,  $\mu_{y,x}$  is not informative anymore.

### Nettalk Corpus

We performed training with two types of features  $\lambda_{y',y,x}$  and  $\mu_{y,x}$ . On Figure 6.2 (Nettalk data), one can see the dependency of the number of active  $\lambda_{y',y,x}$  features to their corresponding  $\mu_{y,x}$ .

- Figure 6.2 on the left represents the case when parameters from the interval  $[-0.25 \ 0.25]$  are set to 0. There are 5,446 parameters with non zero values. We have 949 unigram and 4,497 of type bigram features that are not zeroed. Among 4,497 bigram features there are 83 (2%) bigram dependencies (marked with the green color) whose corresponding unigram parameters are set to 0.
- Figure 6.2 on the right shows the case when parameters from the interval  $[-2 \ 2]$  are set to 0. We get 1,288 not zeroed parameters, among them 161 unigram, and 1,127 bigram. Notice that the number of bigram dependencies with associated unigram features set to 0 is 112 (10%) and is not negligible (these features are highlighted with the green color).

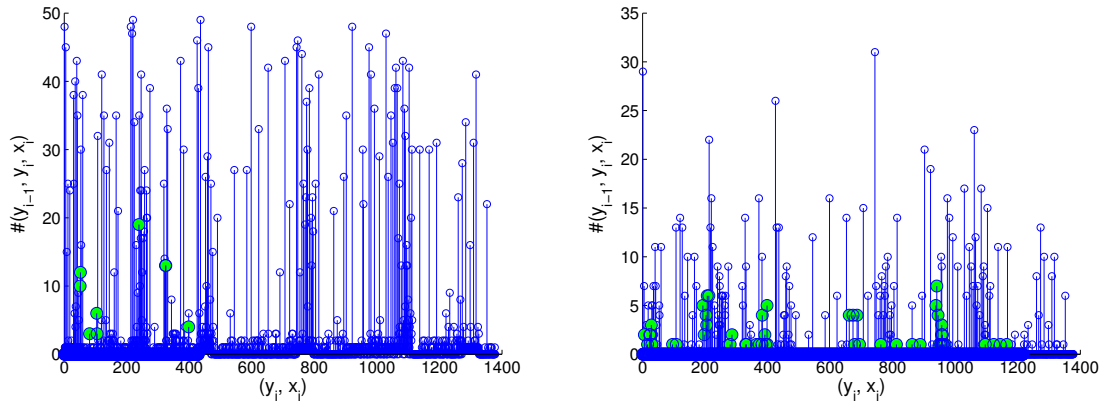


Figure 6.2: Nettetalk. Number of active features  $(y', y, x)$  for every possible  $(y, x)$  dependency. Left: features from the interval  $(-\infty, -0.25], [0.25, +\infty)$ . Right: features from the interval  $(-\infty, -2], [2, +\infty)$ .

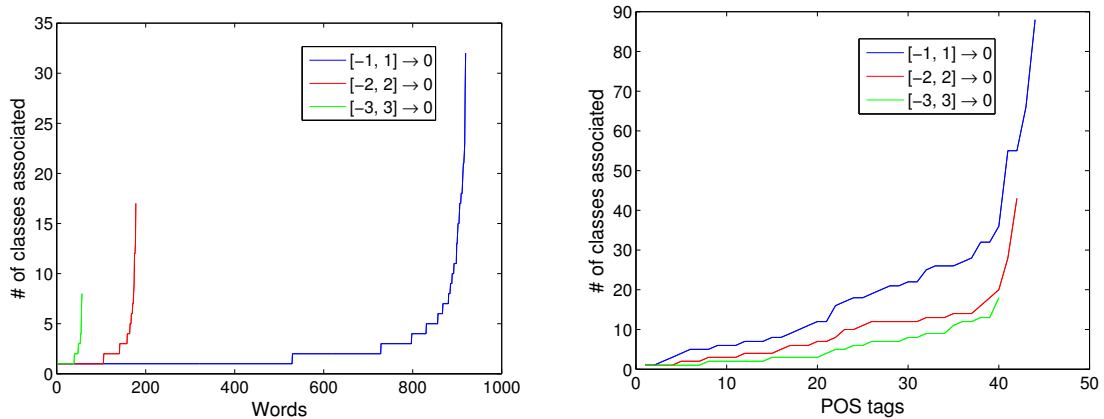


Figure 6.3: CoNLL 2000. Left: The number of dependencies based on words which are active with performance being close to the baseline. Right: the number of dependencies based on POS tags that are active with performance being close to the baseline.

There exists a hierarchical link between redundant features, here, in particular, between  $\lambda_{y',y,x}$  and  $\mu_{y,x}$ . In other words, we can find parameters  $\mu_{y,x}$  that can be set to zero, i.e. eliminated from the model with their corresponding  $\lambda_{y',y,x}$  without degrading performance.

### CoNLL Data Sets

It is less easy to visualize and analyze parameter values of CoNLL 2000 and CoNLL 2003 corpora, since there are more types of feature functions and therefore many more parameters than in the Nettetalk corpus. Figure 6.3 displays the number of active unigram parameters for every observation. Figure 6.3 on the left shows how numerous are the unigram features based on words. On the right, we display the same dependency for features based on part-of-speech tags. The plots contain three curves, each of these corresponds to an interval from which parameters are zeroed,  $[-1, 1]$ ,  $[-2, 2]$ ,  $[-3, 3]$ . The parameter values are sorted according to the number of features active for observations. We see that increasing the interval, the model becomes more and more deterministic, e.g., eliminating

parameters in the interval  $[-3\ 3]$ , we get about 500 words for which only one unigram parameter is not zeroed, and therefore there is the only one possible label (from 23).

Note that setting parameter values from the interval  $[-1\ 1]$  to 0, there are 2,970 active features (2,119 that are based on words, and 851 based on POS tags); setting parameters from the interval  $[-2\ 2]$  to 0, we get 762 values different from 0 (370 for words, and 392 for POS tags), for the baseline performance (the interval  $[-3\ 3]$ ), we have only 310 parameters (95 for words, and 215 for POS tags). Note that originally there are many more word/label features than POS tag/label features. If the goal is to reach a better performance with the least possible number of parameters, the trend seems to be very clear: POS tags/label features are more numerous after severe elimination, since these dependencies are more informative than the ones based on lexical items.

One would expect to attain the high classification accuracy with a much reduced set of feature functions using an appropriate feature selection approach. It is encouraging to know that there are a lot of irrelevant dependencies that can be deleted, since it is feasible to implement a method that would be able to choose the vital features itself. We can also hope to integrate many more and richer features into an initial model.

## 6.2 BRIEF OVERVIEW OF FEATURE SELECTION TECHNIQUES

---

In this section, we briefly consider the approaches and optimization methods to produce sparse models. We explore algorithms applied to the least squares, logistic regression, and conditional random fields.

### 6.2.1 NAIVE MODEL SELECTION METHODS FOR CRFS

The most naive approach for model selection is probably to train a model that is not sparse, and eliminate some dependencies a posteriori, e.g. features whose values are not of sufficient magnitude, as we have done in the previous section to motivate the sparsity of CRF model applied to various natural language processing tasks.

Another simple and not necessarily specific for CRFs heuristic approach used, e.g., in (Toutanova and Manning, 2000) consists in getting rid of rare features a priori. We will refer to this method as to “cut-off”, since it cuts off all the dependencies whose frequencies are smaller than some provided threshold.

Pre-selection of features based on their frequency is not the only possible way. Pre-selection can be based on mutual information, see, e.g. (Yang and Pedersen, 1997). Mutual information or information gain between two discrete random variables  $z$  and  $v$ ,  $\mathbb{1}\{Y = y\} = z$ ,  $\mathbb{1}\{Y \neq y\} = z'$  and  $\mathbb{1}\{X = x\} = v$ ,  $\mathbb{1}\{X \neq x\} = v'$  is defined as follows and can

be directly applied to the features  $\mu_{y,x}$ , in other words, to the unigram features:

$$\begin{aligned}
 I(z, v) = & \sum_{v,z} p(v, z) \log \frac{p(v, z)}{p(v)p(z)} + \sum_{v,z'} p(v, z') \log \frac{p(v, z')}{p(v)p(z')} + \\
 & \sum_{v',z} p(v', z) \log \frac{p(v', z)}{p(v')p(z)} + \sum_{v',z'} p(v', z') \log \frac{p(v', z')}{p(v')p(z')}. \quad (6.3)
 \end{aligned}$$

For the bigram features, we choose to compute the mutual information between the variables  $\mathbb{1}\{Y = y\} = z$  and  $\mathbb{1}\{Y' = y', X = x\} = v$  following the idea that we predict  $y$  given  $y'$  and  $x$ . In this case the equation (6.3) is applicable for bigram features as well.

As we will see in Section 7.3.3, such naive heuristics do not achieve a reasonable accuracy on test data, especially in cases where we want to keep very few active features.

## 6.2.2 HEURISTIC APPROACHES APPLIED TO CRFs

To our knowledge, McCallum (2003) made the first attempt to perform model selection for conditional random fields. The approach was mainly motivated by Della Pietra et al. (1997) and is based on a greedy algorithm which selects features with respect to their impact on the log-likelihood function. Related ideas also appear in (Dietterich et al., 2004).

Cohn (2006) makes another kind of approximation and considers “generalized” feature functions: rather than making each feature function depend on a specific value of the label (or on specific values of label pairs), the author introduces functions that only depend on subsets of (pairs of) labels. This amounts to introducing tying between some parameter values, a property that can then be used to speed-up the forward-backward procedure during training. This technique allows to considerably reduce the training time, with virtually no loss in accuracy. The algorithm relies on a decomposition of the clique potential into two terms, the first has a linear complexity (with respect to the number of labels), and the other is sparse. This idea was already present in (Siddiqi and Moore, 2005). This method however requires to a priori specify the tying pattern.

## 6.2.3 PENALTY TERMS INCLUDING THE $L_1$ NORM

Some kind of penalty, e.g. the  $L_2$  norm, is essential to estimate a model that generalizes well to unseen data. Penalizing approaches concern either the dimensionality of the model, or values of parameters. Examples of approaches penalizing the dimensionality of the model are e.g., AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion), introduced by Akaike (1973) and Schwartz (1978) respectively. The norm penalizing techniques on the contrary, impose penalty on the values of parameters. The  $L_2$  norm has been a widely used penalty term for years, as it performs well and does not violate convexity of a criterion. In the following, we refer to  $\rho_2$  as the regularization parameter associated with the  $L_2$  norm, and  $\rho_1$  with the  $L_1$  penalty term.

An important advantage of the  $L_2$  penalization is that the penalized objective function remains convex and differentiable everywhere, e.g., the least squares criterion penalized

by the  $L_2$  norm

$$\hat{\theta}^{\text{ridge}} = \arg \min_{\theta} \sum_{i=1}^N \left( y_i - \sum_{j=1}^p x_{ij} \theta_j \right)^2 + \rho_2 \sum_{j=1}^p \theta_j^2, \quad (6.4)$$

which is called ridge regression.

The  $L_1$  regularizer for the least squares criterion

$$\hat{\theta}^{\text{lasso}} = \arg \min_{\theta} \sum_{i=1}^N \left( y_i - \sum_{j=1}^p x_{ij} \theta_j \right)^2 + \rho_1 \sum_{j=1}^p |\theta_j|, \quad (6.5)$$

was introduced in (Tibshirani, 1996). The  $L_1$  penalization for the least squares criterion is known as well under another name, lasso (least absolute shrinkage and selection operator). It produces a sparse vector of parameters which is interpretable and that makes a structure of the model more clear.

The  $L_2$  penalty term pushes the parameter values towards zero, hence less important features correspond to values that are close to zero but they are never zeroed. The property of the  $L_1$  penalty is to produce a vector of parameters which contains a lot of zeros. Setting a parameter to zero corresponds to excluding the corresponding dependency from the model.

The major disadvantage of the  $L_1$  regularizer is that although the penalized criterion is still convex (Boyd and Vandenberghe, 2004), it is not differentiable at zero. Therefore, numerical gradient-based optimization methods cannot be applied directly. A number of approaches have been recently proposed to optimize the  $L_1$  penalized criterion and various penalties including the  $L_1$  norm. We consider some of optimization methods in Section 6.3.

## Elastic Net

Some limitations of the  $L_1$ -penalized criterion have been empirically observed. Zou and Hastie (2005) reported that in a case of highly correlated variables the  $L_1$ -penalized criterion tends to select more or less randomly one variable in a group of correlated parameters. In applications considered in this thesis, the parameters are correlated, some of them are highly correlated, since the feature functions are even redundant. Tibshirani (1996) observed that the performance of the  $L_1$ -penalized least squares is dominated by ridge regression in such situations. Taking these remarks into account, a new regularization technique called elastic net has been proposed.

The elastic net penalty (introduced in (Zou and Hastie, 2005), considered in details in (Friedman et al., 2008)) is a compromise between the  $L_2$  norm and the  $L_1$  norm penalties. The use of both types of penalty terms seems preferable in log-linear conditional models, as it makes it possible to control both the number of non zero coefficients (through  $\rho_1$ ) and to avoid the numerical problems that might occur in large dimensional parameter settings if the magnitude of the  $\theta_k$ s is not sufficiently constrained by the penalty. The elastic net criterion is defined as

$$\ell(\mathcal{D}; \theta) + P_{\rho_1, \rho_2}(\theta),$$

where

$$P_{\rho_1, \rho_2}(\theta) = \frac{\rho_2}{2} \|\theta\|_2^2 + \rho_1 \|\theta\|_1 = \sum_{j=1}^p \left( \frac{\rho_2}{2} \theta_j^2 + \rho_1 |\theta_j| \right).$$



For the case of the least squares criterion, if we set  $\rho_2 = 0$ , we get the pure lasso, otherwise, if we set  $\rho_1 = 0$ , the criterion is the same as the ridge regression.

Note that Zou and Hastie (2005) impose the reparameterisation

$$\rho_1 = \gamma\alpha \quad \rho_2 = \gamma(1 - \alpha), \quad (6.6)$$

where  $0 \leq \alpha \leq 1$  controls the ratio between the  $L_1$  and  $L_2$  penalty terms. We do not apply the condition (6.6) to our approach, described in Chapter 7. We fix the value of  $\rho_2$  and examine the sparsity impact of the parameter  $\rho_1$ .

### Group and Hierarchical Structure of Data

The motivation for the group lasso is to select whole blocks of variables, rather than isolated variables. The group lasso estimator, introduced in (Yuan and Lin, 2005), is an extension of the lasso. Its distinctive property is the capability to perform variable selection at the group level, where either all the variables in a group are selected or all the variables in a group are set to zero. The group lasso criterion is defined as

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{2} \sum_{i=1}^N \left( y_i - \sum_{i=1}^p x_{ij} \theta_j \right)^2 + \rho \sum_{g=1}^G \|\theta_{\mathcal{I}_g}\|_2, \quad (6.7)$$

where  $\mathcal{I}_g$  is the index set belonging to the  $g$ th group of variables,  $g = 1, \dots, G$ . Meier et al. (2008) have extended the group lasso to the logistic regression.

For cases where prior information is available, not only on groups of variables but also on a hierarchical structure of the variables, Zhao et al. (2009) proposed an approach which develops the idea of the enclosed norms that create the effect of a certain hierarchy. The approach is called Composite Absolute Penalties (CAP). Hierarchical penalization that combines lasso and group lasso has been introduced by Szafranski et al. (2007). However, the composite and group penalties can be applied only in cases where a prior hierarchy exists. In cases when the hierarchy and group structure exist but are not straightforward to be modeled, it is problematic to apply these criteria.

## 6.3 NUMERICAL OPTIMIZATION OF CRITERIA INCLUDING THE $L_1$ NORM

---

Recently, a number of methods has been introduced to optimize the  $L_1$  criterion. We discuss those that are applicable to large scale log-linear models. To deal with  $L_1$  penalties, the simplest idea is that of Kazama and Tsujii (2003) which was introduced for maximum entropy models but can be directly applied to conditional random fields. The main idea of Kazama and Tsujii (2003) is to split every parameter  $\theta$  into two positive constrained parameters,  $\theta^+$  and  $\theta^-$ , such that  $\theta = \theta^+ + \theta^-$ . The  $L_1$  penalty  $\rho|\theta|$  takes the form  $\rho(\theta^+ + \theta^-)$ , at most one variable in each pair  $\theta^+$  and  $\theta^-$  is non zero.

The optimization procedure is quite simple, but the number of parameters is doubled and Andrew and Gao (2007) reported that the method has a slow convergence rate.

Andrew and Gao (2007) have proposed the Quasi-Newton method adapted for the  $L_1$  penalized criterion which we consider just below.

An approach called Grafting is proposed in (Perkins et al., 2003). The method adds a parameter into the active set at each iteration. In order to decide which parameter is to be integrated in the model, a local derivative information is examined.

Lee et al. (2006) propose the IRLS-LARS (Iteratively Reweighted Least Squares - Least Angle Regression) algorithm to optimize the  $L_1$  regularized logistic regression. The  $L_1$  penalized least squares, solved with LARS (see (Efron et al., 2004) for details), are used to optimize the  $L_1$  penalized logistic regression criterion. Unfortunately, the method based on pure Newton optimization approach cannot be applied to large-scale problems.

### 6.3.1 ORTHANT-WISE LIMITED-MEMORY QUASI-NEWTON

The idea of the method proposed in (Andrew and Gao, 2007), is based on the observation that restricted to a set in which each coordinate never changes sign, the  $L_1$  norm is a differentiable linear function. Such sets are called orthants. The algorithm resembles Quasi-Newton. One of the major distinctions is the usage of the pseudo-gradient instead of the usual gradient. The pseudo-gradient is applied to determine which orthant to explore. The orthant-wise limited-memory quasi-Newton algorithm (OWL-QN) uses the inverse Hessian matrix update described in (Nocedal, 1980).

In the following, we define  $\sigma$  to be the sign function of a real number  $a$ :

$$\sigma(a) = \begin{cases} -1 & a < 0, \\ 0 & a = 0, \\ 1 & a > 0. \end{cases} \quad (6.8)$$

The criterion to be minimized is the negated log-likelihood penalized by the norm  $L_1$

$$\ell(\theta) + \rho_1 \|\theta\|_1. \quad (6.9)$$

The pseudo-gradient, is no more than a generalization of a gradient in that the directional derivative at  $\theta$  is minimized in the direction of  $\diamond \ell(\theta)$ :

$$\diamond_j \ell(\theta) = \begin{cases} \frac{\partial \ell(\bar{\theta})}{\partial \theta_j} + \rho_1 \sigma(\bar{\theta}), \bar{\theta}_j \neq 0, \\ \frac{\partial \ell(\bar{\theta})}{\partial \theta_j} + \rho_1, \frac{\partial \ell(\bar{\theta})}{\partial \theta_j} + \rho_1 < 0, \bar{\theta}_j = 0, \\ \frac{\partial \ell(\bar{\theta})}{\partial \theta_j} - \rho_1, \frac{\partial \ell(\bar{\theta})}{\partial \theta_j} - \rho_1 > 0, \bar{\theta}_j = 0, \\ 0, \text{ otherwise,} \end{cases}$$

The update takes the form

$$\theta^{t+1} = \pi(\theta^t + \tau q^t; \xi^t),$$

where

$$\pi_i(a; b) = \begin{cases} a_i, & \text{if } \sigma(a_i) = \sigma(b_i), \\ 0, & \text{otherwise,} \end{cases}$$

$\tau$  is a step size, usually adjusted by a line search,  $q$  is an update value that is the ratio (taken coordinate-wise) of the pseudo-gradient and the second derivative of the objective function, and  $\xi^t$  is a sign vector that contains information on the sign of every coordinate.

The inverse Hessian is approximated using first-order information, as in L-BFGS approach. To construct the changes of the gradient, the OWL-QN uses the gradient values of the unpenalized loss function which is differentiable everywhere. Coordinates that change sign are set to zero. The orthant is defined as follows:

$$\xi_j^t = \begin{cases} \sigma(\theta_j^t), & \text{if } \theta_j^t \neq 0, \\ \sigma(-\diamond_j \ell(\theta^t)), & \text{if } \theta_j^t = 0. \end{cases}$$

In comparison to other optimization methods mentioned above, OWL-QN can be applied to large-dimensional problems due to the limited memory updates that are similar to those used in L-BFGS.

In Sections 7.3.2 and 7.3.3 we illustrate performance of OWL-QN on real world data.

### 6.3.2 COORDINATE-WISE DESCENT

Although the ideas of coordinate-wise optimization were considered before, notably in (Dudík et al., 2004) and (Krishnapuram et al., 2005), we investigate the approach presented in (Friedman et al., 2007), which proposes to apply coordinate-wise descent to  $L_1$  penalized criteria. The idea of coordinate-wise methods consists in that one updates one parameter per iteration, i.e. “one-at-a-time” (Friedman et al., 2007). Let us start with the coordinate-wise method applied to the criterion of the univariate least-squares, since it is obvious to write its solution analytically.

#### Univariate Least Squares

For the least squares with a single predictor penalized by the elastic net

$$\sum_{i=1}^N (y_i - x_i \theta)^2 + \rho_2 \theta^2 + \rho_1 |\theta| \tag{6.10}$$

we can write the solution in analytical form:

$$\theta_k = \frac{S(\sum_{i=1}^N y_i x_i, \rho_1)}{\sum_{i=1}^N x_i^2 + \rho_2},$$

where the threshold function  $S$  is defined as follows:

$$\begin{aligned} S(a, \rho_1) &\equiv \sigma(a)(|a| - \rho_1)_+ \\ &= \begin{cases} a - \rho_1, & a \geq 0, \rho_1 \leq |a|, \\ a + \rho_1, & a \leq 0, \rho_1 \leq |a|, \\ 0, & \rho_1 \geq |a|. \end{cases} \end{aligned} \tag{6.11}$$

## Multivariate Least Squares

In the case of multiple predictors, if regressors are linearly independent, uncorrelated, and orthonormal, the problem is separable and we can apply coordinate-wise optimization.

Fixing parameters  $\theta_j, \forall j \neq k$  and considering  $\theta_k$  to be a single variable, the criterion takes the form

$$\frac{1}{2} \sum_{i=1}^N (y_i - \sum_{j \neq k} x_{ij} \theta_j - x_{ik} \theta_k)^2 + \rho_1 \sum_{j \neq k} |\theta_j| + \rho_1 |\theta_k| + \rho_2 \sum_{j \neq k} \theta_j^2 + \rho_2 \theta_k^2.$$

We find the minimum with respect to  $\theta_k$ , while looping over all parameters repeatedly until convergence. The update takes the form:

$$\theta_k = \frac{S\left(\sum_{i=1}^N x_{ik}(y_i - y_i^k), \rho_1\right)}{\sum_{i=1}^N x_{ik}^2 + \rho_2}, \quad (6.12)$$

where  $y_i^k = \sum_{j \neq k} x_{ij} \theta_j$ .

## Binary Multivariate Logistic Regression

For the logistic regression, Zou and Hastie (2005) propose to perform coordinate-wise update using a local quadratic approximation of the log-likelihood logistic regression function.

Fixing  $\theta_j \forall j \neq k$ , we can write the approximation the criterion of the binary logistic regression, already presented as equation (2.2) as

$$\begin{aligned} \ell_Q(\theta) = C^{st}(\tilde{\theta}) - \theta_k \sum_{i=1}^N x_i (y_i - g(y = 1|x_i)) + \frac{1}{2} \sum_{i=1}^N (\theta_k - \tilde{\theta}_k)^2 x_{ik}^2 w(x_i) + \\ \frac{1}{2} \sum_{i=1}^N \sum_{j \neq k} (\theta_k - \tilde{\theta}_k) x_{ij}^2 w(x_i) (\theta_j - \tilde{\theta}_j), \end{aligned}$$

and the update takes the form:

$$\theta_k = \frac{S\left(\tilde{\theta}_k \sum_{i=1}^N x_{ik}^2 w(x_i) + \sum_{i=1}^N \sum_{j \neq k} (\theta_j - \tilde{\theta}_j) x_{ij}^2 w(x_i) + \sum_{i=1}^N x_i (y_i - g(y = 1|x_i))\right), \rho_1}{\sum_{i=1}^N x_{ik}^2 w(x_i) + \rho_2}. \quad (6.13)$$

Note that an alternative version of the same idea is presented in (Dudík et al., 2004). The local behavior of the function  $\ell(\mathcal{D}; \theta)$  is approximated by a different function, of the first order only, that leads to a coordinate-wise optimization procedure. However, this approximation is based on the fact that every coordinate  $\theta_k$  is multiplied by a function which takes its values in  $\{0, 1\}$ . It is quite inappropriate for the conditional random fields model, where every parameter is weighted by  $\sum_{t=1}^T f_k(y_{t-1}, y_t, x_t)$ . Although  $f_k$  is a binary function and takes its values in  $\{0, 1\}$ , the sum is more than one if the configuration  $(y_{t-1}, y_t, x_t)$  is present more than once in a training sequence.

## 6.4 CONCLUSIONS

---

In this chapter, we illustrated the potential sparsity of the model. Setting more than 90% of parameters to zero a posteriori, we still achieve an acceptable accuracy. We considered the methods to perform model selection and to obtain sparse solutions based on  $L_1$  penalization. Numerical optimization methods can not be applied directly, since the criterion is not differentiable everywhere. We considered several recent optimization approaches. In particular, orthant-wise limited memory quasi-Newton, a modification of the quasi-Newton which achieves the state-of-the-art performance. A prospective approach that is easy to implement is coordinate-wise descent whose performance on the CRFs criterion and the real data we consider in the following chapter.

The elastic net criterion has been applied by Zou and Hastie (2005) to the binary and multiclass logistic regressions which can be generalized to CRFs. The interest to consider the elastic net in details for the logistic regression is twofold. First, we consider how the quadratic approximation of the log-likelihood function can be used. Second, coordinate-wise descent can be efficient for tasks with a limited number of parameters to be estimated. Zou and Hastie (2005) illustrate the idea to optimize parameters in blocks, where a block contains all features associated with a given class  $l$ . We make use of two above mentioned ideas in the next chapter, while penalizing the conditional random fields criterion with the elastic net penalty. The state-of-the-art results state that although the penalty terms based on the  $L_1$  norm produce a sparse and interpretable model, they do not perform necessarily better than the  $L_2$ -penalized criteria.

---

## CHAPTER 7

# APPLICATION OF COORDINATE-WISE OPTIMIZATION APPROACH TO CRFs

---

### Contents

---

<b>7.1</b>	<b>Coordinate-wise Method for Conditional Random Fields . . .</b>	<b>94</b>
7.1.1	Coordinate Descent and Discussion on the Approximation of the Second Derivatives . . . . .	94
7.1.2	Blockwise Coordinate Descent for CRFs . . . . .	96
<b>7.2</b>	<b>Implications of Sparsity: Sparse Forward-Backward . . . . .</b>	<b>97</b>
<b>7.3</b>	<b>Experiments with Elastic Net Conditional Random Fields . . .</b>	<b>98</b>
7.3.1	Artificial Data Set . . . . .	98
7.3.2	Nettalk Corpus (Phonetisation Task) . . . . .	103
7.3.3	CoNLL 2000 and CoNLL 2003 Data Sets . . . . .	105
<b>7.4</b>	<b>Conclusions . . . . .</b>	<b>112</b>

---

Conditional random fields are able to incorporate a large number of dependencies, however as we have illustrated in the previous chapter, only some part of them has to be kept to reach a reasonable accuracy. In Chapter 3 we have observed that the conditional random fields achieve its best performance with redundant and highly correlated features. Zou and Hastie (2005) noticed that in the case of correlated features it is more appropriate to apply the elastic net penalty to perform model selection than the  $L_1$  norm.

In this chapter, we apply coordinate-wise descent to the negated log-likelihood function of conditional random fields penalized by the elastic net. Real world applications can involve millions of parameters to be estimated, and it is infeasible to perform single coordinate optimization. We investigate blockwise updating schemes to speed up the optimization procedure.

We compare the proposed optimization algorithm for CRFs with simple heuristic model selection methods and with the state-of-the art orthant-wise quasi-Newton approach, considered in the previous chapter.

## 7.1 COORDINATE-WISE METHOD FOR CONDITIONAL RANDOM FIELDS

---

In this section, we consider local quadratic approximations for CRFs and discuss ways to avoid computations of the full Hessian matrix. We denote by  $\ell(\mathcal{D}; \theta)$  the negated log-likelihood of CRF, defined in equation (3.1).

In the multivariate case, fixing all parameters for  $\theta_j \neq \theta_k$ , we write a second order Taylor expression of the negative log-likelihood with respect to the parameter  $\theta_k$ :

$$C^{st}(\tilde{\theta}) + \frac{\partial \ell(\mathcal{D}; \tilde{\theta})}{\partial \theta_k} (\theta_k - \tilde{\theta}_k) + \frac{1}{2} (\theta_k - \tilde{\theta}_k)^2 \frac{\partial^2 \ell(\mathcal{D}; \tilde{\theta})}{\partial \theta_k^2},$$

where  $\tilde{\theta}$  denotes the current value of the parameter. Taking into account the elastic net penalty and the quadratic approximation, the update step is

$$\theta_k = \frac{S\left(\tilde{\theta}_k \frac{\partial^2 \ell(\mathcal{D}; \tilde{\theta})}{\partial \theta_k^2} - \frac{\partial \ell(\mathcal{D}; \tilde{\theta})}{\partial \theta_k}, \rho_1\right)}{\frac{\partial^2 \ell(\mathcal{D}; \tilde{\theta})}{\partial \theta_k^2} + \rho_2}, \quad (7.1)$$

where  $S$  is the soft-threshold function defined by (6.11).

### 7.1.1 COORDINATE DESCENT AND DISCUSSION ON THE APPROXIMATION OF THE SECOND DERIVATIVES

The application of coordinate-wise descent to conditional random fields requires computation of the second derivative of the log-likelihood function. If the first order derivative is readily computable using the forward-backward recursions described in Section 7.2, the exact computation of the second derivative is more problematic for CRFs.

The diagonal elements of the Hessian are given by

$$\frac{\partial^2 \ell(\theta)}{\partial \theta_k^2} = \sum_{i=1}^N \left\{ \mathbb{E}_{p_\theta(\mathbf{y}|\mathbf{x}^{(i)})} \left( \sum_{t=1}^{T_i} f_k(y_{t-1}, y_t, x_t^{(i)}) \right)^2 - \left( \mathbb{E}_{p_\theta(\mathbf{y}|\mathbf{x}^{(i)})} \sum_{t=1}^{T_i} f_k(y_{t-1}, y_t, x_t^{(i)}) \right)^2 \right\}. \quad (7.2)$$

The first term is problematic as it involves the conditional expectation of a square which cannot be computed only from the pairwise probabilities  $p_\theta(y_{t-1} = y', y_t = y | \mathbf{x}^{(i)})$  returned by the forward-backward procedure. It can be shown (see Chapter 4 of (Cappé et al., 2005) and (Cappé and Moulines, 2005)) that (7.2) can be computed using auxiliary recursions related to the usual forward recursion with an overall complexity of order  $|Y|^2 \times T_i$  per sequence. Unfortunately, this recursion is specific for each index  $k$  and cannot be shared between parameters. As we will see below, sharing (part of) the computations between

parameters is desirable feature for handling non trivial CRFs; we thus propose to use instead the approximation

$$\frac{\partial^2 \ell(\mathcal{D}; \theta)}{\partial \theta_k^2} \approx \sum_{i=1}^N \sum_{t=1}^{T_i} \left\{ \mathbb{E}_{p_\theta(\mathbf{y}|\mathbf{x}^{(i)})} f_k(y_{t-1}, y_t, x_t^{(i)}) - \left( \mathbb{E}_{p_\theta(\mathbf{y}|\mathbf{x}^{(i)})} f_k(y_{t-1}, y_t, x_t^{(i)}) \right)^2 \right\}. \quad (7.3)$$

This approximation assumes that, given  $\mathbf{x}^{(i)}$ ,  $f_k(y_{t-1}, y_t, x_t^{(i)})$  and  $f_k(y_{s-1}, y_s, x_s^{(i)})$  are uncorrelated when  $s \neq t$ . Note that this approximation is exact when the feature  $f_k$  is only active at one position along the sequence. It is likely that the accuracy of this approximation is reduced when  $f_k$  is active twice, especially if the corresponding positions  $s$  and  $t$  are close. In the linear-chain CRFs considered here, this can happen only if some symbols in the observation sequence occur repeatedly.

The coordinate descent algorithm applied to CRFs is thus summarized as Algorithm 8.

---

**Algorithm 8** Coordinate-wise Descent for CRF

---

**Require:** Observations and their labels,  $\rho_1, \rho_2$

**Ensure:**  $\theta$

Initialize  $\theta = 0^T$

**while** Convergence criterion is not met **do**

**for** every parameter  $\theta_k$  **do**

**for** all sequences for which  $\theta_k$  is active **do**

      Compute  $\partial \ell(\mathcal{D}; \tilde{\theta}) / \partial \theta_k$ ,  $\partial^2 \ell(\mathcal{D}; \tilde{\theta}) / \partial \theta_k^2$

      Perform update using equation (7.1).

**end for**

**end for**

**end while**

---

A potential issue with this algorithm is the fact that, in contrast to the logistic regression case considered in (Friedman et al., 2008), we are using an approximation to  $\partial^2 \ell(\mathcal{D}; \theta) / \partial \theta_k^2$  which could have a detrimental effect on the convergence of the coordinate descent algorithm. An important observation is that (7.3) used with an approximate second order derivative still yields the correct stationary points (see also (Krishnapuram et al., 2005)).

To see why it is true, assume that  $\tilde{\theta}$  is such that (7.3) leaves  $\tilde{\theta}_k$  unchanged (i.e.,  $\theta_k = \tilde{\theta}_k$ ). If  $\tilde{\theta}_k = 0$ , this can happen only if  $|\partial \ell(\mathcal{D}; \tilde{\theta}) / \partial \theta_k| \leq \rho_1$ , which is indeed the first order optimality condition in 0. Now assume that  $\tilde{\theta}_k > 0$ , the fact that  $\tilde{\theta}_k$  is left unmodified by the recursion implies that  $\tilde{\theta}_k \rho_2 + \partial \ell(\mathcal{D}; \tilde{\theta}) / \partial \theta_k + \rho_1 = 0$ , which is also recognized as the first order optimality condition (note that since  $\tilde{\theta}_k \neq 0$ , the criterion is differentiable at this point). The symmetric case, where  $\tilde{\theta}_k < 0$ , is similar. Hence, the use of an approximated second order derivative does not prevent the algorithm from converging to the appropriate solution. A more subtle issue is the question of stability: it is easily checked that if  $\partial^2 \ell(\mathcal{D}; \theta) / \partial \theta_k^2$  is smaller than it should be (remember that it has to be positive as  $\ell(\mathcal{D}; \theta)$  is strictly convex), the algorithm can fail to converge even for simple functions (e.g., if  $\ell(\mathcal{D}; \theta)$  is a quadratic function). An elaborate solution to this



issue would consist in performing a line search in the “direction”

$$s \left( \frac{\tau^{-1} \frac{\partial^2 \ell(\mathcal{D}; \tilde{\theta})}{\partial \theta_k^2} \tilde{\theta}_k - \frac{\partial \ell(\mathcal{D}; \tilde{\theta})}{\partial \theta_k}, \rho_1 \right),$$

$$\frac{\tau^{-1} \frac{\partial^2 \ell(\mathcal{D}; \tilde{\theta})}{\partial \theta_k^2} + \rho_2}{},$$

where  $0 < \tau \leq 1$ , is chosen as close as possible to 1 with the constraint that it indeeds leads to a decrease of the objective function (note that the step size affects only the second order term in order to preserve the convergence behavior). On the other hand, coordinate descent algorithms are only reasonable if each individual update can be performed very quickly, which means that using line search is not really an option. In our experiments, we found that using a fixed value of  $\tau = 1$  was sufficient for Algorithm 8, probably due to the fact that the second order derivative approximation is usually quite good.

For the blockwise approach described below, we had to use larger values of  $\tau$  to ensure stability. To be precise, in our experiments the second derivative is scaled by

$$\max \left( \kappa_1, \left| \frac{\partial \ell(\mathcal{D}; \tilde{\theta})}{\partial \theta_k} / \frac{\partial^2 \ell(\mathcal{D}; \tilde{\theta})}{\partial \theta_k^2} \right| \right) \kappa_2, \tag{7.4}$$

where  $\kappa_1$  and  $\kappa_2$  are empirically chosen values that guarantee the absence of numerical problems. The heuristics makes the algorithm stable and as we will make sure in the experiments in Section 7.3, it converges within a reasonable number of iterations.

### 7.1.2 BLOCKWISE COORDINATE DESCENT FOR CRFs

The algorithm described in the previous section is efficient in simple problems but cannot be used, even for moderate size applications of CRFs. As for instance, the Nettek application involves  $|Y|^2 * |X| + |Y| * |X| = 75,790$  parameters and single component coordinate descent is definitely ruled out in this case. Following the idea of Friedman et al. (2008), we investigate the use of blockwise updating schemes, which update several parameters simultaneously trying to share as much computations as possible. It turns out that the case of CRFs is rather different from the polytomous logistic regression case considered in (Friedman et al., 2008) and requires specific blocking schemes. In this discussion, we consider the parametrization defined in (6.1) which makes it easier to highlight the proposed block structure.

The forward-backward procedure shows that the computation of the first or second order derivative of the objective function with respect to  $\mu_{y,x}$  or  $\lambda_{y',y,x}$  requires to compute the pairwise probabilities  $p_{\theta}(y_t = y'', y_{t+1} = y' | \mathbf{x}^{(i)})$  for all values of  $(y'', y') \in Y^2$  and for all sequences  $\mathbf{x}^{(i)}$  which contain the symbol  $x$  (at any position in the sequence). Hence, the most natural grouping in this context is to update simultaneously the set of all parameters  $\{\mu_{y,x}, \lambda_{y',y,x}\}_{(y',y) \in Y^2}$  that correspond to the same value of  $x$ . This grouping is orthogonal to the solution adopted for polytomous regression in (Friedman et al., 2008), where parameters are grouped by common values of the target label.

The blockwise procedure is presented as Algorithm 9.

Different variants of this algorithm are possible, including updating only one of the sub blocks  $\{\mu_{y,x}\}_{y \in Y}$  or  $\{\lambda_{y',y,x}\}_{(y',y) \in Y^2}$  at a time or using a full Hessian approximation (at

---

**Algorithm 9** Blockwise Coordinate Descent for CRF with Diagonal Hessian Approximation

---

**Require:** Observations and their labels,  $\rho_1, \rho_2$

**Ensure:**  $\theta$

Initialize  $\theta = 0^T$

**while** Convergence criterion is not met **do**

**for**  $i = 1 : |\text{x alphabet}|$  **do**

    ind  $\leftarrow$  indices associated with  $x_i$

**for**  $k = 1 : |\text{ind}|$  **do**

**for** all sequences for which  $\theta_k$  is active **do**

        Compute  $\partial\ell(\mathcal{D}; \tilde{\theta})/\partial\theta_k$ ,  $\partial^2\ell(\mathcal{D}; \tilde{\theta})/\partial\theta_k^2$

        Perform update using equation (7.1).

**end for**

**end for**

**end for**

**end while**

---

least when  $|Y|$  is not too large). The expression of full Hessian for a block of parameters is provided in Appendix D. On the examples that we have considered so far, the above solution appeared to be preferable to these alternatives. Although the above algorithm requires scanning all the  $|X|$  possible symbols  $x$  at each iteration, it is usually relatively fast due to the fact that only those sequences that contain  $x$  are considered.

## 7.2 IMPLICATIONS OF SPARSITY: SPARSE FORWARD-BACKWARD

---

The standard approach for computing the conditional probabilities in CRFs is inspired by the forward-backward algorithm for hidden Markov models: in the case of the parametrization of (6.1), the algorithm implies the computation of

$$\begin{cases} \alpha_1(y) = \exp(\mu_{y,x_1} + \lambda_{y_0,y,x_1}), \\ \alpha_{t+1}(y) = \sum_{y'} \alpha_t(y') \exp(\mu_{y,x_{t+1}} + \lambda_{y',y,x_{t+1}}), \end{cases} \quad (\text{Forward Recursion})$$

$$\begin{cases} \beta_{T_i}(y) = 1, \\ \beta_t(y') = \sum_y \beta_{t+1}(y) \exp(\mu_{y,x_{t+1}} + \lambda_{y',y,x_{t+1}}), \end{cases} \quad (\text{Backward Recursion})$$

where the joint probabilities  $p_\theta(y_t = y', y_{t+1} = y | \mathbf{x}^{(i)})$  and the normalization constant  $Z_\theta(\mathbf{x}^{(i)})$  are obtained by normalizing  $\alpha_t(y') \exp(\mu_{y,x_{t+1}} + \lambda_{y',y,x_{t+1}}) \theta_{t+1}(y)$  and  $\alpha_{T_i}(y)$ , respectively. These recursions require a number of operations that grows quadratically with the size of  $Y$ .

Let us now consider the case where the set of features  $\{\lambda_{y',y,x_{t+1}}\}_{(y',y) \in Y^2}$  is sparse with only  $r(x_{t+1}) \ll |Y|^2$  non null values and define the  $|Y| \times |Y|$  matrix

$$M_{t+1}(y', y) = \exp(\lambda_{y',y,x_{t+1}}) - 1.$$

Observe that  $M_{t+1}(y', y)$  also is sparse and that the forward and backward equations may

be rewritten as

$$\begin{aligned}\alpha_{t+1}(y) &= \exp(\mu_{y,x_{t+1}}) \left\{ \sum_{y'} \alpha_t(y') + \sum_{y'} \alpha_t(y') M_{t+1}(y', y) \right\}, \\ \beta_t(y') &= \sum_y v_{t+1}(y) + \sum_y M_{t+1}(y', y) v_{t+1}(y),\end{aligned}\tag{7.5}$$

where  $v_{t+1}(y) = \beta_{t+1}(y) \exp(\mu_{y,x_{t+1}})$ . The resulting computational savings stem from the fact that the vector matrix products in (7.5) now only involve the sparse matrix  $M_{t+1}(y', y)$ . This means that they can be computed, using an appropriate sparse matrix implementation, with exactly  $r(x_{t+1})$  multiplications instead of  $|Y|^2$ . If the set  $\{\mu_{y,x_{t+1}}\}_{y \in Y}$  is also sparse, one may use a similar idea although the computation savings will in general be less significant. Of course, the same tricks may also be used to speed up the decoding step.

Using this implementation, the complexity of the forward-backward procedure for the sequence  $\mathbf{x}^{(i)}$  can be reduced from  $T_i \times |Y|^2$  to the cumulated sizes of the feature sets encountered at each position along the sequence. On average, it means that the complexity of the forward-backward procedure is proportional to the average number of active features per position in the parameter set rather than to the actual number of potentially active features. We illustrate the efficiency of the proposed approach on the real data set in Section 7.3.2. This observation suggests that it might even be possible to use some longer term dependencies between labels, as long as only a few of them are active simultaneously.

In Section 7.1.1 we discussed the approximation that no matter how many times a feature  $f_k$  is observed in a sequence, we consider the feature  $f_k$  to be active only on one position in the sequence. In such a case we can perform  $\alpha$ - and  $\beta$ -passes until position  $t$  which is the first occurrence of the feature  $f_k$ . The normalization factor  $Z_\theta(\mathbf{x}^{(i)})$  can be computed as a product of  $\alpha_t$  and  $\beta_t$ .

## 7.3 EXPERIMENTS WITH ELASTIC NET CONDITIONAL RANDOM FIELDS

---

In this section, we discuss the efficiency of the elastic net penalty applied to conditional random fields. We carry out experiments on both artificial data and real world applications in the domain of natural language processing.

### 7.3.1 ARTIFICIAL DATA SET

In this section, we illustrate that a sparse model can reach the same performance as a model with rich and numerous dependencies.

The synthetic data are simulated with hidden Markov models. The observation alphabet contains 5 symbols, the size of the labels alphabet is 6. Note, that the data are generated in such a way that only two transition probabilities  $(y_{t-1}, y_t)$  are important,

all the others transition probabilities from a previous state to a next one are uniform. In other words, the distribution is almost completely defined by the conditional probability of an observation given a state. Figure 7.1 illustrates the data generation mechanism by representing matrices of state transition probabilities and conditional probability of an observation given its state.

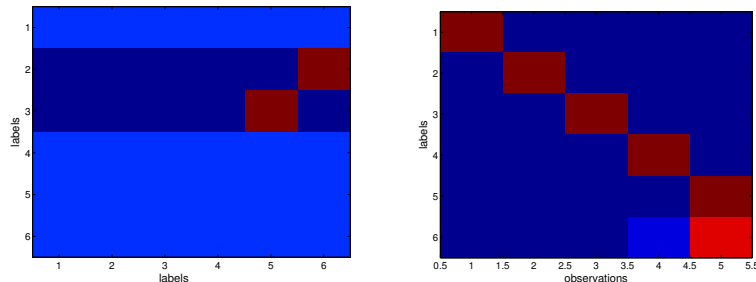


Figure 7.1: Distribution generating synthetic data. Left: distribution  $p(y_t|y_{t-1})$ , right:  $p(x_t|y_t)$ .

Figure 7.2 compares several models: M1 contains both  $(y_{t-1}, y_t, x_t)$  and  $(y_t, x_t)$  features, M2 and M3 are simpler, with M2 containing only the bigram features, and M3 only the unigram features. The models M1–M3 are penalized with the  $L_2$  norm. Models M4–M8 contain both features, bigram and unigram, but are penalized by the elastic net penalty term. For the  $L_2$ -penalized models (M1–M3), the regularization factor  $\rho_2$  is set to its optimal value (obtained by cross validation). For M4–M8 however, the value of  $\rho_2$  does not influence much the performance and is set to 0.001 while M4–M8 correspond to different choices of  $\rho_1$ , as shown in Table 7.1. The  $L_2$  penalty term in the elastic net prevents numerical problems that can occur when the Hessian values are very small. However, the heuristics which we added, equation (7.4) to guarantee the stability, ensures that there are no numerical problem as well. That is why we can fix  $\rho_2$  to a small value.

For this experiment, we used only  $N = 10$  sequences for training, so as to reproduce the situation, which is prevalent in practical uses of CRFs, where the number of training tokens (here  $10 \times 5 = 50$ ) is of the same order as the number of parameters, which ranges from  $6 \times 5 = 30$  for M3 to  $6 \times 5 + 6^2 \times 5 = 210$  for M1 and M4–M8. Figure 7.2 displays box-and-whiskers plots summarizing 100 independent replications of the experiment.

	M4	M5	M6	M7	M8
$\rho_1$	0.001	0.01	0.1	1	2.5
Number of active unigram features	28.5	15.0	10.9	6.2	5.8
Number of active bigram features	50.6	26	17.2	4.9	1.3

Table 7.1: Impact of  $\rho_1$  on the number of active features ( $\rho_2 = 0.001$ ).

Unsurprisingly, M1 and M2, which contain more parameters, perform very well on the training set, much better than M3. The test performance tells a different story: M2 performs in fact much worse than the simple unigram model M3, which is all the more remarkable that we know from the simulation model that the observed tokens are indeed not independent and that the models are nested (i.e. any model of type M3 corresponds to a model of type M2). Thus, even with regularization, richer models are not necessary the best, hence the need for feature selection techniques. Interestingly, M1 which embarks

both unigram and bigram features, achieves the lowest test error, highlighting the interest of using simultaneously both feature types to achieve some sort of smoothing effect. With proper choice of the regularization (here, M7),  $L_1$ -penalized models achieve comparable test set performance. As a side effect of model selection, notice that M7 is somewhat better than M1 at predicting the test performance at training time: for M1, the average train error is 6.4% vs. 18.5% for the test error while for M7, the corresponding figures are 10.3% and 17.9%, respectively. Finally, closer inspection of the sparsity pattern determined by M7 shows that it is most often closely related to the structure of the simulation model which is also encouraging.

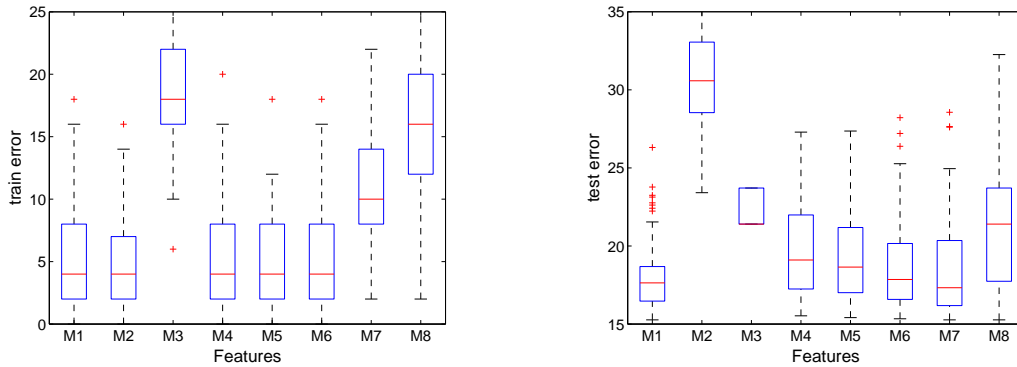


Figure 7.2: Performance of the models on artificial data. Models  $M1 - M3$  are trained with  $L_2$  penalty (L-BFGS), models  $M4 - M8$  with the  $L_1$  penalty term (block coordinate-wise descent). Left: performance on training set. Right: performance on testing set.

The following figures display the average frequency of feature selection for all individual features (we have done 50 Monte-Carlo replications). The training was carried out with the parameters  $\rho_1 = 0.8$  and  $\rho_2 = 0.001$ . The figures illustrate not the values but the average of the coordinate-wise selection frequency, in other words which features are selected by the elastic net and how often. Therefore, a parameter associated with a large value can be either positive or negative. Figure 7.3 on the left illustrates average selection frequency of unigram features. On the right, we display the average values of these features. Notice, that the unigram features associated with  $x = 5$  are active for all  $y$ . However, among these features only two of them correspond to positive values. Figure 7.4 displays the average selection frequency of bigram features  $\lambda$  for every  $x$ .

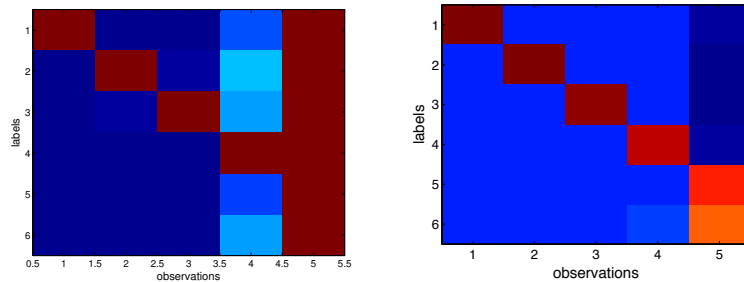


Figure 7.3: Average selection frequency of unigram  $\mu$  features (on the left) and their estimated average values (on the right).

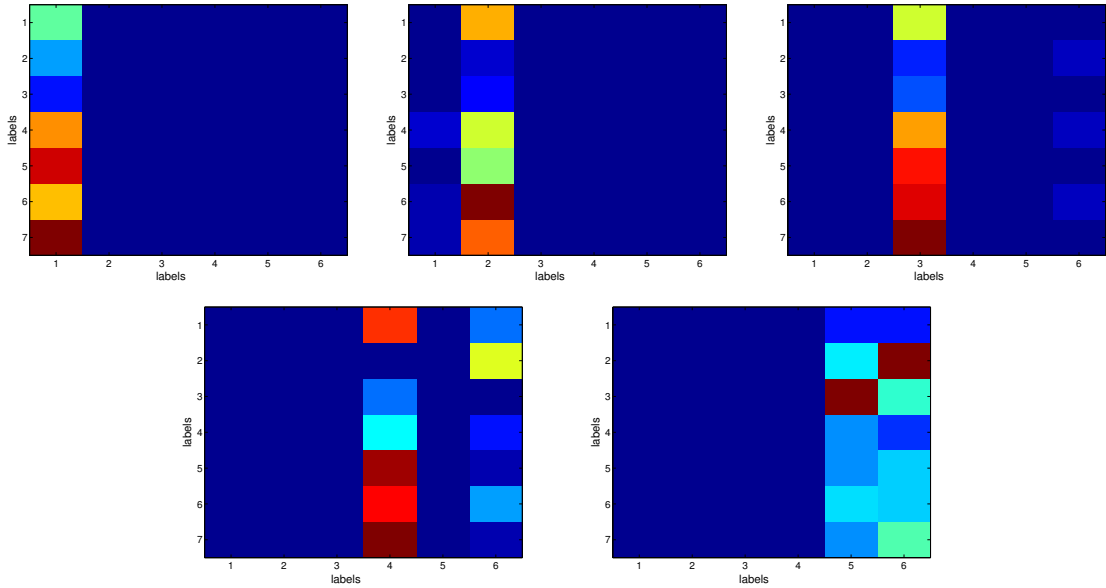


Figure 7.4: Average selection frequency of bigram  $\lambda$  features for  $X = 1, \dots, 5$ .

The estimated parameters capture the initial structure and dependencies quite well. The true and learnt probability distributions are almost completely defined by unigram features probability. Figure 7.3 shows clearly that  $x = 5$  is the most ambiguous among all observations given  $y$ . Figure 7.4 displays the strong dependency of a current  $x$  on a current  $y$ . For the most ambiguous  $x = 5$ , we get the pattern  $(y_t|y_{t-1})$  similar to the true one (on the left of Figure 7.1)

## Coordinate-wise and Blockwise Optimization

Figure 7.5 compares the behavior of the coordinate-wise update policy with the blockwise approach, where one iteration refers to a complete round where all model parameters are updated exactly once. As can be seen on these graphs, the convergence behavior is comparable for both approaches, both in terms of objective function (Figure 7.6) and test error (right plot of Figure 7.5). Each iteration of the blockwise algorithm is however about 50 times faster than the coordinate-wise update, which roughly correspond to the size of each block. Clearly, the blockwise approach is the only viable strategy when tackling more realistic higher-dimensional tasks such as those considered in the next two sections.

The algorithms introduced cycle every iteration over all parameters. The goal is to set a number of parameters to zero, and therefore select the most influential dependencies. The algorithms can change parameter values on every iteration, and hence, on every iteration add or eliminate variables from the model. According to our empirical results, once a parameter is zeroed, it rarely enters the active set in subsequent iterations. Figures 7.5 and 7.6 display the train and test errors, and the values of logistic loss on the simulated data ( $n = 10$ , 50 Monte-Carlo replications, Bayes error  $\approx 15\%$ ) of the coordinate-wise method, blockwise optimization, and the coordinate-wise algorithm with zeroed features not revisited. For the synthetic data, the results are in favor of deleting a dependency once it is zeroed at some iteration, since as we can see, zeroed features practically do not re-enter the model. The approach is fast in comparison to the coordinate-wise approach

that visits all the coordinates. The blockwise update takes 7.8 seconds for 10 iterations, the coordinate-wise method cycling over all features 126.5 seconds, and the coordinate-wise optimization cycling only over features that were active in the previous step, 25.8 seconds.

However, in the following we apply the blockwise procedure, since computationally it is more efficient and faster. The results presented in Sections 7.3.2 and 7.3.3 are obtained with the blockwise version of the algorithm, coded in C<sup>1</sup>. We refer to the proposed method as to Sparse Blockwise Coordinate Descent (SBCD), since the algorithm implements the sparse Forward-Backward discussed in Section 7.2.

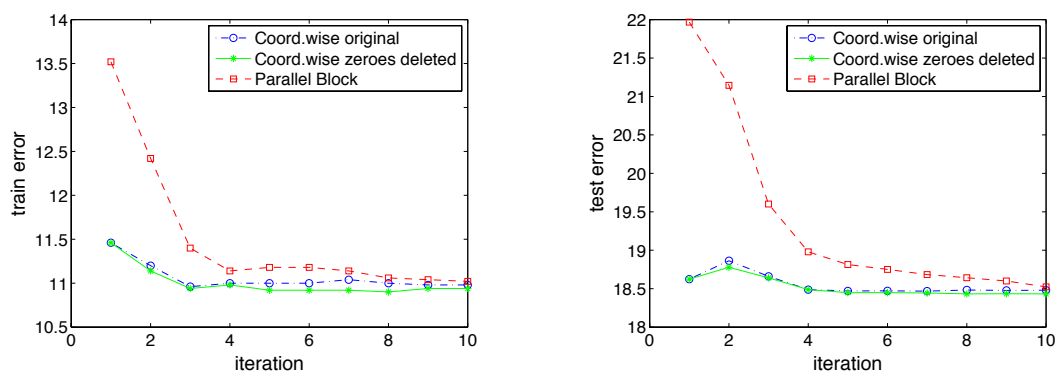


Figure 7.5: Performance comparison of coordinate-wise method, block-wise method, and coordinate-wise method that does not revisit points which have been zeroed in a previous iteration

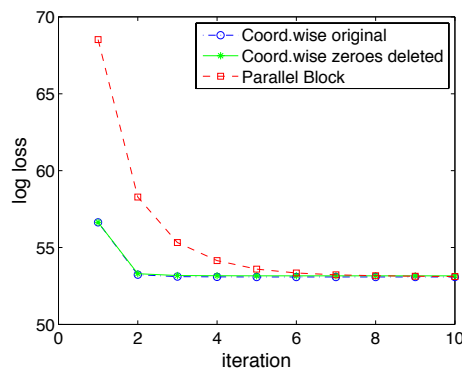


Figure 7.6: Logarithmic loss comparison of coordinate-wise method, block-wise method, and coordinate-wise method that does not revisit points which have been zeroed in a previous iteration

<sup>1</sup>Implementation by Thomas Lavergne, LIMSI, University Paris-Sud, XI

### 7.3.2 NETTALK CORPUS (PHONETISATION TASK)

The SBCD algorithm is tested on the Nettetalk corpus. In our experiments, we consider that each phoneme is a target label, and we only use features that test the value of one single letter. The training set comprises 16,452 sequences and the test set contains 1,628 sequences.

Figure 7.7 displays the parameter sets estimated for the  $L_1$  penalty with  $\rho_1 = 0.2$ . One can see that the algorithm identifies correctly some parameters that are important for the task. The first column corresponds to the null sound, and is associated with almost all letters. One can also directly visualize the ambiguity of the vocalic graphemes which correspond to the first ('a'), fifth ('e'), ninth ('i')... gray rows; this contrasts with the much more deterministic association of one consonant grapheme with one single consonant phoneme. See Appendix B for the list of phonemes.

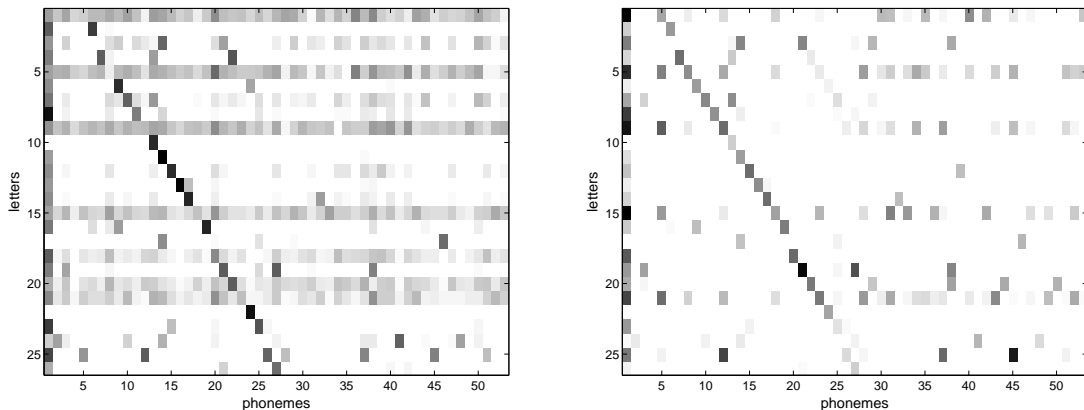


Figure 7.7: Nettetalk experiments,  $\rho_1 = 0.2$ ,  $\rho_2 = 0.05$ . Left: feature values of type unigram. Right: feature values of type bigram:  $\sum_{y_{t-1}} |\lambda_{y_{t-1}, y_t, x_t}|$

Method	Iter.	Time (min.)	Train (%)	Test (%)	$K_\mu$	$K_\lambda$
SBCD, $\rho_1 = 0$	30	125	13.3	14.0	1,378	73,034
SBCD, $\rho_1 = 0.1$	30	76	13.5	14.2	1,155	4,171
SBCD, $\rho_1 = 0.2$	30	70	14	14.2	1,089	3,598
SBCD, $\rho_1 = 0.5$	30	63	13.7	14.3	957	3,077
SBCD, $\rho_1 = 1$	30	55	16.3	16.8	858	3,111
SBCD, $\rho_1 = 2$	30	43	16.4	16.9	760	2,275
SBCD, $\rho_1 = 10$	30	25	17.3	17.7	267	997
OWL-QN, $\rho_1 = 0.1$	50	165	13.5	14.2	1,864	4,079
L-BFGS	90	302	13.5	14.1	74,412	
SGD	30	17	18.5	19.1	74,412	

Table 7.2: Upper part: summary of results for various values of  $\rho_1$  for the proposed Sparse Blockwise Coordinate Descent (SBCD) algorithm (with  $\rho_2 = 0.001$ ) and orthant-wise L-BFGS (OWL-QN). Lower part: results obtained with  $\rho_2$  regularization only, for L-BFGS and stochastic gradient descent (SGD).

Table 7.2 gives the per phoneme accuracy with varying level of sparsity, both for the



proposed algorithm (SBCD) and the orthant-wise L-BFGS (OWL-QN) strategy of Andrew and Gao (2007). For comparison purposes the lower part of the table also reports performance obtained with  $L_2$  regularization only. For  $L_2$ -based methods (L-BFGS and SGD) the regularization constant was set to its optimal value determined by cross validation as  $\rho_2 = 0.02$ . The proposed algorithm (SBCD) is C coded while OWL-QN and L-BFGS use the CRF++ package (Kudo, 2005) modified to use the `liblbfgs` library provided with CRFsuite (Okazaki, 2007) that implements the standard and orthant-wise modified versions of L-BFGS. Finally, SGD uses the software of Bottou (2007). All running times were measured on a computer with an Intel Pentium 4 3.00GHz CPU and 2G of RAM memory. Measuring running time is a difficult issue as each iteration of the various algorithms does not achieve the same improvement in term of performance. For the proposed method, 30 iterations were found necessary to reach reasonable performance in the sense that further iterations did not significantly reduce the error rates (with variations smaller than 0.3%). Proceeding similarly for the other methods showed that OWL-QN and L-BFGS usually require more iterations to reach stable performance, which is reflected in Table 7.2. Finally, SGD requires few iterations (where an iteration is defined as a complete scan of all the training sequences) although we obtained disappointing performance on this dataset with SGD, since the step, equation (3.29), of the algorithm becomes too small to make significant descent along the gradient.

First, Table 7.2 shows that for  $\rho_1 = 0.1$  or  $0.2$  ( $\kappa_1 = 1$  and  $\kappa_2 = 1.2$  guarantee the reasonable rate of convergence) the proposed method reaches an accuracy that is comparable with that of non-sparse trainers (SBCD with  $\rho_1 = 0$  or L-BFGS) but with only about 5000 active features. Note in particular the dramatic reduction achieved for the bigram features  $\lambda_{y',y',x}$  as the best accuracy/sparsity compromise ( $\rho_2 = 0.2$ ) nullifies about 95% of these parameters. We observe that the performance of SBCD (for  $\rho_1 = 0.1$ ) is comparable to that of OWL-QN, which is reassuring as they optimize related criteria, except for the fact that OWL-QN is based on the use of the sole  $L_1$  penalty. There are however minor differences in the number of selected features for both methods. In addition to the slight difference in the penalties used by SBCD and OWL-QN, it was constantly observed in all our experiments that for  $L_1$ -regularized methods the performance stabilizes much faster than the pattern of selected features which may require as much as a few hundreds of iterations to fully stabilize. This effect was particularly noticeable with the OWL-QN algorithm. We have not found satisfactory explanation regarding the poor performance of SGD on this dataset: further iterations do not significantly improve the situation and this failure has not been observed on the CoNLL 2003 data considered below. In general, SGD is initially very fast to converge and no other algorithm is able to obtain similar performance with such small running time. The fact that SGD fails to reach satisfactory performance in this example is probably related to an incorrect decrease of the step size. In this regard, an important difference between the Nettealk data and the CoNLL 2003 example considered below is the number of possible labels which is quite high here (53). A final remark regarding timings is that all methods except SBCD use logarithmic computation in the forward-backward recursions. As discussed in Section 7.2, this option is slower by a factor which, in our implementation, was measured to be about 2.4. Still, the SBCD algorithm compares favorably with other algorithms, especially with OWL-QN which optimizes the same objective function.

Table 7.2 also shows that the running time in the SBCD method depends on the sparsity of the estimated model, which is fully attributable to the sparse version of the

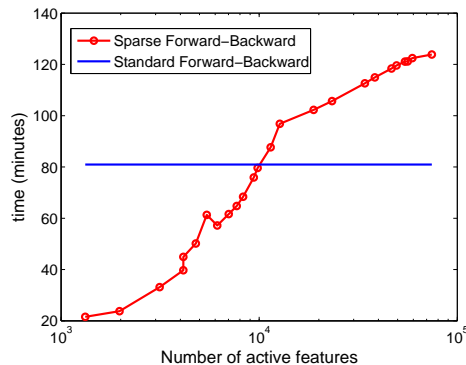


Figure 7.8: Running time as a function of the number of active features for the SBCD algorithm on the Nettetalk corpus. The blue line corresponds to the running time when using non-sparse forward-backward.

forward-backward recursion. To make this connection clearer, Figure 7.8 displays the running time as a function of the number of active features (rather than  $\rho_1$ ). When the number of active feature is less than 10000, the curve indeed shows a decrease that is proportional to the number of active features (beware that the x-axis is drawn on a logarithmic scale). The behavior observed for larger numbers of actives features, where the sparse implementation becomes worse than the baseline (horizontal blue line) can be attributed to the overhead generated by the use of sparse matrix-vector multiplications for matrices that are indeed not sparse. Hence the sparse forward-backward approach really has a strong potential for reducing the computational burden in situations where the active parameter set is very small compared to the total number of available features. Note also that the OWL-QN optimizer could benefit from this idea as well.

### 7.3.3 CoNLL 2000 AND CoNLL 2003 DATA SETS

For the CoNLL 2000 and CoNLL 2003 corpora we apply the blockwise variant of the algorithm, the same approach as for the Nettetalk data. We carry out several experiments, with two feature sets.

The first experiment is performed with  $(y_{t-1}, y_t, x_t^1 x_t^2)$  and  $(y_t, x_t^1 x_t^2)$  for CoNLL 2000 and  $(y_{t-1}, y_t, x_t^1 x_t^2 x_t^3)$  and  $(y_t, x_t^1 x_t^2 x_t^3)$  for CoNLL 2003 features, where  $x^1$  is associated with words,  $x^2$  with part-of-speech tags, and  $x^3$  with syntactic chunks. In other words, the role of observations is played by a Cartesian product of all possible words  $\times$  POS tags for CoNLL 2000 and words  $\times$  POS tags  $\times$  syntactic tags for CoNLL 2003. It is an illustration that the choice of dependencies can be inefficient. The error rate is far from optimal for all possible values of  $\rho_1$  and  $\rho_2$ . On the CoNLL 2000 set we get 13.5% error on the test set, and on the CoNLL 2003 we did not reach a better performance than approximately 6% errors on the test A, and 10% on test B.

For another experiment we decompose all types of observations (“+” as in Chapter 3 is used to denote the superposition of different types of features):

$$(y_{t-1}, y_t, x_t^1) + (y_{t-1}, y_t, x_t^2) + (y_t, x_t^1) + (y_t, x_t^2)$$

for CoNLL 2000 and

$$(y_{t-1}, y_t, x_t^1) + (y_{t-1}, y_t, x_t^2) + (y_{t-1}, y_t, x_t^3) + (y_t, x_t^1) + (y_t, x_t^2) + (y_t, x_t^3)$$

for CoNLL 2003. With such a choice of features we practically achieve state-of-the-art performance. We run 30 iterations of the blockwise algorithm for each corpora, and we see quite well, e.g., on CoNLL 2000, Figure 7.10 that the method converges and the optimization is stabilized. The number of active parameters decreases drastically. As in Chapter 3, the number of extracted parameters associated with unigram features of the form  $(y_t, x_t)$  equals to  $|Y| \times |X'|$ , and for the features  $(y_{t-1}, y_t, x_t)$  to  $|Y|^2 \times |X'|$ , where  $X' \in X$  is the subset of all patterns that are observed in the training data. Tables 7.3 and 7.4 demonstrate a number of parameters of each type that is active on the first iteration step, and that is active after 30 iterations on CoNLL 2000 and CoNLL 2003 respectively.

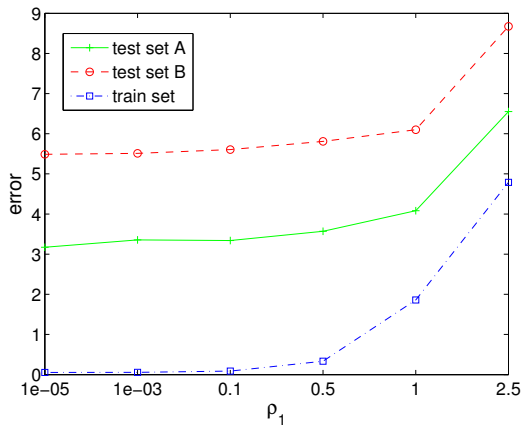


Figure 7.9: CoNLL 2003,  $\rho_2 = 0.001$

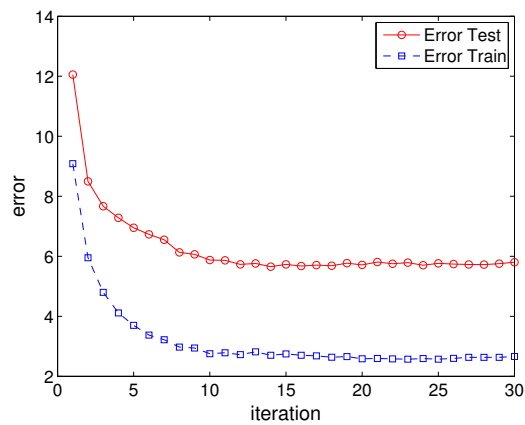


Figure 7.10: CoNLL 2000, error.

We did not notice any drastic influence of the value  $\rho_2$  in our experiments, and let it be rather small, since its role is limited to guarantee the absence of numerical problems. The value of  $\rho_1$  is much more influential. It regulates the number of non-zero parameters, and hence, it influences the performance. Figure 7.9 provides the correspondence between the value of  $\rho_1$  and the error rate for CoNLL 2003. We see that with  $\rho_1$  small enough, in the range  $1e-05 \dots 0.5$ , one reaches an acceptable error rate. For  $\rho_1 = 2.5$  one notices a serious degradation. Figure 7.11 provides the information on the number of active parameters as a function of  $\rho_1$ . It is noteworthy that although the error rate practically does not change for  $\rho_1 = 1e-05 \dots 0.5$ , the number of active parameters decreases. Mainly it is the number of dependencies based on words that decreases, since they are the most numerous, and features based on POS tags and syntactic chunks alone achieve a good generalization. The syntactic tags features are not so numerous, however after 30 iterations we get 46.5% of unigram syntactic parameters, and 16.5% of bigram dependencies.

The number of active parameters in sparse models is still large, therefore it is not possible to make a complete analysis of active parameters. However, we can examine the parameters whose absolute values are of large magnitude. We take into consideration the parameter values associated with bigram and unigram features of the CoNLL 2000 corpus.

Figure 7.12 displays the values of unigram parameters. On Figure 7.12 on the left we provide values of unigram parameters that depend on words. It is not very easy to provide a revealing illustration, because of the large number of words (approximately 20,000). So,

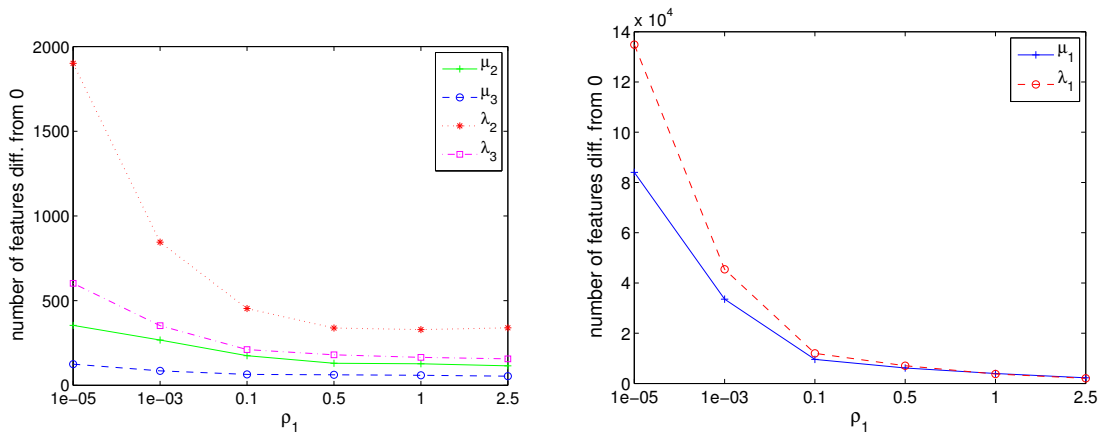


Figure 7.11: CoNLL 2003. Left: number of active parameters that depend on POS tags and syntactic chunks. Right: number of active parameters that depend on words.

Feature	Initial number of parameters	After 30 iterations
$(y_t, x_t^1)$	496,547	5,057
$(y_t, x_t^2)$	1,012	485
$(y_{t-1}, y_t, x_t^1)$	11,917,128	9,439
$(y_{t-1}, y_t, x_t^2)$	24,288	1,591

Table 7.3: Results for CoNLL 2000, with  $\rho_1 = 0.5$ ,  $\rho_2 = 1e - 05$

Feature	Initial number of parameters	After 30 iterations
$(y_t, x_t^1)$	242,320	10,550
$(y_t, x_t^2)$	386	186
$(y_t, x_t^3)$	144	67
$(y_{t-1}, y_t, x_t^1)$	2,180,880	13,585
$(y_{t-1}, y_t, x_t^2)$	3,312	488
$(y_{t-1}, y_t, x_t^3)$	1,296	214

Table 7.4: Results for CoNLL 2003, with  $\rho_1 = 0.1$ ,  $\rho_2 = 1e - 05$

we visualize the most important ones, that are words for which  $\sum_y |\mu_{y,x}| > 5$ . Note, that the parameters having negative values are important as well. Horizontal patterns are typical for this illustration, and can be explained quite easily: the horizontal lines are formed by the most frequent chunks, e.g., 6 - B-NP, 11 - B-VP, 17 - I-NP, 23 - Outside. Figure 7.13 gives information on chunks counts in training and testing data. The most frequent elements are associated with strong patterns. See Appendix C for the description of chunks and POS tags.

Figure 7.12 on the right illustrates the unigram values for POS tags/chunks parameters. The values presented are their absolute values  $|\lambda_{y',y,x}|$ , and at a first glance we notice that there are some horizontal and some much stronger vertical patterns. Figures 7.14 are an attempt to demonstrate the bigram feature values (as a sum over a previous state in order to map into a two-dimensional space). What is here impressive, is the sparsity, especially of POS tags/chunks dependencies. As to the words/chunks dependencies, one can observe the same horizontal lines associated with the most frequent chunks.

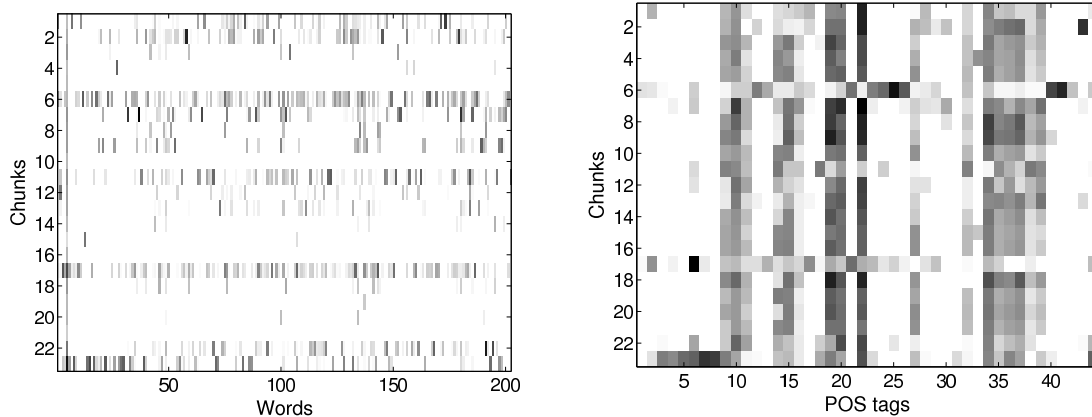


Figure 7.12: CoNLL 2000 ( $\rho_1 = 0.5$ ,  $\rho_2 = 1e - 05$ ). Left: values of unigram parameters that depend on words for which  $\sum_y |\mu_{y,x^1}| > 5$ . Right: values of unigram parameters that depend on POS tags.

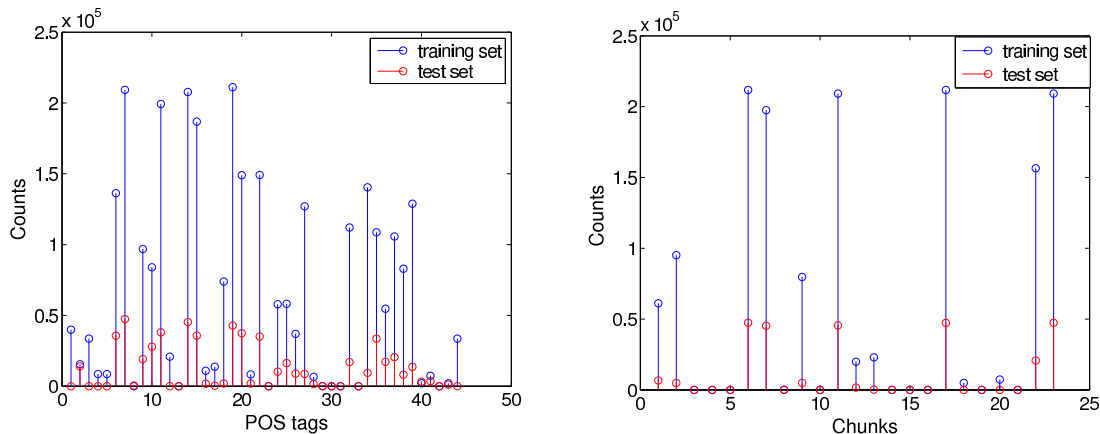


Figure 7.13: CoNLL 2000. Left: Unnormalized frequency of POS tags. Right: Unnormalized frequency of chunks.

Let us consider Figure 7.12 in details and split it into two representations, one for positive and one for negative values illustrated on Figure 7.15. Now it is much more informative. The positive horizontal values are associated, exactly as it is for words, with the most frequent chunks. For example, the chunk B-NP (6) has strong associations with POS tags such as Noun, Proper Noun Singular and Plural, Predeterminer, Possessive Endings, Personal Pronoun, Possessive Pronoun, Wh-determiner, and Wh-pronoun (20-26, 40-41). The chunk Outside (23) leads to strong values when associated with punctuation symbols (3 - 8).

On Figure 7.15, large positive values of the parameters result from high joint frequency of a POS tag and a chunk. Large negative values are associated with parameters whose POS tags are frequent but joint occurrence with a chunk is low. (The counts of POS tags in the training and testing sets are shown on Figure 7.13.)

On the CoNLL 2000 data we carried out one supplementary experiment. We introduced feature functions that do not depend on observations but on labels only. Figure 7.16 demonstrates the positive and negative components of parameter values for transi-

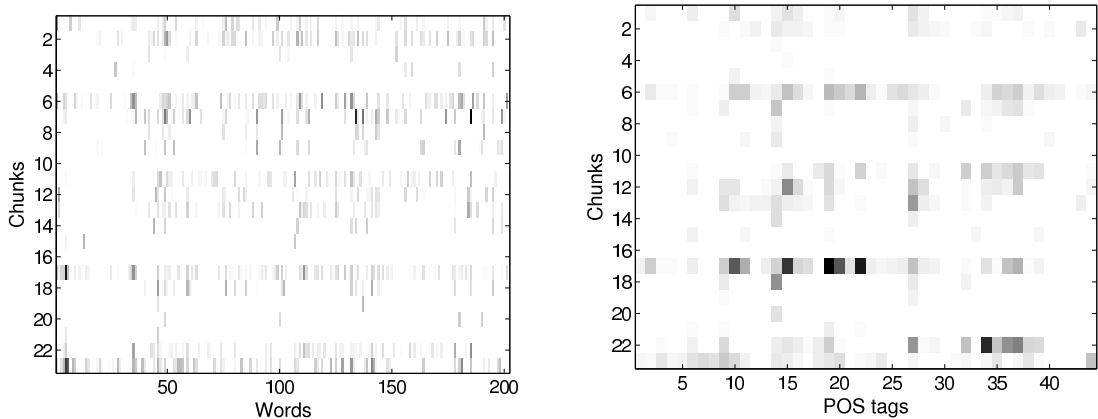


Figure 7.14: CoNLL 2000 ( $\rho_1 = 0.5$ ,  $\rho_2 = 1e - 05$ ). Left: values of bigram parameters ( $\sum_{y_{t-1}} |\lambda_{y_{t-1}, y_t, x_t^1}|$ ) that depend on words for which  $\sum_y |\mu_{y, x^1}| > 5$ . Right: values of bigram parameters ( $(\sum_{y_{t-1}} |\lambda_{y_{t-1}, y_t, x_t^2}|)$ ) that depend on POS tags.

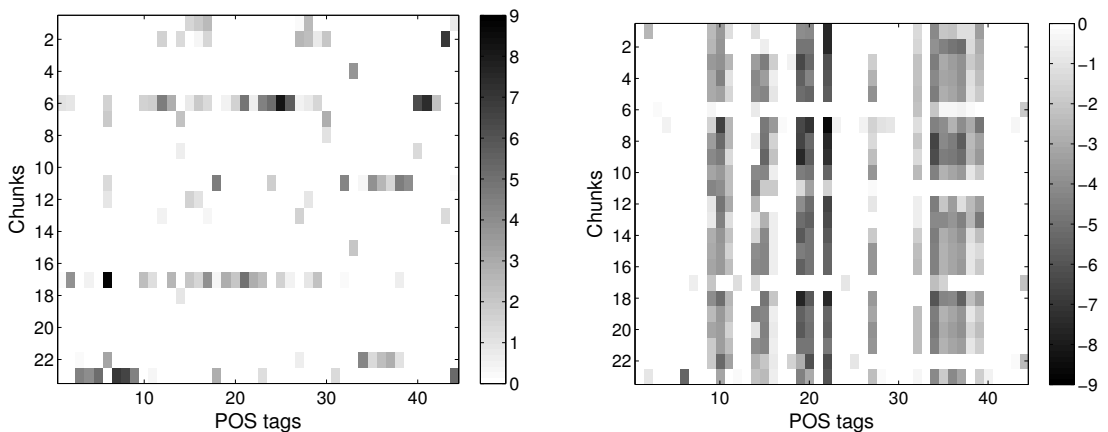


Figure 7.15: CoNLL 2000. Left: positive unigram POS parameters. Right: negative unigram POS parameters.

tions  $(y_{t-1}, y_t)$ . Let us consider the maximal positive values from Figure 7.16 on the left. The strongest transitions are connected with the rare deterministic transitions, such as Begin List Marker - Inside of Interjection (4 - 15), Begin of Unlike Coordinated Phrase - Inside of Unlike Coordinated Phrase (10 - 21), Inside of Interjection - Inside of Interjection (15 - 15), Inside of Particles - Inside of Particles (19 -19). Strong but not deterministic values have Begin Noun Phrase - Inside Noun Phrase (6 - 17) and Begin Verbal Phrase - Inside Verbal Phrase (11 - 22).

The negative transitions on Figure 7.16 (on the right) can be interpreted as prohibited previous state/current state transitions. For example, transitions starting from I-NP (17) to the following chunks are very unlikely: Begin of Prepositional Clause (7), Begin of Particles (8), Begin of Unlike Coordinated Phrase (10), Begin of Verb Phrase (11), Inside of Adjective Phrase (12), Inside of Adverb Phrase (13), Inside of Conjunction Phrase (14), and Inside of Verb Phrase (22).

We skip the detailed analysis of parameters based on words, since they are too numer-

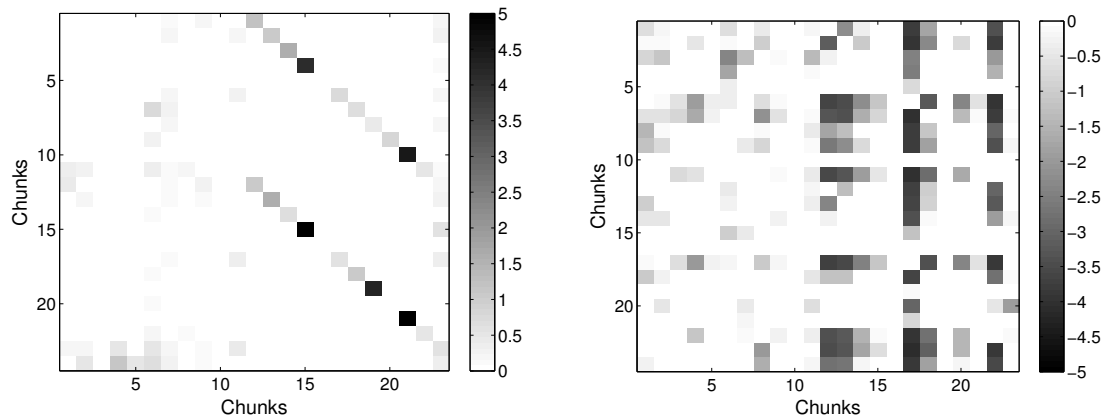


Figure 7.16: CoNLL 2000. Left: positive  $(y_{t-1}, y_t)$  parameters. Right: negative  $(y_{t-1}, y_t)$  parameters.

ous. Let us, however, look at the most influential words, i.e. the ones that are involved in selected parameters and whose absolute weights are the maximal ones (Tables 7.5 and 7.6 for CoNLL 2000 and CoNLL 2003). The set of the most influential words is connected to the task. For CoNLL 2000, we get mostly functional words that define the structure of language. For the CoNLL 2003 corpus, on the contrary, we get a lot of adjectives and nouns, that are parts of some named entities. Note that the most important lexical items of CoNLL 2003 have low frequency and the associated parameter values are negative, with large absolute values.

about	after	and	as	but	depressed
down	due	following	for	half	n't
if	in	including	is	like	not
now	of	off	on	or	out
pending	rather	right	tax	that	the
times	to	today	up	whether	while

Table 7.5: CoNLL 2000, words with the maximal absolute unigram values,  $\sum_y |\mu_{y,x^1}| > 10$

Afghan	African	Albanian	American	Australian
Belgian	Bosnian	British	Cup	Democratic
Democrats	Dutch	English	European	French
Frenchman	German	Indian	Israel	Italian
July	June	Kurdish	Lebed	London
Men	Mickelson	Nepal	Nigerian	OSCE
Olympics	Palestinian	President	Regulation	Republicans
Russian	Sampras	September	Stansted	Treasury
Turkish	U.S.	Wednesday	Western	Wimbledon

Table 7.6: CoNLL 2003, words with the maximal absolute unigram values,  $\sum_y |\mu_{y,x^1}| > 7$

To illustrate the efficiency of  $L_1$ -based feature selection, we compare it to three simple minded approaches of feature selection, which are often used in practice. The first one, termed “cut-off”, consists in incorporating only those features that have been observed

sufficiently often in the training corpus. This amounts to deleting *a priori* all the rare dependencies. The second option consists in training a model that is not sparse (e.g., with an  $L_2$  penalty term) eliminating, *a posteriori*, all parameters whose values are not of sufficient magnitude. Another method to perform a pre-selection is based on mutual information. The methods have already been mentioned in Section 6.2.1.

Figure 7.17 compares the error rates and Figure 7.18 F-measures obtained with these strategies on the CoNLL 2003 data set to those achieved by the SBCD and OWL-QN algorithms. Obviously, the *a priori* cut-off strategy is very poor. The *a posteriori* thresholding strategy is more efficient but cannot be used to obtain well-performing models that are very sparse (here, with less than 10,000 features).

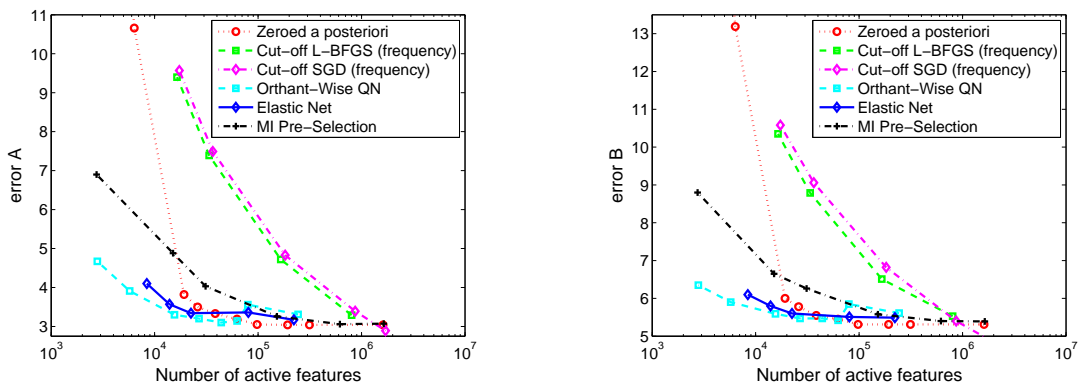


Figure 7.17: CoNLL 2003 Data. Dependence of performance on the number of active features. Left: on set Test A, right: on set Test B.

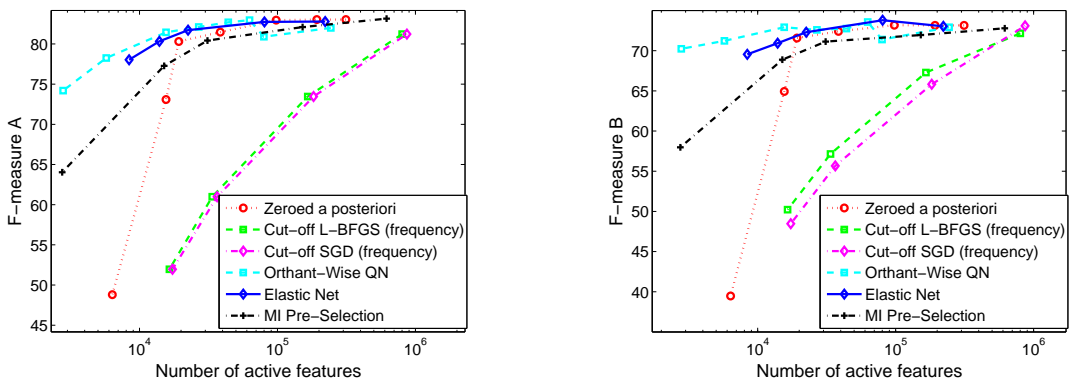


Figure 7.18: CoNLL 2003 Data. Dependence of F-measure on the number of active features. Left: on set Test A, right: on set Test B.

In this experiment, SBCD is less efficient from a computational point of view compared to the phonetisation task considered in Section 7.3.2 as the number of blocks is of the same order as the number of training sequences and, in addition, the sparse forward-backward implementation is less efficient than in the case of the phonetisation task as the number of labels is much smaller: SBCD needs 42 minutes (with  $\rho_1 = 1$ , corresponding to 6,656 active features) to achieve a reasonable performance while OWL-QN is faster, taking about 5 minutes to converge. If sparsity is not needed, SGD appears to be the most



efficient method for this corpus as it converges in less than 4 minutes. L-BFGS in contrast requires about 25 minutes to reach a similar performance.

## 7.4 CONCLUSIONS

---

We have proposed to apply the elastic net penalty term to CRFs. The benefits of working with sparse parameter vectors are twofold: obviously, less parameters need be computed and stored; more importantly, sparsity can be used to speed up the forward-backward and the Viterbi algorithms.

The  $\rho_2$  parameter ensures that there are no numerical problems. The heuristic presented as equation (7.4) guarantees the stability. With  $\kappa_1 = 1$  and  $\kappa_2 = 2$ , as in our experiments, the block coordinate-wise descent does not suffer from numerical problems, the  $L_2$  penalty term does not practically influence the performance.

To make the method feasible, we have introduced and validated the approximation that consists in ignoring the off-diagonal terms of the Hessian of the objective function and which allows to reduce the computational load through blockwise gradient descent. This method has been tested on artificial and real-world data, yielding accuracy that is comparable with conventional training algorithms, and much sparser parameter vectors.

The results achieved open several avenues that we wish to explore in the future. A first extension of this work is related to finding the optimal weights for the penalization terms, a task that is usually achieved through heuristic search for the value(s) that will deliver the best performance on a development set. Based on our experiments, this search can be performed efficiently using pseudo regularization-path techniques, which amount here to starting the tuning with a very constrained model, and to progressively reduce the weight of the  $L_1$  term so as to increase the number of active features. This can be performed effectively at very little cost by restarting the coordinate-wise optimization from the parameter values obtained with the previous weights setting, thereby greatly reducing the number of iterations needed to reach convergence.

A second line of research, aiming at improving the training speed, is based on the observation that the number of active features stabilizes very quickly, typically in a dozen iterations or so. This suggests that those features that are inactive at that stage will remain inactive till the convergence of the procedure. Instead of repeatedly trying to recompute the gradient for those features, one might well decide to zero them once and for all. Some encouraging results have been obtained on the artificial data set considered in Section 7.3.1.

Another interesting observation is that the parameter vector is not only sparse, but that the sparsity patterns are closely correlated with the structure of the feature set: as discussed above, in Chapter 6, bigram features testing label pairs tend to be active only when the corresponding unigram feature is significant. There might be other ways to take advantage from this observation, such as, for instance, hierarchical penalties introduced in (Zhao et al., 2009), or growing a model by progressively introducing high-order features when the corresponding low-order have proved useful.

---

## CHAPTER 8

# CONCLUSIONS AND FUTURE DIRECTIONS

---

In this thesis, we studied two significant strata of modern machine learning, semi-supervised learning and model selection, in the context of conditional random fields.

In the first part of our thesis, we have made an attempt to show the importance of statistical and probabilistic methods for machine learning, above all the importance of approaches that allow to take structure into account.

In the framework of semi-supervised learning, we presented our semi-supervised estimator that allows to introduce unlabeled data into discriminative models under the form of marginal probability of observations. We provide proofs that the proposed semi-supervised estimator is asymptotically optimal, and illustrate its functioning and its competitiveness with the logistic regression, both on the artificial and real-world problem of binary classification.

The next step would be a generalization of the semi-supervised estimator to sequential tasks which can be solved with conditional random fields. Here, we face another challenge that is to estimate or to approximate probability or importance weights of sequences. Another question is how to introduce both marginal probabilities: one of observations and marginal probabilities of labels. The introduction of probabilities of labels has been already considered in, e.g., (Mann and McCallum, 2007b). Another avenue concerned with importance weights is active learning. Kanamori and Shimodaira (2003) made an attempt to attack active learning tasks with the maximum weighted log-likelihood estimator.

We noticed that the non-asymptotic analysis of the semi-supervised estimator, i.e., for cases when the number of labeled observations is very small, could give much more information not only on the proposed semi-supervised algorithm, but also on related issues.

Another problem considered in this thesis is model selection. Empirical experiments on real data show that a huge number of extracted patterns can be eliminated from the model without degrading the performance. We applied the elastic net criterion, which is a combination of the  $L_1$  and  $L_2$  norms, to conditional random fields, where the  $L_1$  norm is responsible for sparsity, and the  $L_2$  norm is introduced to protect the optimization from numerical problems.

The optimization approach is the coordinate-wise gradient descent, which we approxi-

mated by the block-wise gradient descent in order to speed up the optimization procedure. In our applications, the choice of blocks was naive and natural at the same time. A block corresponds to all features associated with an observation. However, the possibility to form more practical blocks, groups, and integrate hierarchical dependencies is still an important open problem, although some research has been made, see e.g., (Szafranski et al., 2007), (Meier et al., 2008), and (Zhao et al., 2009). The approximation of the second derivative is worth studying much deeper than what we did. A regularisation path, i.e. the set of solutions as a function of  $\rho_1$  can be performed at low cost. We can progressively reduce the weight of the  $L_1$  term to increase the number of active features by restarting the blockwise optimization based on parameter values obtained from the previous setting.

Nowadays, the majority of algorithms that return sparse solutions cycle over all parameters at every iteration. In this thesis, we discussed an approach to speed up the optimization, namely, the forward-backward procedure for sparse vectors of parameters. Another idea which was mentioned but not deeply studied, is to avoid cycling over all parameters.

We analyzed the results of our model selection experiments and examined the parameters which have been chosen as significant ones by the elastic net conditional random fields. The values appeared to be interpretable and pertinent, illustrating also the fact that the never observed configurations are as important as the observed ones.

In the context of structured output prediction, the feature engineering is not to be ignored. A literature survey shows that the feature choice resulting in a baseline performance is task-independent, but the domain knowledge plays a significant role for a good accuracy. The complexity reduction in conditional random fields is an important issue which is an active field of research (Qian et al., 2009). First of all, it concerns richer structural dependencies. When elaborated features are important, and how to avoid the increase in complexity, is still an open problem.

There are some approaches that are not considered but related to the thesis and therefore they should be explored in the future. Among them are, e.g., methods that allow to select training instances, such as active learning. Feature kernelization, which is not investigated in the thesis, is worth studying to perform a low-dimensional mapping (Balcan et al., 2006).

---

## APPENDIX A

# ASYMPTOTIC PERFORMANCE OF THE SEMI-SUPERVISED ESTIMATOR FOR $K$ -CLASSES LOGISTIC REGRESSION

---

Here, we follow the simplified notations  $g(k|X; \theta_\star) = g_k$  and  $\eta(k|X) = \eta_k$ .

For the  $K$ -class logistic regression without covariate shift:

$$I(\theta_\star) = \mathbb{E}_q \left[ XX^T \otimes \begin{pmatrix} (\eta_1 - g_1)^2 + \eta_1(1 - \eta_1) & \cdots & (g_1 g_{k-1} - \eta_1 \eta_{k-1}) - \eta_1 \eta_{k-1} \left( \frac{g_{k-1}}{\eta_{k-1}} + \frac{g_1}{\eta_1} - 1 \right) \\ \vdots & \ddots & \vdots \\ (g_1 g_{k-1} - \eta_1 \eta_{k-1}) - \eta_1 \eta_{k-1} \left( \frac{g_{k-1}}{\eta_{k-1}} + \frac{g_1}{\eta_1} - 1 \right) & \cdots & (\eta_{k-1} - g_{k-1})^2 + \eta_{k-1}(1 - \eta_{k-1}) \end{pmatrix} \right],$$

$$J(\theta_\star) = \mathbb{E}_q \left[ XX^T \otimes \begin{pmatrix} g_1(1 - g_1) & \cdots & -g_1 g_{k-1} \\ \vdots & \ddots & \vdots \\ -g_1 g_{k-1} & \cdots & g_{k-1}(1 - g_{k-1}) \end{pmatrix} \right].$$

For the proposed semi-supervised estimator:

$$I(\theta_\star) = \mathbb{E}_q \left[ XX^T \otimes \begin{pmatrix} \eta_1(1 - \eta_1) & \cdots & -\eta_1 \eta_{k-1} \\ \vdots & \ddots & \vdots \\ -\eta_1 \eta_{k-1} & \cdots & \eta_{k-1}(1 - \eta_{k-1}) \end{pmatrix} \right],$$

$$J(\theta_\star) = \mathbb{E}_q \left[ XX^T \otimes \begin{pmatrix} g_1(1 - g_1) & \cdots & -g_1 g_{k-1} \\ \vdots & \ddots & \vdots \\ -g_1 g_{k-1} & \cdots & g_{k-1}(1 - g_{k-1}) \end{pmatrix} \right].$$

For the Shimodaira criterion under the covariate shift:

$$I(\theta_\star) = \mathbb{E}_{q_0} \left[ \frac{q_1^2(X)}{q_0^2(X)} XX^T \otimes \begin{pmatrix} (\eta_1 - g_1)^2 + \eta_1(1 - \eta_1) & \cdots & (g_1 g_{k-1} - \eta_1 \eta_{k-1}) - \eta_1 \eta_{k-1} \left( \frac{g_{k-1}}{\eta_{k-1}} + \frac{g_1}{\eta_1} - 1 \right) \\ \vdots & \ddots & \vdots \\ (g_1 g_{k-1} - \eta_1 \eta_{k-1}) - \eta_1 \eta_{k-1} \left( \frac{g_{k-1}}{\eta_{k-1}} + \frac{g_1}{\eta_1} - 1 \right) & \cdots & (\eta_{k-1} - g_{k-1})^2 + \eta_{k-1}(1 - \eta_{k-1}) \end{pmatrix} \right],$$

$$J(\theta_\star) = \mathbb{E}_{q_0} \left[ \frac{q_1(X)}{q_0(X)} XX^T \otimes \begin{pmatrix} g_1(1 - g_1) & \cdots & -g_1 g_{k-1} \\ \vdots & \ddots & \vdots \\ -g_1 g_{k-1} & \cdots & g_{k-1}(1 - g_{k-1}) \end{pmatrix} \right].$$

For our semi-supervised criterion under covariate shift:

$$I(\theta_\star) = \mathbb{E}_{q_0} \left[ \frac{q_1^2(X)}{q_0^2(X)} XX^T \otimes \begin{pmatrix} \eta_1(1 - \eta_1) & \cdots & -\eta_1 \eta_{k-1} \\ \vdots & \ddots & \vdots \\ -\eta_1 \eta_{k-1} & \cdots & \eta_{k-1}(1 - \eta_{k-1}) \end{pmatrix} \right],$$

$$J(\theta_\star) = \mathbb{E}_{q_0} \left[ \frac{q_1(X)}{q_0(X)} XX^T \otimes \begin{pmatrix} g_1(1 - g_1) & \cdots & -g_1 g_{k-1} \\ \vdots & \ddots & \vdots \\ -g_1 g_{k-1} & \cdots & g_{k-1}(1 - g_{k-1}) \end{pmatrix} \right].$$

---

## APPENDIX B

# NETTALK CORPUS

---

The original Nettetalk corpus has been introduced in (Sejnowski and Rosenberg, 1987). The Nettetalk corpus we use for our experiments has been suggested for the Pascal Letter-to-Phoneme Conversion Challenge<sup>1</sup>. The English data set contains 16280 words aligned with their phonetical transcriptions. The corpus is split into 10 parts, each of which includes 1628 sequences of observations and corresponding labels.

We provide the Table of the correspondence of Nettetalk phonetical symbols with international phonetic alphabet.

Associated Number	IPA	Nettalk Symbol	Example
1		/./	empty sound
2	[gz]	/1/	<b>ex</b> act
3	[ʒ]	/2/	mea <b>su</b> re
4	[ε]	/3/	elab <b>or</b> ation
5	[ə]	/A/	<b>a</b> bove
6	[b]	/B/	<b>b</b> at
7	[d]	/D/	am <b>e</b> nd
8	[e]	/E/	se <b>t</b>
9	[f]	/F/	<b>f</b> ine
10	[g]	/G/	<b>g</b> ot
11	[h]	/H/	<b>h</b> at
12	[i]	/I/	pi <b>t</b> y
13	[dʒ]	/J/	<b>j</b> ust
14	[k]	/K/	<b>k</b> iss
15	[əL]	/L/	typ <b>ic</b> al
16	[m]	/M/	ra <b>m</b>
17	[n]	/N/	<b>n</b> ut
18	[ɔ]	/O/	wa <b>sh</b>
19	[p]	/P/	<b>p</b> ate
20	[r]	/R/	<b>r</b> an
21	[s]	/S/	si <b>t</b>
22	[t]	/T/	<b>t</b> able

---

<sup>1</sup><http://pascallin.ecs.soton.ac.uk/Challenges/PRONALSYL/Datasets/>

Associated Number	IPA	Nettalk Symbol	Example
23	[ʌ]	/U/	fun
24	[v]	/V/	vine
25	[w]	/W/	wet
26	[y]	/Y/	yet
27	[z]	/Z/	buzz
28	[ə:]	/a/	burn
29	[tʃ]	/b/	chance
30	[æ]	/c/	apple
31	[ʊ]	/d/	full
32	[ɪ]	/e/	rang
33	[əʊ]	/f/	ago
34	[i:]	/g/	see
35	[ei]	/h/	fate
36	[au]	/i/	now
37	[ai]	/j/	lie
38	[ʃ]	/k/	ship
39	[l]	/l/	little
40	[a:]	/m/	calm
41	[ks]	/n/	convex
42	[ɔ:]	/o/	born
43	[yʊ]	/p/	curlew
44	[ð]	/q/	this
45	[i]	/r/	harmony
46	[kw]	/s/	quit
47	[ɔi]	/t/	voice
48	[kʃ]	/u/	anxious
49	[ʊə]	/v/	jury
50	[θ]	/w/	math
51	[iə]	/x/	near
52	[ʊ:]	/y/	boon
53	[ɛə]	/z/	tare

---

## APPENDIX C

# CoNLL 2000 AND CoNLL 2003 DATA SETS

---

### CoNLL 2000

The Conference on Computational Natural Language Learning 2000 challenge was introduced in (Tjong Kim Sang and Buchholz, 2000). The purpose is to chunk already divided into syntactically correlated groups, e.g.,

He	reckons	the	current	account	deficit	will	narrow	...
B-NP	B-VP	B-NP	I-NP	I-NP	I-NP	B-VP	I-VP	...

The labels are chunks providing information whether a word is the first word of a group X (chunk B-X), or is inside of a group X (I-X). Words that do not belong to any group, in other words, outside of any group, and are labelled with O. There are 11 types of groups, therefore, there 23 ( $2 \times 11 + 1$ ) chunks (see Table C.2). The data contains two types of observations: lexical items and their part-of-speech tags (see Table C.1) derived by the Brill tagger. The labels have been extracted from the PennTreeBank. The size of the lexical items dictionary of the CoNLL 2000 corpus is 21589.

### CoNLL 2003

Named entity recognition consists in extracting groups of syntagms that correspond to named entities (e.g., names of persons, organizations, places, etc.). The data used for our experiments are taken from the CoNLL 2003 challenge (Tjong Kim Sang and de Meulder, 2003) and implies four distinct types of named entities, and 8 labels. Labels have the form B-X or I-X, that is begin or inside of a named entity X (however, the label B-PER is not present in the corpus). Words that are not included in any named entity, are labeled with O (outside), e.g.,

U.N.	official	Ekeus	heads	for	Baghdad
B-ORG	O	B-PER	O	O	B-LOC

At each position in the text, the input consists of three separate components: a word (with 30290 distinct words in the corpus), its part-of-speech(44), and syntactic (18) tags.

See Table C.3 for the details on the CoNLL 2000 and 2003 English corpora.



Associated number	Abbreviation	Explanation
1	#	
2	\$	
3	"	
4	(	
5	)	
6	,	
7	.	
8	:	
9	CC	Coordinating conjunction
10	CD	Cardinal Number
11	DT	Determiner
12	EX	Existential "there"
13	FW	Foreign word
14	IN	Preposition or subordinating conjunction
15	JJ	Adjective
16	JJR	Adjective, comparative
17	JJS	Adjective, superlative
18	MD	Modal
19	NN	Noun, singular or mass
20	NNP	Proper noun, singular
21	NNPS	Proper noun, plural
22	NNS	Noun, plural
23	PDT	Predeterminer
24	POS	Possessive ending
25	PRP	Personal pronoun
26	PRP\$	Possessive pronoun
27	RB	Adverb
28	RBR	Adverb, comparative
29	RBS	Adverb, superlative
30	RP	Particle
31	SYM	Symbol
32	TO	"to"
33	UH	Interjection
34	VB	Verb, base form
35	VBD	Verb, past tense
36	VBG	Verb, gerund or present participle
37	VBN	Verb, past participle
38	VBP	Verb, non-3rd person singular present
39	VBZ	Verb, 3rd person singular present
40	WDT	Wh-determiner
41	WP	Wh-pronoun
42	WP\$	Possessive wh-pronoun
43	WRB	Wh-adverb
44	"	

Table C.1: Part of Speech Tags.

Associated number	Abbreviation	Explanation
1	B-ADJP	Begin of Adjective Phrase
2	B-ADVP	Begin of Adverb Phrase
3	B-CONJP	Begin of Conjunction Phrase
4	B-INTJ	Begin of Interjection
5	B-LST	List Marker
6	B-NP	Begin of Noun Phrase
7	B-PP	Begin of Prepositional Phrase
8	B-PRT	Begin of Particles
9	B-SBAR	Begin of Subordinated Clause
10	B-UCP	Begin of Unlike Coordinated Phrase
11	B-VP	Begin of Verb Phrase
12	I-ADJP	Inside of Adjective Phrase
13	I-ADVP	Inside of Adverb Phrase
14	I-CONJP	Inside of Conjunction Phrase
15	I-INTJ	Insider of Interjection
16	I-LST	Inside of List Marker
17	I-NP	Inside of Noun Phrase
18	I-PP	Inside of Prepositional Phrase
19	I-PRT	Inside of Particles
20	I-SBAR	Inside of Subordinated Clause
21	I-UCP	Inside of Unlike Coordinated Phrase
22	I-VP	Inside of Verb Phrase
23	O	Outside

Table C.2: Chunks of CoNLL 2000.

	Chunking		Named Entities		
	Phrases	Tokens	Articles	Phrases	Tokens
Entraînement	9,836	211,727	946	14,987	203,621
Developpement	-	-	216	3,466	51,362
Test	2012	47,377	231	3,684	46,435

Table C.3: Corpora CoNLL 2000 and CoNLL 2003 details.



---

## APPENDIX D

# EXPRESSION OF THE FULL HESSIAN FOR THE BLOCK OF PARAMETERS ASSOCIATED WITH $\mu_{y,x}$ AND $\lambda_{y',y,x}$

---

The off diagonal terms of the Hessian  $\partial^2 \ell(\mathcal{D}, \theta) / \partial \theta_j \partial \theta_k$  are approximated. One replaces  $f_k^2(y_{t-1}, y_t, x_t^{(i)})$  by  $f_j(y_{t-1}, y_t, x_t^{(i)}) f_k(y_{t-1}, y_t, x_t^{(i)})$  and the final squared term by

$$\mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_j(y_{t-1}, y_t, x_t^{(i)}) \times \mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_k(y_{t-1}, y_t, x_t^{(i)}).$$

Therefore, we have

$$\frac{\partial^2 \ell(\mathcal{D}; \theta)}{\partial \theta_k \partial \theta_l} \approx \sum_{i=1}^N \sum_{t=1}^{T_i} \sum_{k, l \in K} \begin{cases} p_k(\mathbf{y}|\mathbf{x}_t^{(i)})(1 - p_k(\mathbf{y}|\mathbf{x}_t^{(i)})), & k = l, \\ -p_k(\mathbf{y}|\mathbf{x}_t^{(i)})p_l(\mathbf{y}|\mathbf{x}_t^{(i)}), & k \neq l. \end{cases}$$

The full Hessian for a block of parameters associated with  $\mu_{y,x}$  and  $\lambda_{y',y,x}$  takes the following form:

$$H_g^{(i)} = \begin{pmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C} & \mathbf{B} \end{pmatrix},$$

where

$$\begin{aligned} \mathbf{A} &= H_{g_{1:|Y|, 1:|Y|}}^{(i)} = \begin{cases} -\mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_k(y_t, x_t^{(i)})(1 - \mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_k(y_t, x_t^{(i)})), & \text{if } k = l, \\ \mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_k(y_t, x_t^{(i)}) \mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_l(y_t, x_t^{(i)}), & \text{if } k \neq l, \end{cases} \\ \mathbf{B} &= H_{g_{|Y|+1:|Y|+|Y^2|, |Y|+1:|Y|+|Y^2|}}^{(i)} \\ &= \begin{cases} -\mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_k(y_{t-1}, y_t, x_t^{(i)})(1 - \mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_k(y_{t-1}, y_t, x_t^{(i)})), & \text{if } k = l, \\ \mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_k(y_{t-1}, y_t, x_t^{(i)}) \mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_l(y_{t-1}, y_t, x_t^{(i)}), & \text{if } k \neq l, \end{cases} \\ \mathbf{C} &= H_{g_{1:|Y|, |Y|+1:|Y|+|Y^2|}}^{(i)} = H_{g_{|Y|+1:|Y|+|Y^2|, 1:|Y|}}^{(i)} \\ &= \begin{cases} \mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_k(Y_t = a, x_t^{(i)}) \mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_l(y_{t-1}, Y_t = b, x_t^{(i)}), & \text{if } a \neq b, \\ \mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_l(y_{t-1}, Y_t = b, x_t^{(i)}) \\ \quad - \mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_k(Y_t = a, x_t^{(i)}) \mathbb{E}_{p_{\theta}(\mathbf{y}|\mathbf{x}^{(i)})} f_l(y_{t-1}, Y_t = b, x_t^{(i)}), & \text{if } a = b. \end{cases} \end{aligned}$$



---

## BIBLIOGRAPHY

---

- Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Proc. 2nd International Symposium on Information Theory*.
- Altun, Y., Johnson, M., and Hofmann, T. (2003). Investigating loss functions and optimization methods for discriminative learning of label sequences. In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)*, Geneva, Switzerland.
- Altun, Y., McAllester, D., and Belkin, M. (2005). Maximum margin semi-supervised learning for structured variables. In *Advances in Neural Information Processing Systems (NIPS)*.
- Andrew, G. and Gao, J. (2007). Scalable training of l1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning (ICML)*, pages 33–40, Corvallis, Oregon.
- Bach, F. (2006). Active learning for misspecified generalized linear models. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS)*, volume 19.
- Bakir, G. H., Hofmann, T., Schölkopf, B., Smola, A. J., Taskar, B., and Vishwanathan, S. V. N., editors (2007). *Predicting Structured Data*. Neural Information Processing. The MIT Press.
- Balcan, M.-F., Blum, A., and Vempala, S. (2006). Kernels as features: On kernels, margins, and low-dimensional mappings. *Mach. Learn.*, 65(1):79–94.
- Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Statist.*, 41(1):164–171.
- Benajiba, Y. and Rosso, P. (2008). Arabic named entity recognition using conditional random fields. In *Arabic Language and local languages processing: Status Updates and Prospects, 6th Int. Conf. on Language Resources and Evaluation*.
- Bender, O., Och, F. J., and Ney, H. (2003). Maximum entropy models for named entity recognition. In *Proceedings of CoNLL-2003*, pages 148–151, Edmonton, Canada.
- Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71.
- Besag, J. (1975). Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195.

- Bickel, S., Brückner, M., and Scheffer, T. (2007). Discriminative learning for differing training and test distributions. In *ICML*.
- Bilmes, J. A. (1998). A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report TR-97-021, Department of Electrical Engineering and Computer Science U.C. Berkeley.
- Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, New York, NY, USA.
- Blunsom, P. (2004). Maximum entropy markov models for semantic role labelling. In *Proceedings of the Australasian Language Technology Workshop*.
- Blunsom, P. and Cohn, T. (2006). Discriminative word alignment with conditional random fields. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 65–72, Morristown, NJ, USA.
- Bottou, L. (1991). *Une Approche théorique de l'Apprentissage Connexionniste: Applications à la Reconnaissance de la Parole*. PhD thesis, Université de Paris XI, Orsay, France.
- Bottou, L. (2004). Stochastic learning. In Bousquet, O. and von Luxburg, U., editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, Berlin.
- Bottou, L. (2007). Stochastic gradient descent (SGD) implementation.
- Bouchard, G. and Triggs, B. (2004). The trade-off between generative and discriminative classifiers. In *IASC 16th International Symposium on Computational Statistics*, pages 721 – 728.
- Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge, UK.
- Brefeld, U. and Scheffer, T. (2006). Semi-supervised learning for structured output variables. In *Proceedings of the 23rd international conference on Machine learning (ICML)*, pages 145 – 152.
- Cappé, O. and Moulines, E. (2005). Recursive computation of the score and observed information matrix in hidden Markov models. In *IEEE Workshop on Statistical Signal Processing (SSP)*, Bordeaux, France.
- Cappé, O., Moulines, E., and Rydén, T. (2005). *Inference in Hidden Markov Models*. Springer.
- Carreras, X., Màrques, L., and Padró, L. (2002). Named entity extraction using adaboost. In *Proceedings of CoNLL-2002*, pages 167–170. Taipei, Taiwan.
- Carreras, X. and Màrquez, L. (2003). Phrase recognition by filtering and ranking with perceptrons. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*.

- Castelli, V. and Cover, T. (1996). The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory*, 42(6):2102 – 2117.
- Chapelle, O., Schölkopf, B., and Zien, A. (2006). *Semi-Supervised Learning*. MIT Press, Cambridge, MA.
- Chapelle, O. and Zien, A. (2005). Semi-supervised classification by low density separation. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*.
- Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan.
- Chen, S. F. and Rosenfeld, R. (2000). A survey of smoothing techniques for maximum entropy models. *IEEE transactions on Speech and Audio Processing*, 8(2):37–50.
- Chen, X., Chen, S., and Xiao, K. (2008). K-similar conditional random fields for semi-supervised sequence labeling. In *Advanced Language Processing and Web Information Technology*, pages 21–26.
- Cohen, I., G.Cozman, F., Sebe, N., C.Cirelo, M., and Huang, T. S. (2004). Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction. *IEEE Transactions on Pattern Analysis and Machine Learning*, 26(12):1553–1567.
- Cohn, T. (2006). Efficient inference in large conditional random fields. In *Proceedings of the 17th European Conference on Machine Learning*, pages 606–613, Berlin.
- Cohn, T. and Blunsom, P. (2005). Semantic role labelling with tree conditional random fields. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 169–172, Ann Arbor, Michigan. Association for Computational Linguistics.
- Cohn, T., Smith, A., and Osborne, M. (2005). Scaling conditional random fields using error-correcting codes. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 10–17, Ann Arbor, Michigan.
- Collins, M. and Duffy, N. (2002). New ranking algorithms for parsing and tagging: kernels over discrete structures and the voted perceptron. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 489–496, Philadelphia, PA.
- Collins, M., Globerson, A., Koo, T., Carreras, X., and Bartlett, P. L. (2008). Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. *J. Mach. Learn. Res.*, 9:1775–1822.
- Corduneanu, A. and Jaakkola, T. (2003). On information regularization. In *In the Proceedings of the 19th conference on Uncertainty in Artificial Intelligence (UAI)*.



- Cortes, C., Mohri, M., Riley, M., and Rostamizadeh, A. (2008). Sample selection bias correction theory. In *In Proceedings of The 19th International Conference on Algorithmic Learning Theory*.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273.
- Culotta, A., Kulp, D., and McCallum, A. (2005). Gene prediction with conditional random fields. Technical Report UM-CS-2005-028, University of Massachusetts, Amherst.
- Culotta, A., McCallum, A., and Betz, J. (2006). Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 296 – 303.
- Darroch, J. N. and Ratcliff, D. (1972). Generalized iterative scaling for log-linear models. *Ann. Math. Statist.*, 43(5):1470 – 1480.
- Daumé III, H. (2009). Semi-supervised or semi-unsupervised? In *NAACL Workshop on Semi-supervised Learning for NLP*.
- Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *23rd International Conference on Machine Learning (ICML)*.
- Della Pietra, S., Della Pietra, V. J., and Lafferty, J. D. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393.
- Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of The Royal Statistical Society Series B*, 39(1):1–38.
- Denis, F., Gilleron, R., Laurent, A., and Tommasi, M. (2003). Co-training from positive and unlabeled examples. In *Proceedings of the ICML Workshop: the Continuum from Labeled Data to Unlabeled Data in Machine Learning and Data Mining*, pages 80 – 87.
- DeRose, S. J. (1988). Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 2(14):31–39.
- Dietterich, T. G., Ashenfelder, A., and Bulatov, Y. (2004). Training conditional random fields via gradient tree boosting. In *ICML, Banff, Canada*.
- Dudík, M., Phillips, S. J., and Schapire, R. E. (2004). Performance guarantees for regularized maximum entropy density estimation. In Shawe-Taylor, J. and Singer, Y., editors, *Proceedings of the 17th annual Conference on Learning Theory, (COLT 2004), Banff, Canada*, volume 3120 of *Lecture Notes in Computer Science*, pages 472–486. Springer.
- Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, 2(32):407–499.
- Elworthy, D. (1994). Does Baum-Welch re-estimation help taggers? In *Proceedings of the 4th Conference on Applied Natural Language Processing*.
- Finkel, J. R., Kleeman, A., and Manning, C. D. (2008). Efficient, feature-based, conditional random field parsing. In *Proceedings of ACL-08: HLT*, pages 959–967, Columbus, Ohio.

- Florian, R., Ittycheriah, A., Jing, H., and Zhang, T. (2003). Named entity recognition through classifier combination. In Daelemans, W. and Osborne, M., editors, *Proceedings of CoNLL-2003*, pages 168–171. Edmonton, Canada.
- Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the thirteenth International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann.
- Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332.
- Friedman, J., Hastie, T., and Tibshirani, R. (2008). Regularization paths for generalized linear models via coordinate descent. Technical report, Department of Statistics, Stanford University.
- Galley, M. (2006). A skip-chain conditional random field for ranking meeting utterances by importance. In *EMNLP*.
- Goel, V. and Byrne, W. J. (2000). Minimum Bayes-risk automatic speech recognition. *Computer Speech and Language*, 14(2):115–135.
- Grandvalet, Y. and Bengio, Y. (2004). Semi-supervised learning by entropy minimization. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS)*, volume 17, pages 529–536.
- Gross, S. S., Russakovsky, O., Do, C. B., and Batzoglu, S. (2006). Training conditional random fields for maximum labelwise accuracy. In *Advances in Neural Information Processing Systems*, volume 19.
- Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.
- Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer.
- Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409 – 437.
- Holub, A. and Perona, P. (2005). A discriminative framework for modelling object classes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 664 – 671.
- Ireland, C. and Kullback, S. (1968). Contingency tables with given marginals. *Biometrika*.
- Jebara, T. (2004). *Machine Learning: Discriminative And Generative*. Kluwer Academic Publishers.
- Jiao, F., Wang, S., Lee, C. H., Greiner, R., and Schuurmans, D. (2006). Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of the joint conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (ACL/COLING 2006)*, Sidney, Australia.

- Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 200–209.
- Jordan, M. I. (1999). *Learning in graphical models*. MIT Press, Cambridge, MA.
- Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183.
- Jousse, F., Gilleron, R., Tellier, I., and Tommasi, M. (2006a). Champs conditionnels aléatoires pour l’annotation d’arbres. In *8ème Conférence francophone sur l’Apprentissage automatique (CAp’2006)*, pages 171–186.
- Jousse, F., Gilleron, R., Tellier, I., and Tommasi, M. (2006b). Conditional random fields for xml trees. In *Proceedings of the ECML Workshop on Mining and Learning in Graphs*.
- Kanamori, T. and Shimodaira, H. (2003). Active learning algorithm using the maximum weighted log-likelihood estimator. In *Journal of Statistical Planning and Inference*, volume 116, pages 149 – 162.
- Kazama, J. and Tsujii, J. (2003). Evaluation and extension of maximum entropy models with inequality constraints. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 137–144, Morristown, NJ, USA.
- Kim, Y. (2001). Application of maximum entropy markov models on the protein secondary structure prediction. Technical report, Department of Chemistry and Biochemistry, University of California, San Diego.
- Kindermann, R. and Snell, J. L. (1980). *Markov Random Fields and Their Applications*. AMS.
- Klein, D. and Manning, C. D. (2002). Conditional structure versus conditional estimation in nlp models. In *Conference on Empirical Methods in Natural Language Processing*, pages 9 – 16.
- Klein, D. and Manning, C. D. (2004). Corpus-based induction of syntactic structure: models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 478, Morristown, NJ, USA. Association for Computational Linguistics.
- Koo, T., Globerson, A., Carreras, X., and Collins, M. (2007). Structured prediction models via the matrix-tree theorem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 141–150, Prague, Czech Republic. Association for Computational Linguistics.
- Koski, T. (2001). *Hidden Markov models for bioinformatics*. Springer Verlag.
- Krishnapuram, B., Carin, L., Figueiredo, M. A., and Hartemink, A. J. (2005). Sparse multinomial logistic regression: fast algorithms and generalization bounds. *IEEE Transaction on pattern analysis and machine learning*, 27(6).
- Kudo, T. (2005). CRF++: Yet another CRF toolkit.

- Kudo, T., Yamamoto, K., and Matsumoto, Y. (2004). Applying conditional random fields to japanese morphological analysis. In Lin, D. and Wu, D., editors, *Proceedings of EMNLP 2004*, pages 230–237, Barcelona, Spain.
- Kupiec, J. (1992). Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language*, 6:225–242.
- Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 282–289. Morgan Kaufmann, San Francisco, CA.
- Lasserre, J. A., Bishop, C. M., and Minka, T. P. (2006). Principled hybrids of generative and discriminative models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 87–94. IEEE Computer Society.
- Lauritzen, S. S. (1996). *Graphical Models*. Oxford University Press.
- Lee, C.-H., Schmidt, M., Murtha, A., Bistritz, A., Sander, J., and Greiner, R. (2005). Segmenting brain tumors with conditional random fields and support vector machines. In *International Conference on Computer Vision workshop (ICCV CVBIA)*.
- Lee, S.-I., Lee, H., Abbeel, P., and Ng, A. (2006). Efficient l1 regularized logistic regression. In *Proceedings of the Twenty-first National Conference on Artificial Intelligence (AAAI-06)*, pages 1–9, Boston, MA, USA.
- Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In *ECML*.
- Liang, P. and Jordan, M. (2008). An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *Proceedings of the 25th international conference on Machine learning (ICML'08)*, pages 584–591.
- Liu, Y., Carbonell, J., Weigele, P., and Gopalakrishnan, V. (2005). Segmentation conditional random fields (scrfs): A new approach for protein fold recognition. In *Proc. of the 9th Ann. Intl. Conf. on Comput. Biol. (RECOMB)*, pages 14–18. ACM Press.
- Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning*.
- Mann, G. and McCallum, A. (2007a). Efficient computation of entropy gradient for semi-supervised conditional random fields. In *NAACL/HLT*, pages 109 – 112.
- Mann, G. and McCallum, A. (2007b). Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 593 – 600.
- Mann, G. and McCallum, A. (2008). Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of Association of Computational Linguistics*.

- Mao, Y. and Lebanon, G. (2007). Isotonic conditional random fields and local sentiment flow. In *Advances in Neural Information Processing Systems 19*, pages 961–968.
- Mason, J. (2002). SpamAssassin corpus.
- McCallum, A. (2003). Efficiently inducing features of conditional random fields. In *Proceedings of the conference Uncertainty in Artificial Intelligence (UAI)*, Acapulco, Mexico.
- McCallum, A., Freitag, D., and Pereira, F. (2000). Maximum entropy markov models for information extraction and segmentation. In *Proc. 17th International Conf. on Machine Learning (ICML)*.
- McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of CoNLL-2003*, pages 188–191, Edmonton, Canada.
- McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>.
- Meier, L., van de Geer, S., and Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of The Royal Statistical Society Series B*, 70(1):53–71.
- Mérialdo, B. (1993). Tagging english text with a probabilistic model. *Computational linguistics, volume 20, Issue 2*.
- Merz, C., St. Clair, D., and Bond, W. (1992). Semi-supervised adaptive resonance theory (smart2). In *International Joint Conference on Neural Networks*, volume 3, pages 851 – 856.
- Minka, T. (2005). Discriminative models, not discriminative training. Technical Report TR-2005-144, Microsoft Cambridge.
- Ng, A. and Jordan, M. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naïve Bayes. In *NIPS*.
- Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3):103–134.
- Nocedal, J. (1980). Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35:773–782.
- Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer.
- Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Okazaki, N. (2007). CRFsuite: A fast implementation of conditional random fields (CRFs).
- O’Neill, T. (1978). Normal discrimination with unclassified observations. *Journal of the American Statistical Association*, 73(364):821–826.

- Pal, C., Sutton, C., and McCallum, A. (2006). Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2006*, Toulouse, France.
- Peng, F., Feng, F., and McCallum, A. (2004). Chinese segmentation and new word detection using conditional random fields. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.
- Perkins, S., Lacker, K., and Theiler, J. (2003). Grafting: Fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research (JMLR)*, 3:1333–1356.
- Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 2nd edition.
- Qi, Y. A., Szummer, M., and Minka, T. P. (2005a). Bayesian conditional random fields. In *Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS)*.
- Qi, Y. A., Szummer, M., and Minka, T. P. (2005b). Diagram structure recognition by bayesian conditional random fields. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Qian, X., Jiang, X., Zhang, Q., Huang, X., and Wu, L. (2009). Sparse higher order conditional random fields for improved sequence labeling. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 849–856, New York, NY, USA. ACM.
- Quattoni, A., Collins, M., and Darrell, T. (2004). Conditional random fields for object recognition. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS)*.
- Quattoni, A., Wang, S., Morency, L.-P., Collins, M., and Darrell, T. (2007). Hidden conditional random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(10):1848–1852.
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.
- Ramos, F., Fox, D., and Durrant-Whyte, H. (2007). Crf-matching: Conditional random fields for feature-based scan matching. In *Robotics Science and Systems*.
- Rathnaparkhi, A. (1998). *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania.
- Riezler, S. and Vasserman, A. (2004). Incremental feature selection and l1 regularization for relaxed maximum-entropy modeling. In Lin, D. and Wu, D., editors, *Proceedings of EMNLP 2004*, pages 174–181, Barcelona, Spain. Association for Computational Linguistics.
- Rigollet, P. (2007). Generalization error bounds in semi-supervised classification under the cluster assumption. *J. Mach. Learn. Res.*, 8:1369–1392.

- Rigouste, L., Cappé, O., and Yvon, F. (2007). Inference and evaluation of the multinomial mixture model for text clustering. *Information Processing & Management*, 43(5):1260–1280.
- Rosenfeld, R. (1996). A maximum entropy approach to adaptive statistical learning modeling. *Computer, Speech and Language*, 10:187 – 228.
- Rozenknop, A. (2002). *Modèles syntaxiques probabilistes non-génératifs*. PhD thesis, Dpt. d’informatique, École Polytechnique Fédérale de Lausanne.
- Rubinstein, Y. D. and Hastie, T. (1997). Discriminative vs informative learning. In *KDD*, pages 49 – 53.
- Sarawagi, S. and Cohen, W. W. (2004). Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems (NIPS\*18)*, Alberta, CA.
- Schwartz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6:461 – 464.
- Scudder, H. (1965). Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11:363–371.
- Seeger, M. (2002). Learning with labeled and unlabeled data. Technical report, University of Edinburgh, Institute for Adaptive and Neural Computation.
- Sejnowski, T. J. and Rosenberg, C. R. (1987). Parallel networks that learn to pronounce english text. *Complex Systems*, 1.
- Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of Human Language Technology-NAACL 2003*, pages 213–220, Edmonton, Canada.
- Sha, F. and Saul, L. K. (2005). Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *Proceedings of the Twenty Second International Conference on Machine Learning (ICML-05)*, pages 785–792.
- Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244.
- Siddiqi, S. M. and Moore, A. W. (2005). Fast inference and learning in large-state-space hmms. In *Proceedings of the 22nd international conference on Machine learning*, pages 800–807, Bonn, Germany.
- Smith, A. and Osborne, M. (2006). Using gazetteers in discriminative information extraction. In *CoNLL*.
- Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization*. Wiley.
- Stanke, M. and Waack, S. (2003). Gene prediction with hidden Markov model and a new intron submodel. *Bioinformatics*, pages 215–225.
- Sugiyama, M., Krauledat, M., and Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005.

- Sung, Y.-H., Boulis, C., Manning, C., and Jurafsky, D. (2007). Regularization, adaptation, and non-independent feature improve hidden conditional random fields for phone classification. In *IEEE Automatic Speech Recognition and Understanding Workshop*.
- Sutton, C. and McCallum, A. (2005). Piecewise training for undirected models. In *UAI*.
- Sutton, C. and McCallum, A. (2006). An introduction to conditional random fields for relational learning. In Getoor, L. and Taskar, B., editors, *Introduction to Statistical Relational Learning*. The MIT Press, Cambridge, MA.
- Sutton, C. and McCallum, A. (2007). Piecewise pseudolikelihood for efficient training of conditional random fields. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 863–870, Corvallis, Oregon.
- Sutton, C., Rohanimanesh, K., and McCallum, A. (2004). Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of the International Conference on Machine Learning*, Alberta, Canada.
- Suzuki, J., Fujino, A., and Isozaki, H. (2007). Semi-supervised structured output learning based on a hybrid generative and discriminative approach. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.
- Suzuki, J. and Isozaki, H. (2008). Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data. In *Proceedings of ACL-08: HLT*.
- Suzuki, J., Isozaki, H., Carreras, X., and Collins, M. (2009). An empirical study of semi-supervised structured conditional models for dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 551 – 560.
- Szafranski, M., Grandvalet, Y., and Morizet-Mahoudeaux, P. (2007). Hierarchical penalization. In *Advances in Neural Information Processing Systems 20*, pages 1457–1464. MIT press.
- Szummer, M. and Jaakkola, T. (2002). Information regularization with partially labeled data. In *NIPS*.
- Tappen, M. F., Liu, C., Adelson, E. H., and Freeman, W. T. (2007). Learning gaussian conditional random fields for low-level vision. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J.R.Statist.Soc.B*, 58(1):267–288.
- Tjong Kim Sang, E. F. and Buchholz, S. (2000). Introduction to the conll-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132, Lisbon, Portugal.
- Tjong Kim Sang, E. F. and de Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 155–158, Edmonton, Canada.



- Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 173–180, Morristown, NJ, USA.
- Toutanova, K. and Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, pages 63–70, Morristown, NJ, USA.
- Tseng, H., Chang, P., Andrew, G., Jurafsky, D., and Manning, C. (2005). A conditional random field word segmenter. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 168–171.
- Tsuboi, Y., Kashima, H., Mori, S., Oda, H., and Matsumoto, Y. (2008). Training conditional random fields using incomplete annotations. In *Proceedings of 22nd International Conference on Computational Linguistics (COLING)*.
- Tu, Z. (2007). Learning generative models via discriminative approaches. In *IEEE Computer Vision and Pattern Recognition*.
- Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley.
- Vishwanathan, S. V. N., Schraudolph, N. N., Schmidt, M., and Murphy, K. (2006). Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23th International Conference on Machine Learning*, pages 969–976. ACM Press, New York, NY, USA.
- Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.
- Wainwright, M. J. and Jordan, M. I. (2003). Graphical models, exponential families, and variational inference. Technical report, University of California, Berkeley.
- Wallach, H. M. (2002). *Efficient Training of Conditional Random Fields*. PhD thesis, University of Edinburgh, Division of Informatics.
- Watanabe, Y., Asahara, M., and Matsumoto, Y. (2007). Graph-based approach to named entity categorization in wikipedia using conditional random fields. In *In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 649–657.
- White, H. (1982). Maximum likelihood estimation in misspecified models. *Econometrica*, 50(1):1–25.
- Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *ICML*, pages 412–420. Morgan Kaufmann Publishers.
- Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196.

- Yuan, M. and Lin, Y. (2005). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B* 68, 1:49–67.
- Zhang, T., Damerou, F., and Johnson, D. (2001). Text chunking using regularized winnow. In *ACL*, Toulouse, France.
- Zhang, T. and Johnson, D. (2003). A robust risk minimization based named entity recognition system. In *Proceedings of CoNLL-2003*, Edmonton, Canada.
- Zhang, X., Aberdeen, D., and Vishwanathan, S. V. N. (2007). Conditional random fields for multi-agent reinforcement learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 1143–1150, New York, NY, USA. ACM.
- Zhao, P., Rocha, G., and Yu, B. (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *to appear, Annals of Statistics*.
- Zhu, X. (2005a). Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.
- Zhu, X. (2005b). *Semi-Supervised Learning with Graphs*. PhD thesis, Carnegie Mellon University.
- Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University.
- Zhu, X., Kandola, J., Ghahramani, Z., and Lafferty, J. (2005). Nonparametric transforms of graph kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS) 17*.
- Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. Royal. Stat. Soc. B.*, 67(2):301–320.

