DOMAIN KNOWLEDGE, UNCERTAINTY, AND PARAMETER CONSTRAINTS

A Thesis
Presented to
The Academic Faculty

by

Yi Mao

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computational Science & Engineering

Georgia Institute of Technology
December 2010

DOMAIN KNOWLEDGE, UNCERTAINTY, AND PARAMETER CONSTRAINTS

Approved by:

Professor Guy Lebanon, Advisor
School of Computational Science & Engineering
Georgia Institute of Technology

Professor Alexander Gray
School of Computational Science & Engineering
Georgia Institute of Technology

Professor Chin-Hui LEE
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Alexander Shapiro
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Hongyuan Zha
School of Computational Science & Engineering
Georgia Institute of Technology

Date Approved: 23 August 2010

ACKNOWLEDGEMENTS

First and foremost I would like to thank my advisor Guy Lebanon. I was very fortunate to have Guy as my advisor. Guy introduced me into the field of machine learning, and helped me in every aspect of my research. Always being approachable and patient, he is open enough to let me explore ideas freely while provides the strongest support I can ever imagine as an advisor. Talking to Guy is always a pleasure and I benefit a lot from his ability to think out of box.

I would like to thank all members of my thesis committee: Alexander Gray, Chin-Hui Lee, Alexander Shapiro and Hongyuan Zha. They have provided valuable feedback on my thesis and helped guide it to completion. Gray's FASTlab group meeting has been a valuable source of machine learning topics. Lee is encouraging and introduced me the work of Bayesian adaptive learning for acoustic modeling. I also benefit from Shapiro who pointed out to me the literature on probabilistic optimization and from Zha for discussions on potential information retrieval applications. Chris Clifton, William Cleveland and Robert Givan were on my thesis committee while I was at Purdue. I appreciate their valuable comments and suggestions on the initial draft of my thesis.

While this thesis is largely based on close interactions with Guy, discussions with other people have helped me to view the problem from different perspectives. Among them, I would especially like to thank Joshua Dillon, Yang Zhao, Paul Kidwell, Krishnakumar Balasubramanian, Mingxuan Sun, Seungyeon Kim, Jiang Bian, Shuanghong Yang, Da Kuang, Jingu Kim and all members from the Fastlab.

I would like to thank John Platt, Denny Zhou and Kevyn Collins-Thompson from Microsoft research and Rómer Rosales, Harald Steck and Le Lu from Siemens.

John is the most knowledgeable person I have ever met. He is always a source of help whenever I get stuck. Rómer and Harald had been interested in my work and provided me with the opportunity to explore their commercial datasets. All had been supportive when I was interning there and helped me with my job search.

I would also like to thank many other people from ECE, CS and statistics at Purdue, and CSE at Georgia tech. This is such a long list and I won't enumerate your names here. But thank you for making West Lafayette and Atlanta much more fun places than they are supposed to be.

Finally, I would like to dedicate my thesis to my parents, for their tremendous support during my Ph.D. study and for tolerance of my occasional impatience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

CRFs	Conditional Random Fields.
EB	empirical Bayes.
l	number of constraints.
m	number of training examples.
MAP	maximum posterior.
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with mean μ and covariance Σ.
MLE	maximum likelihood estimation.
n	number of features.
PCA	Principle Component Analysis.
RCV1	Reuters Corpus Volume I.
t-SNE	t-distributed stochastic neighbor embedding.
Θ	model space.
tr	trace.

SUMMARY

This thesis identifies three major issues in incorporating domain knowledge into supervised learning and discusses tentative attempts to solve them.

Domain knowledge is usually provided in a certain form that is not necessarily compatible with the supervised learning algorithm. In statistical modeling, it is a common practice to assume that data are generated iid from some distribution p_θ parameterized by $\theta \in \mathbb{R}^n$. Learning algorithms then try to identify the true parameter θ^{true} given a set of examples. It is therefore natural to expect that domain knowledge should directly relate to the parameters θ for it to be useful. However, users tend to specify their knowledge in the form of probabilities of certain events rather than parameter value constraints, and it is not so obvious how to convert between them, especially for conditional models such as conditional random fields (CRFs). To overcome the difficulty, Chapter 3 developed a systematic way to obtain domain-dependent priors through probability elicitation and to incorporate them through parameter space regularization for conditional models. This leads to isotonic CRFs which are variants of CRFs with isotonic constraints over the parameter space. Chapter 3 applied isotonic CRFs to sentiment prediction and information extraction, and demonstrated their promising usage in modelling local sentiment flow, analyzing author's writing style and summarizing document content.

Domain knowledge provided by humans often holds with some degree of uncertainty. The uncertainty may arise when the knowledge set of an expert changes. This has already been demonstrated in a recent study showing that the ratings in the Netflix dataset are strongly affected by the time the ratings were provided by users. The uncertainty can be even more pronounced when the domain knowledge is obtained

implicitly by interpreting user feedback such as clicks. It is not hard to imagine that a click on a search result does not necessarily mean the result is relevant to the query. The click can be random. In view of this, Chapter 4 proposes to explicitly model domain knowledge uncertainty by specifying the probability the knowledge is expected to hold, and aggregate both domain knowledge and its uncertainty into the learning process within a hierarchical Bayes framework. In contrast to hard parameter constraints, the approach is effective even when the domain knowledge is inaccurate and generally results in superior modelling accuracy. It therefore enables us to incorporate other non-traditional types of knowledge, such as information from trained classifiers whose usage has been severely limited due to its accuracy.

Standard approaches of incorporating domain knowledge admit information only at the initial stage and no user feedback is allowed afterwards. This contrasts with the belief that users may provide better knowledge if they are informed of intermediate learning results. For example in web search, one may build a webpage ranking model based on users' click feedback. When the model is in operation, new click feedback will be collected and the model should be refined accordingly. Therefore it is essential to provide users with a visual summary of the available information, and allow them to provide valuable feedback in real-time. This requires both an efficient learning procedure and the ability to support effective user interactions. Chapter 5 addressed this problem in the context of metric learning for text documents where users specify word similarity information on the fly. The problem is approached via learning techniques such as online update and Bregman projection. The effort leads to an improved metric for documents, and fosters better visual understanding of text corpus.

CHAPTER I

INTRODUCTION

During the years of 1952–1962, Arthur Samuel wrote the first program for computer checkers. Samuel's program tried to teach computers to play checkers, and is considered the world's first self-learning program. Since then more than half a century has passed, and the field of machine learning has experienced substantial growth in terms of theories, algorithms and applications. In fact, machine learning techniques are considered the-state-of-the-art in quite a few areas such as information retrieval, natural language processing, computer vision and speech processing.

The early history of machine learning draws inspirations from psychology and biology, where people had focused on developing complicated models that mimic the human learning behavior. An example is the neural network, where complicated tasks are accomplished by interactions between simple building blocks such as perceptrons. With the advance in data acquisition techniques and the popularity of sharing the data through internet, machine learning now is more concerned with analyzing the data, by seeking a (possibly simple) model that both explains the data and generalizes well.

In this thesis, we focus on supervised learning, though there exist many other problem settings such as unsupervised learning or semi-supervised learning. Briefly speaking, given a set of m training examples $\mathcal{D} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ where $x^{(i)} \in \mathcal{X} \subseteq \mathbb{R}^n$ is some observation with corresponding label $y^{(i)} \in \mathcal{Y} \subseteq \mathbb{R}$, the goal of supervised learning is to seek some function $f : \mathcal{X} \to \mathcal{Y}$ from a function class $\mathcal{F}$ such that f not only captures the relationship between $\mathcal{X}$ and $\mathcal{Y}$ that has been encoded by the dataset $\mathcal{D}$, but also generalizes well on unseen data. The function f is usually chosen by means of empirical risk minimization or structural risk minimization.

For supervised learning, we need training data to be fully labeled. While it may be relatively easy to obtain a collection of unlabeled data, there can be a considerable amount of difficulty in getting the labels due to either a limit on the budget in hiring labeling person or difficulty in getting competent labeling person. For example, Reuters hired around a dozen people working full time to handle the coding of RCV1 dataset [64], which consists over 800,000 news stories produced during the years 1996 and 1997 by Reuters. On the other side, how many news stories are posted every year on http://news.google.com? The answer is obvious, way more than 800,000. In fact, there is a branch of machine learning called active learning that discusses means of selecting examples to label under various resource constraints such as cost [110].

This thesis considers the case of limited labeled data for supervised learning, and presents a general approach to select a model $f \in \mathcal{F}$ that explains the dataset $\mathcal{D}$. Our high-level description of the approach is to use domain knowledge. To be more concrete, let us first examine the following two questions.

Question 1 : What kind of domain knowledge can we leverage?

> Under the restrictive setting of supervised learning, domain knowledge can be roughly categorized according to the target it describes. Domain knowledge about observations in $\mathcal{X}$ is often referred to as feature engineering, i.e., selecting a set of features that best describe the objects. For example, color, texture and SIFT (scale-invariant feature transform) features are considered effective in describing an image object, and TF–IDF (term frequency–inverse document frequecy) representation has been widely used to represent a document.
>
> Domain knowledge about labels in $\mathcal{Y}$ in general is concerned with the co-occurrence relationship among a subset of labels. An example is image annotation where an image is assigned multiple labels each describing one aspect

of the image. If we know a priori that a region of the image has been tagged with the label `computer`, it is unlikely that another region from the same image will be tagged with the label `park`.

Finally, we have domain knowledge that is expressed in terms of the relationship between $\mathcal{X}$ and $\mathcal{Y}$. Domain knowledge falling into this category exhibits more degrees of freedom. For example, given two observations, one may ask whether they belong to the same class or not. This type of knowledge has been used extensively in metric learning and constrained clustering. On the other hand, users may wish to express for some observation x how likely it will be labeled as y. This has been commonly practiced in natural language processing applications. An example is the name entity recognition where capitalized words are likely to be the names of persons or organizations.

Question 2 : To what extent can we rely on domain knowledge?

Domain knowledge is uncertain in nature. There are many ways that uncertainty is introduced. The data collection procedure is not error free. An example is to build a digital library by first scanning books and then applying OCR (optical character recognition) techniques to translate them into text. Errors may be introduced during the translation.

The labels can be uncertain. For example, it is difficult to justify why a movie should be rated 8 instead of 7 (1–10 scales). Even if the labeling task is unambiguous, labels may be obtained from sources that are either incompetent or with little dedication. This is usually the case with outsourcing, see e.g. the Amazon Mechanical Turk (https://www.mturk.com/). This phenomenon is even more pronounced when the labels are obtained by implicit user feedback, such as the clickthrough data collected from user browsing behavior.

The relationship between observations and labels can also be uncertain. For example, in sentiment prediction, the presence of the word `good` corresponds usually, but not always to a positive sentiment.

Uncertainty may be reduced by choosing appropriate questions to ask. For the task of rating movies, though it is different to determine the score to be assigned to a movie, it is relatively easy to identify from two movies which one should receive a higher score. Similar strategy has been practiced in [56] where the clicked URLs are assumed to be more relevant to the query than the unclicked URLs. However, the uncertainty can not be eliminated. For example, the click can be random in the above example.

In this thesis we consider function class $\mathcal{F}$ is parameterized by θ and domain knowledge about the relationship between $\mathcal{X}$ and $\mathcal{Y}$ is expressed as constraints over parameter space. Such type of knowledge is inspired from the following two lines of work:

- Parameter constraints have shown to be extremely helpful for generative models where parameters directly relate to the probabilities of generating the data. For example, consider using the unigram language model to model documents annotated as `computer`. The parameter associated with a word is simply the probability that this word appears in a document from `computer` category. It is therefore natural to expect that parameters associated with word `c++` or `compiler` are large even if such words may not appear frequently in the training dataset. Parameter constraints have also been widely employed in Bayesian networks. One type of constraint is parameter sharing which constrains parameters to have the same value [76]. More general types of parameter constraints such as inequality constraints or constraints on sums of parameter values have been considered in [77].

- Isotonic regression [2, 99] is an important method in constrained statistical inference. It can be traced back to the problem of maximizing the likelihood of univariate normal distributions subject to an ordered restriction on the means. The term *isotonic* is interpreted as order-preserving: for a finite set $S = \{1, \cdots, n\}$ on which a full order $\leq$ is defined, a real vector $(\beta_1, \cdots, \beta_n)$ is isotonic if $i, j \in S$, $i \leq j$ imply $\beta_i \leq \beta_j$. Given real vector $(x_1, \cdots, x_n)$ with weights $(w_1, \cdots, w_n)$, the isotonic regression takes the form of a weighted least square fitting which minimizes $\sum_{i=1}^{n} w_i(x_i - \beta_i)^2$ subject to the constraint that $(\beta_1, \ldots, \beta_n)$ is isotonic. Various extensions have been proposed for isotonic regression. Some of them consider relationships other than a full order. Examples include the tree order $\beta_1 \leq \beta_2, \ldots, \beta_1 \leq \beta_n$, and the umbrella order $\beta_1 \leq \ldots \leq \beta_i \geq \ldots \geq \beta_n$ for some fixed i. Most similar to our work is the ordering constraint proposed in [52] for normal means from a two-way layout experiment

  $$\beta_{i+1,j+1} - \beta_{i+1,j} - \beta_{i,j+1} + \beta_{i,j} \geq 0 \quad i = 1, \ldots, m-1,\ j = 1, \ldots, n-1$$

  which states that the differences $\beta_{i'j} - \beta_{ij}$ grow as the level j increases for any $i' > i$.

  The same motivation is shared by fused lasso [109] where features are ordered in some meaningful way. The fused lasso penalizes the L_1-norm of both the coefficients and their successive differences. It encourages sparsity in the coefficients, and also sparsity in their differences, i.e. the coefficients are locally constant.

Chapter 3 considers in the case of conditional models, how parameter constraints are related to domain knowledge expressed as probability constraints. We develop isotonic CRFs which are variants of CRFs with isotonic constraints over the parameter space, and provide a way to obtain domain-dependent constraints through their probability counterparts. We apply the model to sentiment prediction and information extraction, and demonstrate its promising usage in modeling local sentiment flow, analyzing author's writing style and summarizing document content.

Domain knowledge provided by humans often holds with some degree of uncertainty. In view of this, Chapter 4 proposes to explicitly model domain knowledge uncertainty by specifying the probability the knowledge is expected to hold, and aggregate both domain knowledge and its uncertainty into the learning process within a hierarchical Bayes framework. In contrast to hard parameter constraints, the approach is effective even when the domain knowledge is inaccurate and generally results in superior modelling accuracy. It therefore enables us to incorporate other non-traditional types of knowledge, such as information from trained classifier whose usage has been severely limited due to its accuracy.

Standard approaches of incorporating domain knowledge admit information only at the initial stage. This contrasts with the belief that users may provide better knowledge if they are informed of intermediate learning results. Therefore it is essential to provide users with a visual summary of the available information, and allow them to provide valuable feedback in real-time. Chapter 5 addresses this problem in the context of metric learning and text visualization, and demonstrates how to modify the geometry of model space by explicit domain knowledge from experts and general linguistic resources.

Chapter 3 is based on the work published in NIPS 2006 [69] and Machine Learning 2009 [71]. Chapter 4 draws significantly from the work published in UAI 2009 [70]. The work of Chapter 5 were published in COLING 2010 [67], coauthored with Krishnakumar Balasubramanian and Guy Lebanon. Krishnakumar Balasubramanian created the word hierarchy for the 20 newsgroup dataset (Figure 14) and did experiments described in Section 5.2.5. Some results in this thesis were also presented in ICML workshop on Learning in Structured Output Spaces 2006.

CHAPTER II

RELATED WORK

In this chapter, different types of domain knowledge are summarized and their usage in modifying supervised learning models is discussed. We also provide some background information about several natural language processing tasks on which the experiments are conducted.

2.1 Domain Knowledge for Supervised Learning

Incorporating domain knowledge into the learning process can substantially improve modeling accuracy, especially when the training data is scarce. In some cases the knowledge may be incorporated by modifying the underlying statistical model. In other cases standard off-the-shelf models are used such as logistic regression, SVM, mixture of Gaussians, etc., and the domain knowledge is integrated into the training process of these models by constraining the parameters to a certain region.

2.1.1 Knowledge about Observations

Domain knowledge about observations in $\mathcal{X}$ is often referred to as feature engineering, i.e., selecting a set of features that best describe the objects. For example, color, texture and SIFT features are considered effective in describing an image object, and TF–IDF representation has been widely used to represent document.

Since this type of knowledge varies from application to application, it is not the focus of this thesis. However, it is worth noticing that this type of knowledge is of crucial importance for the success of solving an engineering problem. In fact, domain knowledge is equivalent to feature engineering in some circumstances.

2.1.2 Knowledge about Labels

Domain knowledge about labels in general is concerned with the co-occurrence relationship among a subset of labels. It arises naturally in the problems of multiple label classification or structural prediction. An example is object detection which assigns a label to each object within an image. If we know a priori an object is very likely to be `computer`, it is then less likely that another object from the same image will be labeled as `park`. This observation has motivated a couple of algorithms in computer vision which incorporate contextual information to augment object detection, including the work of [39] and [111].

In natural language processing, Roth and Yih [91] consider non-local and non-sequential constraints over the output sequence and propose a novel inference procedure based on integer linear programming (ILP) which extends the CRFs to naturally support such constraints. They demonstrated their algorithm for the problem of semantic role labeling which, given an input sentence, attempts to identify semantic arguments for each verb in the sentence, and assign role for each argument. The constraints that have been considered include "no duplicated argument labels", i.e. a verb cannot have two arguments of the same type, and "at least one argument" which means that each verb must have at least one core argument.

Recently developed Markov logic networks (MLNs) [88] naturally combine first-order logic and probabilistic graphical models. MLNs allow easy integration of domain knowledge expressed in terms of logic clauses. For example, when applying MLNs to segment bibliographic citations, Poon and Domingos [83] consider the following mutual exclusivity constraint which states that a token can only be part of at most one field. It would be straightforward to add more constraints to MLNs. One of them might be that a `Venue` token cannot appear right after an `Author` token.

2.1.3 Knowledge about Models

Domain knowledge about supervised learning models can be roughly categorized as either explicit knowledge of the model parameters, or implicit knowledge via the associations between observations and their corresponding labels.

2.1.3.1 Knowledge from Domain Expert and its Elicitation

In a statistical framework, the expert's knowledge has to be in probabilistic form for it to be used. However, unless the expert is a statistician, or is very familiar with statistical concepts, efforts have to be made to formulate the expert's knowledge and beliefs in probabilistic terms. This is done through elicitation [41, 78] in the statistical literature. Psychological literature suggests that people are prone to certain heuristics and biases in the way they respond to situations involving uncertainty. As a result, elicitation is conducted in a principled way where stages involving eliciting summaries, fitting a distribution and testing adequacy may repeat several times before a faithful elicitation is reached. The usefulness of elicitation has been demonstrated in statistical literature where most work concentrates on eliciting univariate probability distributions. Multivariate elicitation is largely unexplored due to the complexity of formulating variable interactions.

2.1.3.2 Prior and Regularization

Perhaps the most widely used Bayesian approach is to impose a univariate Gaussian prior with mean zero and variance σ_i^2 on each parameter θ_i

$$p(\theta_i) \sim \mathcal{N}(0, \sigma_i^2) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{\theta_i^2}{2\sigma_i^2}\right). \quad (1)$$

By specifying a mean of zero for each Gaussian, we encode our prior belief that the parameter θ_i will be near zero. The larger the value of σ_i^2, the less the confidence in the prior belief. In the simplest case we let σ_i^2 equal the same σ^2 for all i. Assuming θ_i are independent of each other, the overall prior for θ is then the product of the priors for each θ_i, and the maximum a posterior (MAP) estimate of θ is

$$\hat{\theta}_{\text{MAP}} = \arg\min_{\theta} \ -\ell(\theta|\mathcal{D}) + c\|\theta\|_2^2 \tag{2}$$

where $\ell(\theta|\mathcal{D})$ is data likelihood and $c = \frac{1}{2\sigma^2}$. The term $\|\theta\|_2^2$ is called the ridge penalty, and c controls the tradeoff between loss and penalty, with the loss function being the log loss in (2). As stated above, the ridge penalty shrinks the parameters θ towards zero. This shrinkage has the effect of reducing the variances of θ, hence possibly improves the model prediction accuracy, especially when there are many highly correlated features [46].

Double-exponential prior is another popular prior distribution given by

$$p(\theta_i) \sim \frac{\alpha_i}{2}\exp(-\alpha_i|\theta_i|). \tag{3}$$

Again, we assume the independence among θ_i. If we let α_i equal the same α for all i, the MAP estimate of θ is

$$\hat{\theta}_{\text{MAP}} = \arg\min_{\theta} \ -\ell(\theta|\mathcal{D}) + \alpha\|\theta\|_1 \tag{4}$$

where $\|\theta\|_1$ is called the lasso penalty. Similar to the ridge penalty, the lasso penalty also shrinks the parameters θ towards zero. In addition, the lasso penalty does a kind of continuous feature selection, causing some parameters θ_i to be zero when α is sufficiently large. This property, which leads to a sparse solution in a high dimensional space, comes from the L_1 nature of the lasso penalty and does not hold for the ridge penalty.

Figure 1 shows that the double-exponential density has heavier tails than the Gaussian density. This indicates that the lasso penalty is more likely to produce some large fitted parameters and leaves others at zero, especially in high dimensions.

2.1.3.3 Knowledge of Label-Feature Association

In some cases experts' knowledge is formulated as the association between label y and feature f_i (or instance $x^{(i)}$). As an example, in the part-of-speech tagging, we may

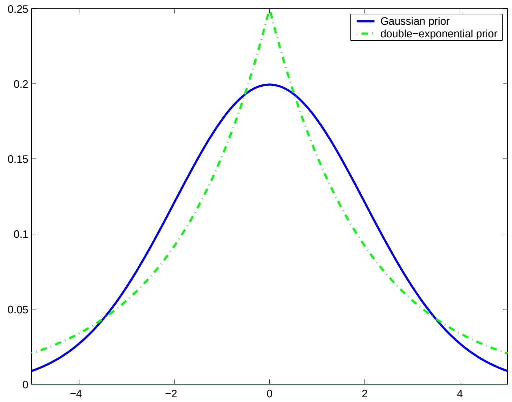


Figure 1: Plots of the Gaussian density with $\sigma = 2$, and the double-exponential density with $\alpha = .5$.

estimate that with 90% probability a word is a noun if it ends with "-ion". Similarly, in the named-entity recognition, we may estimate that 50% of capitalized words are named entities. This type of knowledge is usually in the form of a probability distribution on labels y conditioned on either some feature f_i or instance $x^{(i)}$. Relative entropy [22] is often used to measure the fit between the model and the prior. Because of the nature of such knowledge, it can be easily exploited to work with a semi-supervised learning algorithm.

Schapire et al. [93] consider the prior knowledge of mapping each training instance $x^{(i)}$ to an estimated conditional probability distribution $\pi(\cdot|x^{(i)})$ over possible label values. Their approach is based on the boosting-style algorithm for logistic regression. For each training example $x^{(i)}$, relative entropy is used to measure the fit between the prior $\pi(\cdot|x^{(i)})$ and the model $p_\theta(\cdot|x^{(i)})$. Prior knowledge is incorporated via adding a penalty term during the model fitting. Wu and Srihari [113] explore the same prior for semi-supervised learning using support vector machines (SVMs). They modify SVMs to allow weighted input samples, and formulate the problem as maximizing weighted margin. Prior knowledge $\pi(\cdot|x)$ is specified for a subset of testing examples, and $\pi(y|x)$ is considered as the weight for sample (x, y).

Jin and Liu [55] use label distribution over the testing set as prior to handle the case where the class distribution of the training data is not representative of the true class distribution. They provide an iterative algorithm: in the first step of each iteration, the conditional model is trained using both training and testing data, with class distribution estimated for each test example being fixed; in the second step, the model is fixed and the per-instance class distribution is re-estimated to minimize the divergence from model prediction subject to the constraint that the overall class distribution matches the prior.

Mann and McCallum [66] introduce expectation regularization for semi-supervised learning. The idea is similar to that of [93] with the prior knowledge being replaced by $\pi(\cdot|f_i)$, the conditional probability distribution of labels given feature f_i. A special case of expectation regularization called label regularization is examined in [66] where the feature is activated for every instance. More realistic feature-class associations are discussed in [74].

Feature-class association has also been considered for structured output models. Chang et al. [15] specify the prior knowledge as a set of constraints to be satisfied by the input/output sequences. They use an expectation maximization (EM) like algorithm to incorporate the prior knowledge into semi-supervised learning. In the E-step, the inference procedure produces N best outputs for an unannotated sequence according to a score function which considers both data likelihood and penalties for violating constraints. In the M-step, the N best assignments are used to re-estimate the model parameters. Haghighi and Klein [45] define prototype to be some canonical examples (e.g. words) of each target annotation label (e.g. part-of-speech). Their idea is based on distributional similarity, and is implemented through the use of prototypes as additional features in learning a log-linear model.

A generative model type of prior is introduced in [37] to work with a discriminative classifier such as support vector machines. The prior is a parametric family of distributions $p(x|y)$. A bilevel program is formulated such that the bottom layer, given $p(x|y)$, selects the Bayesian optimal decision rule; and the top layer learns $p(x|y)$ which has a high probability of occuring and at the same time forces the bottom layer to select the decision rule that minimizes the discriminative error on the training set.

Prior knowledge is also useful when learning a generative model. Niculescu et al [77] provide a comprehensive summary of various types of knowledge to be used in a Bayesian network. The knowledge can be conveniently transformed into parameter constraints to be satisfied during model fitting. Graça et al. [43] describe an approach to add a-priori information about latent variables in graphical models without making the models overly complex or intractable. They modify the EM algorithm where, in the E-step, the algorithm finds a distribution that minimizes the KullbackLeibler (KL) divergence from model prediction subject to a set of constraints from prior knowledge. The constraints are specified by bounding expectations of given functions describing instance-label associations. Note, taking expectation with respect to a conditional probability distribution results in per-instance constraints on the output variables.

For problems that no probabilistic models are involved, knowledge is usually expressed implicitly via the relationship between observations and their corresponding labels. For example, in distance metric learning, we may consider whether a pair of observations belong to the same class (must-link) or not (must-not-link) [114, 96]. Alternatively, we may restrict a pair from the same class to have a small distance value while assign a large distance value to a pair from different classes [30]. Instead of dealing with absolute distance values, we may use triplets and relative comparisons such as i and j are more similar to each other than i and k [56, 94, 17].

2.1.3.4 Learning an Informative Prior

Instead of specifying the functional form of the prior directly, one may cast the problem of finding an informative prior as a learning task. The prior may be estimated from an auxiliary problem in a separate procedure, or is an integrated part of the whole learning problem and subject to a joint estimation with model parameters. Learning an informative prior may have the following benefits: (1) it introduces a systematic approach to derive a prior where any advance in machine learning may lead to a better solution for it; (2) the auxiliary problems are often related problems (e.g. the same model on different datasets, or different models on the same dataset) and modeling the similarities among related tasks is reasonable and often effective, resulting in an improved modeling accuracy; (3) the procedure of learning a prior is often informative, and helps better understand the problem.

Learning an informative prior through auxiliary problems usually appears in the literature of transfer learning. In the most general case, knowledge at any level of abstraction, including data representation, distance metric, and model parameters, may be transferred from auxiliary tasks to the primary task. Closely related to our problem is the transfer of model parameters, where parameters learned from auxiliary tasks serve as a prior in learning the target model. The target model can be thought of as a posterior obtained by updating the prior with examples from the target task. Marx et al. [72] compute the mean and variance of a Gaussian prior for Bayesian logistic regression from the parameters of the same model learned from other datasets. Raina et al. [86] present an algorithm for automatically constructing a multivariate Gaussian prior with a full covariance matrix for a given supervised learning task. The algorithm first estimates the covariance for pairs of individual parameters empirically, and then uses a semi-definite program to combine these estimates and learn a good prior for target task. Zhang [116] combines Rocchio algorithm with logistic regression via a Gaussian prior to yield a low-variance model for adaptive filtering. Fei-fei et al. [38] implement the Bayesian prior in more sophisticated models for learning visual models of object categories.

Unlike in transfer learning where the knowledge transfer is unidirectional (from auxiliary tasks to target task), in multi-task learning the knowledge transfer is mutual and between any related tasks. A natural choice for representing the relatedness among tasks is a hierarchical Bayes model, where hyper-parameters are shared among multiple tasks. The use of hierarchical Bayes models for multi-task learning is first discussed in [3] and analysis is given from a Bayesian/information theoretical point of view. Heskes [51] presents a model for multi-task learning by assuming that response variable of each task follows a normal distribution. The mean of the normal distribution is learned using a two-layer neural network, and the variance is composed of a task specific term and a task independent term. Bakker and Heskes [1] implement a hierarchical Bayes model using neural network with the input-to-hidden weights shared among all tasks. Task clustering is introduced for differentiating similarities among tasks and is implemented through designing prior distributions capable of discriminating between tasks. Teh et al. [107] propose a semi-parametric model for multi-task learning. They make use of a set of Gaussian processes that are linearly mixed to capture existing dependencies among tasks. Yu et al. [115] assume that multiple tasks are drawn independently from the same Gaussian process (GP) prior, and learn a model via an EM-based algorithm. The work of Lawrence and Platt [61] makes the same assumption, but fits a model with the informative vector machine. In [63], the prior is learned as part of a single coherent objective, which encompasses both data likelihood and prior, and is jointly optimized for both parameters and hyperparameters.

2.2 Uncertainty in Domain Knowledge

Prior work on incorporating uncertainties into the learning process is naturally divided to data uncertainty, label uncertainty and parameter uncertainty.

2.2.1 Data Uncertainty

Data uncertainty has been addressed by robust linear programming [8], robust linear discrimination [60], total support vector machine [5] and second order cone programming for SVM [98]. It has also been applied to query expansion [21]. They differ slightly in their assumptions about how data is distributed. For example, [8, 60, 98, 21] assume data is distributed as a Gaussian with given mean and covariance matrix, while [60, 98] also assume that data may come from some arbitrary distribution with fixed first and second order moments. The assumption made in [5] is that true data is uniformly distributed over a disk centered at the observed value.

2.2.2 Label Uncertainty

There has been recent interest to use outsourcing websites such as Amazon Mechanical Turk as a cheap and fast way to collect annotations from non-experts over the web [102, 14]. For example, high agreement between non-expert annotations and existing gold standard labels is reported in [102] on five natural language processing tasks.

The annotations from non-experts can be noisy, due to expertise, competence and dedication of the annotators. Sheng et al. [97] show that when labeling is not perfect, selective acquisition of multiple labels is a strategy to perform. Different variants of the strategy may be considered. Among them the simplest approach is to label each example multiple times, and measure the inter-annotator agreement. In fact, repeated labeling is commonly practiced in learning with uncertain labels [65, 100, 101, 87].

A more realistic setting has been considered in [32, 31] where the crowd is large and each annotator only labels a couple of examples. For example, we may think of each user of a search engine as an annotator, and each click as a label. While there can be a huge amount of annotators, most of the links either are not clicked or are clicked just once. The click-patterns of most users are informative, while the click-patterns of others contain noise. Algorithms that can handle noisy labels but without resorting to repeated labeling are of crucial importance in this case.

2.2.3 Parameter Uncertainty

In the machine learning community, parameter uncertainty has been addressed by a variety of techniques, many of which are algorithmic in nature. Related research on incorporating parameter uncertainty are [34, 24] which consider a linear classifier with a Gaussian prior over the model parameter, and update the hyperparameters online using probabilistic parameter constraints. Extension to multi-class classification has been considered in [25], while extension to learning across multiple domains is addressed in [35]. The uncertainty principle has also been applied in [26] and is viewed as a probabilistic version of the geometric large-margin principle there.

2.3 Natural Language Processing Applications

In this section, we briefly review the following natural language processing problems: sentiment prediction, readability prediction and information extraction. Experiments concerning those tasks will be presented in later chapters.

2.3.1 Sentiment Prediction

The World Wide Web and other textual databases provide a convenient platform for exchanging opinions. Many documents, such as reviews and blogs, are written with the purpose of conveying a particular opinion or sentiment. Other documents may not be written with the purpose of conveying an opinion, but nevertheless they contain one. Opinions, or sentiments, may be considered in several ways, the simplest of which is varying from positive opinion, through neutral, to negative opinion.

Most of the research in information retrieval has focused on predicting the topic of a document, or its relevance with respect to a query. Predicting the document's sentiment would allow matching the sentiment, as well as the topic, with the user's interests. It would also assist in document summarization and visualization. Sentiment prediction was first formulated as a binary classification problem to answer questions such as: “What is the review’s polarity, positive or negative?” Pang et al. [82] demonstrated the difficulties in sentiment prediction using solely the empirical rules (a subset of adjectives), which motivates the use of statistical learning techniques. The task was then refined to allow multiple sentiment levels, facilitating the use of standard text categorization techniques [81].

Various statistical learning techniques have been suggested for sentiment prediction, treating the data either as categorial (naive Bayes, maximum entropy and support vector machine [82, 81]) or as ordinal (support vector regression and metric labeling [81]). Although most methods report over 90% accuracy on text categorization, their performance degrades drastically when applied to sentiment prediction.

Indeed, sentiment prediction is a much harder task than topic classification tasks such as Reuters or WebKB. It is different from traditional text categorization: (1) in contrast to the categorical nature of topics, sentiments are ordinal variables; (2) several contradicting opinions might co-exist, which interact with each other to produce the global document sentiment; (3) context plays a vital role in determining the sentiment. In view of this, Mao and Lebanon [69] suggest to model local sentiment flow in documents rather than predicting the sentiment of the entire document directly. The idea is further exploited in [75] where the sentiments of text at varying levels of granularity are jointly classified.

2.3.2 Readability Prediction

We focus on corpus-based statistical models for readability prediction. One example is the popular Lexile measure [105] which uses word frequency statistics from a large English corpus. Collins-Thompson and Callan [20] introduced a new approach based on statistical language modeling, treating a document as a mixture of language models for individual grades. Recent advances in methods for readability prediction include using machine learning techniques such as support vector machines [95], log-linear models [50], k-NN classifiers and combining semantic and grammatical features [49].

2.3.3 Information Extraction

Information extraction involves identifying instances of a particular class of events or relationships in a natural language text, extracting the relevant arguments of the event or relationship, and creating structured representation of extracted arguments. It has received wide attention over the last decade through the series of Message Understanding Conferences[1]. One example is the name entity recognition (NER) that locates and classifies words and phrases in text into predefined categories such as names of persons, organizations, locations, etc. Another example is a relationship extraction task, such as identifying the source of opinion [18].

Much research has been done to improve the extraction performance. Hidden Markov models (HMM) [85], maximum entropy Markov models (MEMM) [73] and conditional random fields [59] are three most popular tagging models up to date. HMM model the joint probability of the observation sequence and the label sequence. It is a generative model that makes a strong independence assumption about observations to ensure inference tractability. This assumption is often inappropriate for real applications, where we believe that the representation should consist of many overlapping features. MEMM remove the assumption by modeling the conditional probability of the next state given the current state and the current observation. Since they use per-state exponential models, MEMM potentially suffer from the label bias problem. CRFs combine the advantages of two previous models by introducing a single exponential model for the joint probability of the entire label sequence given

[1] http://en.wikipedia.org/wiki/Message_Understanding_Conference

the observation sequence, and report superior experimental results. The state-of-the-art results of information extraction are reported on conditional random fields [59]. The generalized perceptron proposed by Collins [19] is another widely used model which is closely related to the CRFs.

CHAPTER III

GENERALIZED ISOTONIC CONDITIONAL RANDOM FIELDS

The most common technique of estimating a distribution $p_\theta(x)$, $x \in \mathcal{X}, \theta \in \Theta$ based on iid samples $x^{(1)}, \ldots, x^{(n)} \sim p_{\theta_0}$ is to maximize the loglikelihood function $\ell(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x^{(i)})$ i.e.,

$$\hat{\theta}^{\text{mle}}(x^{(1)}, \ldots, x^{(n)}) = \arg\max_{\theta \in \Theta} \ell(\theta). \tag{5}$$

The maximum likelihood estimator (MLE) $\hat{\theta}^{\text{mle}}$ enjoys many nice theoretical properties. In particular it is strongly consistent i.e. it converges to the true distribution $\hat{\theta}^{\text{mle}}(x^{(1)}, \ldots, x^{(n)}) \to \theta_0$ with probability 1 as $n \to \infty$. It is also asymptotically efficient which indicates that its asymptotic variance is the inverse Fisher information - the lowest possible variance according to the Cramer-Rao lower bound. These theoretical motivations, together with ample experimental evidence have solidified the role of the maximum likelihood estimate as the method of choice in many situations.

In some cases, additional information concerning the domain $\mathcal{X}$ is available which renders some parametric values $N \subset \Theta$ unrealistic. In the presence of this extra information the unrestricted maximum likelihood estimator (5) loses its appeal in favor of the constrained MLE

$$\hat{\theta}^{\text{cmle}}(x^{(1)}, \ldots, x^{(n)}) = \arg\max_{\theta \in \Theta \setminus N} \ell(\theta). \tag{6}$$

The constrained maximum likelihood (6) achieves a lower asymptotic error since its parameteric set is smaller (assuming its underlying assumption $\theta_0 \not\in N$ is correct). Though it is defendable on frequentist grounds the constrained MLE is often given a

Bayesian interpretation as the maximizer of the posterior under a prior assigning 0 probability to N and a uniform distribution over $\Theta \setminus N$.

The process of obtaining the set $\Theta \setminus N$ may rely on either domain knowledge or auxiliary dataset. In either case it is important to relate the constrained parametric subset $\Theta \setminus N$ to the corresponding set of possible probabilities

$$\mathcal{P}(\Theta \setminus N) = \{p_\theta(x) : \theta \in \Theta \setminus N\}\,.$$

Identifying p_θ as vectors of probabilities $(p_1, \ldots, p_{|\mathcal{X}|}) \in \mathbb{R}^{|\mathcal{X}|}$ we have that the constrained set of probabilities is a subset of the probability simplex $\mathcal{P}(\Theta \setminus N) \subset \mathbb{P}_{\mathcal{X}}$ where

$$\mathbb{P}_{\mathcal{X}} = \left\{ \left(p_1, \ldots, p_{|\mathcal{X}|}\right) : p_i \geq 0\,, \sum_{i=1}^{|\mathcal{X}|} p_i = 1 \right\}.$$

Above, we assume that the space $\mathcal{X}$ is finite turning the simplex $\mathbb{P}_{\mathcal{X}}$ of all possible distributions over $\mathcal{X}$ into a subset of a finite dimensional vector space. We maintain this assumption, which is standard in many structured prediction tasks, throughout the chapter in order to simplify the notation.

Expressing the constraints as a parametric subset $\Theta \setminus N$ is essential for deriving the constrained MLE estimator (6). Nevertheless, it is important to consider the corresponding subset of probabilities $\mathcal{P}(\Theta \setminus N)$ since it is much more interpretable for a domain expert and easy to test based on auxiliary data. In other words, it is much easier for a domain expert to specify constraints on the probabilities assigned by the model $\mathcal{P}(\Theta \setminus N)$ than constraints on abstract parameters $\Theta \setminus N$. The framework that we propose is thus to first specify the constrained probability set $\mathcal{P}(\Theta \setminus N)$ based on domain knowledge or auxiliary data, and then to convert it to $\Theta \setminus N$ in order to derive effective optimization schemes for the problem (6).

In many cases, the derivation of the set $\Theta \setminus N$ corresponding to $\mathcal{P}(\Theta \setminus N)$ is straightforward. For example in the case of the following simple exponential family model

$$p_\theta(x) = Z^{-1}(\theta)\exp\left(\sum_i \theta_i x_i\right) \quad x, \theta \in \mathbb{R}^d,$$

we have

$$p_\theta(x) > p_\theta(x') \quad \Leftrightarrow \quad \theta^\top(x - x') > 0. \tag{7}$$

In other cases, however, the conversion $\mathcal{P}(\Theta \setminus N) \Rightarrow \Theta \setminus N$ is highly non-trivial. This is also the case with conditional random fields which is the focus of this chapter.

We thus consider, in this chapter, the following problems in the context of conditional random fields

- Specifying a set of probability constraints $\mathcal{P}(\Theta \setminus N)$ based on domain knowledge or auxiliary data.
- Deriving the equivalent set of parametric constraints $\Theta \setminus N$.
- Deriving efficient algorithms for obtaining the constrained MLE.
- Experimental investigation of the benefit arising from the added constraints in the context of the structured prediction tasks of local sentiment analysis and information extraction.

3.1 Structured Prediction and Conditional Random Fields

Structured prediction is the task of associating a sequence of labels $\mathbf{y} = (y_1, \ldots, y_n), y_i \in \mathcal{Y}$ with a sequence of observed values $\mathbf{x} = (x_1, \ldots, x_n), x_i \in \mathcal{X}$. Two examples are NLP tagging where x_i are words and y_i are morphological or syntactic tags, and image processing where x_i are the pixel brightness values and y_i indicate the segment or object the pixel belongs to.

Conditional random fields (CRF) [59] are parametric families of conditional distributions $p_\theta(\mathbf{y}|\mathbf{x})$ that correspond to joint Markov random fields $p(\mathbf{y}, \mathbf{x})$ distributions

$$p_\theta(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{y}, \mathbf{x})}{\sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y})} = \frac{\prod_{C \in \mathcal{C}} \phi_C(\mathbf{x}|_C, \mathbf{y}|_C)}{Z(\mathbf{x}, \theta)}. \tag{8}$$

Above, $\mathcal{C}$ is the set of cliques in a graph over $\mathcal{X} \times \mathcal{Y}$ and $\mathbf{x}|_C$ and $\mathbf{y}|_C$ are the restriction of $\mathbf{x}$ and $\mathbf{y}$ to variables representing nodes in the clique $C \in \mathcal{C}$. The functions ϕ_C are arbitrary positive-valued functions called clique potentials and $Z(\theta, \mathbf{x})$ represents the conditional normalization term ensuring $\sum_{\mathbf{y}} p_\theta(\mathbf{y}|\mathbf{x}) = 1$ for all $\mathbf{x}, \theta$.

It is generally assumed that ϕ_C are exponential functions of features f_C modulated by decay parameters θ_C i.e.

$$\phi_C(\mathbf{x}|_C, \mathbf{y}|_C) = \exp\left(\sum_k \theta_{C,k} f_{C,k}(\mathbf{x}|_C, \mathbf{y}|_C)\right)$$

leading to the parametric family of conditional distributions

$$p_\theta(\mathbf{y}|\mathbf{x}) = Z^{-1}(\mathbf{x}, \theta) \exp\left(\sum_{C \in \mathcal{C}} \sum_k \theta_{C,k} f_{C,k}(\mathbf{x}|_C, \mathbf{y}|_C)\right) \quad \theta_{C,k} \in \mathbb{R}. \tag{9}$$

CRF models have been frequently applied to sequence annotation, where $\mathbf{x} = (x_1, \ldots, x_n)$ is a sequence of words and $\mathbf{y} = (y_1, \ldots, y_n)$ is a sequence of labels annotating the words. The standard graphical structure in this case is a chain structure on $y_1, \ldots, y_n$ with noisy observations $\mathbf{x}$ leading to the clique structure $\mathcal{C} = \{\{y_{i-1}, y_i\}, \{y_i, \mathbf{x}\} : i = 1, \ldots, n\}$ (see Figure 6, left). Note that this graphical structure is more general than the original chain CRF [59] and includes it as a special case.

Together with the standard choice of feature functions this leads to the CRF model

$$\begin{aligned} p_\theta(\mathbf{y}|\mathbf{x}) =& \frac{1}{Z(\mathbf{x}, \theta)} \cdot \\ & \exp\left(\sum_{i=1}^{n} \sum_{\sigma,\tau \in \mathcal{Y}} \lambda_{\langle\sigma,\tau\rangle} f_{\langle\sigma,\tau\rangle}(y_{i-1}, y_i) + \sum_{i=1}^{n} \sum_{\sigma \in \mathcal{Y}} \sum_{k=1}^{l} \mu_{\langle\sigma,A_k\rangle} g_{\langle\sigma,A_k\rangle}(y_i, \mathbf{x}, i)\right) \end{aligned} \tag{10}$$

where $\theta = (\lambda, \mu)$ is the parameter vector and

$$f_{\langle\sigma,\tau\rangle}(y_{i-1}, y_i) = 1_{\{y_{i-1}=\sigma\}} 1_{\{y_i=\tau\}} \qquad \sigma, \tau \in \mathcal{Y} \tag{11}$$

$$g_{\langle\sigma,A_k\rangle}(y_i, \mathbf{x}, i) = 1_{\{y_i=\sigma\}} A_k(\mathbf{x}, i) \qquad \sigma \in \mathcal{Y}. \tag{12}$$

The values σ, τ correspond to arbitrary values of labels in $\mathcal{Y}$ and A_k corresponds to binary functions of both observation $\mathbf{x}$ and some position i in the sequence. The choice of A_k is problem dependent. A common practice of choosing $A_k(\mathbf{x}, i) = 1_{\{x_i = w_k\}}, k = 1, \ldots, |\mathcal{X}|$ reduces the CRF model to its most traditional form measuring appearances of individual words in a vocabulary. More complex patterns of A_k may consider x_i as well as its neighbors x_{i-1} and x_{i+1} (e.g. $A_k(\mathbf{x}, i) = 1_{\{x_i = w, x_{i-1} = w'\}}$ for some $w, w' \in \mathcal{X}$), or consider properties other than word appearance (e.g. $A_k(\mathbf{x}, i) = 1_{\{x_i \text{ is capitalized}\}}$). The flexibility in the specification of A_k is the key advantage of CRF over generative sequential models such as hidden Markov models (HMMs). In particular, it enables the modeling of sequences of sentences rather than words as is the case of local sentiment prediction [69].

In the above formulation, we have $|\mathcal{Y}|^2$ feature functions $f_{\langle \sigma, \tau \rangle}$ measuring the transitions between successive label values and $|\mathcal{Y}| \cdot l$ feature functions $\{g_{\langle \sigma, A_k \rangle} : k = 1, \ldots, l, \sigma \in \mathcal{Y}\}$ describing observations $\mathbf{x}$ associated with label σ and function A_k. For the case of an m-order CRF where m is finite, it is possible to write the probabilistic model in the form of (10) by constructing $Y_i = (y_i, \ldots, y_{i+m-1})$, the ordered m-tuple of y_i values. Note, however, that not all transitions between states Y_i and Y_j are allowed for the m-order CRF.

Given a set of iid training samples $D = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) : i = 1, \ldots, m\}$ the parameters $\theta = (\lambda, \mu)$ are typically estimated by maximizing the regularized conditional likelihood

$$\ell(\theta|D) = \frac{1}{m} \sum_{i=1}^{m} \log p_\theta(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}) + C\|\theta\|_2^2 \tag{13}$$

which corresponds to the posterior under a Gaussian prior over θ. The maximum likelihood estimation is usually carried out using standard numeric techniques such as iterative scaling, conjugate gradient, or quasi-Newton. Other popular approaches of learning a CRF model include maximum margin Markov networks [106] where the model is trained discriminatively using a margin-based optimization problem, and Searn [27], an algorithm that decomposes a structured prediction problem into a set of classification problems solved by standard classification methods.

Unlike the situation in Markov random fields (Equation 7), the relationship between parameter and probability constraints in CRF is highly complicated. In particular, constraints over the probability vectors $p_\theta(\mathbf{y}|\mathbf{x}) \in [\alpha - \epsilon, \alpha + \epsilon]$ or $p_\theta(\mathbf{y}|\mathbf{x}) \geq p_\theta(\mathbf{y}'|\mathbf{x}')$ are not easily converted to the corresponding parametric constraints on θ. We explore in the next section several types of probability constraints that are intuitive and interpretable and yet correspond to simple ordering constraints on the parameters θ.

3.2 Ordered Domain Knowledge and Generalized Isotonic Constraints

In this section, we define a taxonomy of probability ordering constraints for CRF models based on probability ratios. These ordering constraints are intuitive and interpretable, and are easily specified using domain knowledge or auxiliary data. We derive the corresponding parameter constraints which we refer to as generalized isotonic constraints due to the similarity with the parameter constraints in the isotonic regression model [2].

Directly constraining the probability values assigned by the model

$$p_\theta(\mathbf{y}|\mathbf{x}) \in S \tag{14}$$

is impractical due to the large variability in the sequences $\mathbf{x}, \mathbf{y}$. It is difficult to imagine being able to ascertain probabilities of a large set of sequences $\mathbf{x}, \mathbf{y}$.

Another important difficulty in expressing direct probability constraints as in (14) is that it is hard to express domain knowledge in terms of absolute probabilities. Humans are notoriously bad at making statements concerning the probability of observing a particular event.

In this chapter, we propose a novel set of probability constraints which eliminates the two difficulties mentioned above, and have a simple corresponding parameter constraints. We resolve the first difficulty by dealing with constraints involving variability in a local region of the graph. For example, in the sentiment prediction task [69] we consider the effect an appearance of a particular word such as `superb` has on the probability of it conveying positive sentiment. We resolve the second difficulty by constraining probabilities ratios involving a text sequence $\mathbf{x}$ and a locally perturbed version of it. As we shall see, such constraints depend only on the perturbed variables and are independent of the values of $\mathbf{x}$ on the remainder of the graph.

Formally, we define the probability constraints in terms of a probability ratio $p_\theta(\mathbf{y}|\mathbf{x})/p_\theta(\mathbf{y}|\mathbf{x}')$ where $\mathbf{x}'$ is identical to $\mathbf{x}$, except on a small graph neighborhood. Thus, instead of specifying the precise probability value, we specify whether the perturbation $\mathbf{x} \mapsto \mathbf{x}'$ increases or decreases the conditional probability of $\mathbf{y}$. Surprisingly, we show that constraining probability ratios corresponds to simple partial order constraints on the parameters or parameter differences.

In the case of linear chain CRF, if we restrict ourselves to perturbations $\mathbf{x} \mapsto \mathbf{x}'$ that modify only the j-component of $\mathbf{x}$ in a simple way, the choices of $\{y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n\}$ and $\{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\}$ are immaterial making the probability ratio especially easy to assert and interpret.

We start with Proposition 1 below which relates the probability ratio to expectation over the parameters.

Proposition 1. *Let* $\mathbf{x}$ *be an arbitrary sequence over* $\mathcal{X}$ *and* $\mathbf{x}'$ *be identical to* $\mathbf{x}$ *except that* $A_v(\boldsymbol{x}', j) = 1$ *whereas* $A_v(\boldsymbol{x}, j) = 0$*. Then, for a linear chain CRF* $p_\theta(\mathbf{y}|\mathbf{x})$ *as in* (10) *we have*

$$\forall \mathbf{y} \qquad \frac{p_\theta(\mathbf{y}|\mathbf{x})}{p_\theta(\mathbf{y}|\mathbf{x}')} = E_{p_\theta(\mathbf{y}'|\mathbf{x})} \exp\left(\mu_{\langle y'_j, A_v\rangle} - \mu_{\langle y_j, A_v\rangle}\right). \tag{15}$$

Proof.

$$\begin{aligned}
\frac{p_\theta(\mathbf{y}|\mathbf{x})}{p_\theta(\mathbf{y}|\mathbf{x}')} &= \frac{Z(\mathbf{x}',\theta)}{Z(\mathbf{x},\theta)} \frac{\exp\left(\sum_{i,\sigma,\tau} \lambda_{\langle\sigma,\tau\rangle} f_{\langle\sigma,\tau\rangle}(y_{i-1},y_i) + \sum_{i,\sigma,k} \mu_{\langle\sigma,A_k\rangle} g_{\langle\sigma,A_k\rangle}(y_i,\mathbf{x},i)\right)}{\exp\left(\sum_{i,\sigma,\tau} \lambda_{\langle\sigma,\tau\rangle} f_{\langle\sigma,\tau\rangle}(y_{i-1},y_i) + \sum_{i,\sigma,k} \mu_{\langle\sigma,A_k\rangle} g_{\langle\sigma,A_k\rangle}(y_i,\mathbf{x}',i)\right)} \\
&= \frac{Z(\mathbf{x}',\theta)}{Z(\mathbf{x},\theta)} \exp\left(-\mu_{\langle y_j,A_v\rangle}\right) = \exp\left(-\mu_{\langle y_j,A_v\rangle}\right) \\
&\cdot \frac{\sum_{\mathbf{y}'} \exp\left(\sum_{i,\sigma,\tau} \lambda_{\langle\sigma,\tau\rangle} f_{\langle\sigma,\tau\rangle}(y'_{i-1},y'_i) + \sum_{i,\sigma,k} \mu_{\langle\sigma,A_k\rangle} g_{\langle\sigma,A_k\rangle}(y'_i,\mathbf{x}',i)\right)}{\sum_{\mathbf{y}'} \exp\left(\sum_{i,\sigma,\tau} \lambda_{\langle\sigma,\tau\rangle} f_{\langle\sigma,\tau\rangle}(y'_{i-1},y'_i) + \sum_{i,\sigma,k} \mu_{\langle\sigma,A_k\rangle} g_{\langle\sigma,A_k\rangle}(y'_i,\mathbf{x},i)\right)} \\
&= \exp\left(-\mu_{\langle y_j,A_v\rangle}\right) \frac{\sum_{r\in\mathcal{Y}} \alpha_r(\mathbf{x}) \exp\left(\mu_{\langle r,A_v\rangle}\right)}{\sum_{r\in\mathcal{Y}} \alpha_r(\mathbf{x})} \\
&= \sum_{r\in\mathcal{Y}} \frac{\alpha_r(\mathbf{x})}{\sum_{r'\in\mathcal{Y}} \alpha_{r'}(\mathbf{x})} \exp\left(\mu_{\langle r,A_v\rangle} - \mu_{\langle y_j,A_v\rangle}\right) \\
&= \sum_{\mathbf{y}'} p_\theta(\mathbf{y}'|\mathbf{x}) \exp\left(\mu_{\langle y'_j,A_v\rangle} - \mu_{\langle y_j,A_v\rangle}\right)
\end{aligned}$$

where

$$\alpha_r(\mathbf{x}) = \sum_{\mathbf{y}':y'_j=r} \exp\left(\sum_{i,\sigma,\tau} \lambda_{\langle\sigma,\tau\rangle} f_{\langle\sigma,\tau\rangle}(y'_{i-1},y'_i) + \sum_{i,\sigma,k} \mu_{\langle\sigma,A_k\rangle} g_{\langle\sigma,A_k\rangle}(y'_i,\mathbf{x},i)\right).$$

□

Proposition 1 is used below to derive two types of probability ordering constraints and their corresponding parametric constraints.

3.2.1 One-way Ordering

In one way ordering the probability ratios defined in Proposition 1 are constrained to follow a partial order. This results in a simple ordering between the corresponding parameters.

Proposition 2. *Let* $p_\theta(\mathbf{y}|\mathbf{x}), \mathbf{x}, \mathbf{x}'$ *be as in Proposition 1. For all label sequences* $\mathbf{s}, \mathbf{t}$*, we have*

$$\frac{p_\theta(\mathbf{s}|\mathbf{x})}{p_\theta(\mathbf{s}|\mathbf{x}')} \geq \frac{p_\theta(\mathbf{t}|\mathbf{x})}{p_\theta(\mathbf{t}|\mathbf{x}')} \qquad \Longleftrightarrow \qquad \mu_{\langle t_j,A_v\rangle} \geq \mu_{\langle s_j,A_v\rangle}. \tag{16}$$

Proof. By Proposition 1 we have

$$\log \frac{p_\theta(\mathbf{s}|\mathbf{x})}{p_\theta(\mathbf{s}|\mathbf{x}')} - \log \frac{p_\theta(\mathbf{t}|\mathbf{x})}{p_\theta(\mathbf{t}|\mathbf{x}')} = \mu_{\langle t_j, A_v\rangle} - \mu_{\langle s_j, A_v\rangle}.$$

Since $p_\theta(\cdot|\mathbf{x})$, $p_\theta(\cdot|\mathbf{x}')$ are strictly positive, Equation (16) follows. □

Surprisingly, the probability ratio inequality in Proposition 2 is equivalent to an ordering of only two parameters $\mu_{\langle t_j, A_v\rangle} \geq \mu_{\langle s_j, A_v\rangle}$. What makes this remarkable is that only the j-components of the sequences $\mathbf{t}, \mathbf{s}, \mathbf{x}$ matter making the remaining components immaterial. In particular, we can consider $\mathbf{s}, \mathbf{t}$ that are identical except for their j-components. In this case the interpretation of the probability ratio constraint in Proposition 2 is as follows: the perturbation $\mathbf{x} \mapsto \mathbf{x}'$ increases the probability of s_j more than it does the probability of t_j. Since $\mathbf{s}, \mathbf{t}$ and $\mathbf{x}, \mathbf{x}'$ differ in only the j-components such probability ratio constraints are relatively easy to specify and interpret.

Given a set of probability ratio constraints as in Proposition 2, we obtain a partial order on the parameters $\{\mu_{\langle \tau, A_j\rangle} : \tau \in \mathcal{Y}, j = 1, \ldots, l\}$ which corresponds to a partial order on the pairs $\{\langle \tau, A_j\rangle : \tau \in \mathcal{Y}, j = 1, \ldots, l\}$ i.e.,

$$\langle \tau, A_j\rangle \geq \langle \sigma, A_k\rangle \quad \text{if} \quad \mu_{\langle \tau, A_j\rangle} \geq \mu_{\langle \sigma, A_k\rangle}. \tag{17}$$

In particular fixing a certain A_v we get a partial order on $\mathcal{Y}$ corresponding to the ordering of $\{\mu_{\langle \tau, A_v\rangle} : \tau \in \mathcal{Y}\}$. In the case of sentiment prediction, the elements of $\mathcal{Y}$ correspond to opinions such as very negative, negative, objective, positive, very positive, associated with the standard order. A complete specification of probability ratio constraints would result in a full ordering over $\{\mu_{\langle \tau, A_v\rangle} : \tau \in \mathcal{Y}\}$ for some v. In this case, assuming that A_v measures the presence of word v, we have that if v corresponds to a positive word (e.g. `superb`) we obtain the ordering

$$\mu_{\langle \tau_1, A_v\rangle} \geq \cdots \geq \mu_{\langle \tau_{|\mathcal{Y}|}, A_v\rangle} \quad \text{where} \quad \tau_1 \geq \cdots \geq \tau_{|\mathcal{Y}|} \tag{18}$$

and the reverse ordering if v corresponds to a negative word (e.g. `horrible`)

$$\mu_{\langle \tau_1, A_v \rangle} \leq \cdots \leq \mu_{\langle \tau_{|\mathcal{Y}|}, A_v \rangle} \quad \text{where} \quad \tau_1 \geq \cdots \geq \tau_{|\mathcal{Y}|}. \tag{19}$$

3.2.2 Two-way Ordering

Two-way ordering is similar to one-way ordering but, in addition to the activation of a certain feature, it involves the deactivation of a second feature. The following proposition describes the probability constraints more formally and derives the corresponding parameter constraints. The proof is similar to that of Proposition 2 and is omitted.

Proposition 3. *Let* $\mathbf{x}$ *be a sequence over* $\mathcal{X}$ *in which* $A_v(\mathbf{x}, j) = 1$ *and* $A_w(\mathbf{x}, j) = 0$ *and* $\mathbf{x}'$ *be identical to* $\mathbf{x}$ *except that* $A_v(\mathbf{x}', j) = 0$ *and* $A_w(\mathbf{x}', j) = 1$*. Then for a linear chain CRF* $p_\theta(\mathbf{y}|\mathbf{x})$ *as in* (10) *we have that for all* $\mathbf{s}, \mathbf{t}$*,*

$$\frac{p_\theta(\mathbf{s}|\mathbf{x})}{p_\theta(\mathbf{s}|\mathbf{x}')} \geq \frac{p_\theta(\mathbf{t}|\mathbf{x})}{p_\theta(\mathbf{t}|\mathbf{x}')} \quad \Leftrightarrow \quad \mu_{\langle t_j, A_w \rangle} - \mu_{\langle s_j, A_w \rangle} \geq \mu_{\langle t_j, A_v \rangle} - \mu_{\langle s_j, A_v \rangle}. \tag{20}$$

In a similar way to the one-way ordering, the parameter constraint depends only on the j-components of $\mathbf{s}, \mathbf{t}$ and thus to aid the interpretation we can select $\mathbf{s}, \mathbf{t}$ that are identical except for s_j, t_j. The probability ratio constraint then measures whether perturbing $\mathbf{x} \mapsto \mathbf{x}'$ increases the probability of s_j more than that of t_j. However, in contrast to the one-way ordering the perturbation $\mathbf{x} \mapsto \mathbf{x}'$ involves deactivating the feature A_v and activating A_w. For example in the case of sentiment prediction these features could correspond to the replacement of word v in the j-position with word w.

In contrast to the one-way ordering, a collection of probability ratio constraints in Proposition 3 do not correspond to full or partial ordering on the model parameters. Instead they correspond to a full or partial order on the set of all pairwise differences between the model parameters.

One-way and two-way probability ratio constraints are complimentary in nature and they are likely to be useful in a wide variety of situations. In the case of elicitation from domain knowledge they provide a general framework for asserting statements that are immediately translatable to parameter constraints.

We conclude this section with the following observations regarding possible generalizations of the one-way and two-way constraints

(1) The definition of $f_{\langle\sigma,\tau\rangle}$ in (11) may be extended to a more general form $f_{\langle\sigma,\tau,B_k\rangle}(y_{i-1}, y_i, \mathbf{x}, i) = 1_{\{y_{i-1}=\sigma\}}1_{\{y_i=\tau\}}B_k(\mathbf{x}, i)$ where B_k are some binary functions of observation $\mathbf{x}$. Without loss of generality, we assume that the set $\{A_k\}$ and $\{B_k\}$ are disjoint. Otherwise, they can be made disjoint by defining a set of new parameters $\lambda_{\sigma,\tau,B_k} \leftarrow \lambda_{\sigma,\tau,B_k} + \mu_{\tau,B_k}$ corresponding to $f_{\sigma,\tau,B_k} \leftarrow f_{\sigma,\tau,B_k} + g_{\tau,B_k}$ for functions that appear in both $\{A_k\}$ and $\{B_k\}$. It is then straightforward to modify Proposition 1 - 3 with respect to parameters $\lambda_{\sigma,\tau,B_k}$.

(2) The simple form of parameter constraints on the right hand side of(16) and (20) results from the fact that only the j-components of the sequences matter in computing the probability ratio (15). For perturbations $\mathbf{x} \mapsto \mathbf{x}'$ involving labels from multiple positions in the sequence, the probability ratio constraints become linear parameter constraints with coefficients 1 or -1. These linear constraints are still considered simple, but they lose the intuitive ordering interpretation and are not the focus of this work.

3.3 Algorithms and Optimization

Conceptually, the parameter estimates for generalized isotonic CRF may be found by maximizing the likelihood or posterior subject to a collection of constraints of type (16) or (20). Since the constraints form a convex feasible set, the constrained MLE becomes a convex optimization problem with a unique global optimum. Unfortunately, due to the large number of possible constraints, a direct incorporation of them into a numerical maximizer is a relatively difficult task. We propose a re-parameterization of the CRF model that simplifies the constraints and converts the problem to a substantially easier constrained optimization problem. The re-parameterization, in the case of fully ordered parameter constraints is relatively straightforward. In the more general case of constraints forming a partially ordered set we need the mechanism of Möbius inversions on finite partially ordered sets.

The re-parameterization is based on the partial order on pairs $\{\langle \tau, A_j\rangle : \tau \in \mathcal{Y}, j = 1, \ldots, l\}$ defined in (17). Instead of enforcing the constraints on the original parameters $\mu_{\langle \tau, A_j\rangle}$, we reparameterize the model by introducing a new set of features $\{g^*_{\langle \sigma, A_k\rangle} : \sigma \in \mathcal{Y}, k = 1, \ldots, l\}$ defined as

$$g^*_{\langle \sigma, A_k\rangle}(y_i, x_i) = \sum_{\langle \tau, A_j\rangle : \langle \tau, A_j\rangle \geq \langle \sigma, A_k\rangle} g_{\langle \tau, A_j\rangle}(y_i, x_i) \tag{21}$$

and a new set of corresponding parameters $\mu^*_{\langle \sigma, A_k\rangle}$ satisfying the equality

$$\sum_{\sigma,k} \mu_{\langle \sigma, A_k\rangle} g_{\langle \sigma, A_k\rangle} = \sum_{\sigma,k} \mu^*_{\langle \sigma, A_k\rangle} g^*_{\langle \sigma, A_k\rangle} \tag{22}$$

and leading to the re-parameterized CRF

$$\begin{aligned} p_\theta(\mathbf{y}|\mathbf{x}) =& \frac{1}{Z(\mathbf{x}, \theta)} \cdot \\ & \exp\left(\sum_i \sum_{\sigma,\tau} \lambda_{\langle \sigma, \tau\rangle} f_{\langle \sigma, \tau\rangle}(y_{i-1}, y_i) + \sum_i \sum_{\sigma,k} \mu^*_{\langle \sigma, A_k\rangle} g^*_{\langle \sigma, A_k\rangle}(y_i, x_i) \right). \end{aligned} \tag{23}$$

Obtaining the maximum likelihood for the reparameterized model (23) instead of the original model has the benefit of converting the complex partial orders in (17) to simple non-negativity constraints $\mu^*_{\langle \sigma, A_k\rangle} \geq 0$ for a subset of the new parameters $\{\mu^*_{\langle \sigma, A_k\rangle} : \sigma \in \mathcal{Y}, k = 1, \ldots, l\}$. As a result, solving the constrained MLE problem on the reparameterized model (23) is substantially simpler to implement and is more efficient computationally. The constrained MLE can be computed in practice using a trivial adaptation of gradient based methods such as conjugate gradient or quasi-Newton.

The parameters $\mu^*_{\langle\sigma,A_k\rangle}$ may be obtained from the original parameters by convolving $\mu_{\langle\sigma,A_k\rangle}$ with the Möbius function of the partially ordered set (17). The reparameterization (23) is justified by the Möbius inversion theorem which states that $\mu^*_{\langle\sigma,A_k\rangle}$ satisfy

$$\mu_{\langle\sigma,A_k\rangle} = \sum_{\langle\tau,A_j\rangle:\langle\tau,A_j\rangle\leq\langle\sigma,A_k\rangle} \mu^*_{\langle\tau,A_j\rangle}. \tag{24}$$

In the case of two-way ordering, we have ordering on parameter differences rather than the parameters themselves. The mechanism of Möbius inversions can still be applied, but over a transformed feature space instead of the original feature space. In particular, for (t_j, s_j, A_w, A_v) that satisfy (20), we apply the re-parameterization described in (21) - (23) to the feature functions $\tilde{g}$ defined by

$$\begin{aligned} \tilde{g}_{\langle t_j,A_v\rangle} &= g_{\langle t_j,A_v\rangle} & \tilde{g}_{\langle s_j,A_v\rangle} &= g_{\langle s_j,A_v\rangle} + g_{\langle t_j,A_v\rangle} \\ \tilde{g}_{\langle t_j,A_w\rangle} &= g_{\langle t_j,A_w\rangle} & \tilde{g}_{\langle s_j,A_w\rangle} &= g_{\langle s_j,A_w\rangle} + g_{\langle t_j,A_w\rangle} \end{aligned}$$

and parameters $\tilde{\mu}$ defined by

$$\begin{aligned} \tilde{\mu}_{\langle s_j,A_v\rangle} &= \mu_{\langle s_j,A_v\rangle} & \tilde{\mu}_{\langle t_j,A_v\rangle} &= \mu_{\langle t_j,A_v\rangle} - \mu_{\langle s_j,A_v\rangle} \\ \tilde{\mu}_{\langle s_j,A_w\rangle} &= \mu_{\langle s_j,A_w\rangle} & \tilde{\mu}_{\langle t_j,A_w\rangle} &= \mu_{\langle t_j,A_w\rangle} - \mu_{\langle s_j,A_w\rangle}. \end{aligned}$$

More information concerning the Möbius inversion theorem for partially ordered sets may be found in standard textbooks on combinatorics, for example [104].

3.4 Elicitation of Constraints

There are two ways in which probability constraints such as the ones in Propositions 2 and 3 can be elicited. The first is by eliciting domain knowledge from experts. This is similar to prior elicitation in subjective Bayesian analysis, but has the advantage that the knowledge is specified in terms of probability ratios, rather than model parameters.

The second way to elicit probability constraints is by relying on auxiliary data. The auxiliary data should be related to the domain on which inference is conducted, but does not have to have the same distribution as the training data. Automatic elicitation results in probability ratios satisfying inequalities or more generally having values in some sets. As such, some amount of inconsistency between the auxiliary data and the train and test data is permissible. For example, in sentiment prediction modeling of a particular author, we may have auxiliary data written by another author. In information extraction we may have a secondary corpus from a different source whose label taxonomy is related to the primary dataset.

Inferring probability constraints concerning the full conditionals $p_\theta(\mathbf{y}|\mathbf{x})$ from data is difficult due to the fact that each sequence $\mathbf{x}$ or $\mathbf{y}$ appears only once or a small number of times. The approach below makes some conditional independence assumptions which will simplify the elicitation to the problem of ordering probability ratios of univariate conditional distributions $p(A_v|t_j)/p(A_w|t_j)$.

Proposition 4. *Let* $\mathbf{x}, \mathbf{x}'$ *be as in Proposition 1 and* $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p_\theta(\mathbf{y}|\mathbf{x})$ *where* $p_\theta(\mathbf{y}|\mathbf{x})$ *is a CRF model and* $p(x_j|y_j)$ *is being modeled by*[1] $p\left(\cap_{k\in I} A_k \mid y_j\right)$*,* $I = \{k \in \{1, \ldots, l\} : x_j \in A_k\}$*, satisfying the following conditional independencies*

$$p\left(\bigcap_{k\in I} A_k \mid y_j\right) = \prod_{k\in I} p\left(A_k \mid y_j\right). \tag{25}$$

If the CRF model satisfies (16) *then*

$$p(A_v|t_j) \geq p(A_v|s_j). \tag{26}$$

[1] *We implicitly assume here that sequences* $\boldsymbol{x}$ *are identified by their feature signature i.e. the feature functions constitute a 1-1 mapping. In some cases this does not hold and some correction term is necessary.*

Proof. We have

$$\begin{aligned}\text{LHS of (16)} \quad &\Rightarrow \quad \frac{p_\theta(\mathbf{s}|\mathbf{x})}{p_\theta(\mathbf{t}|\mathbf{x})} \geq \frac{p_\theta(\mathbf{s}|\mathbf{x}')}{p_\theta(\mathbf{t}|\mathbf{x}')} \quad \Rightarrow \quad \sum_{\mathbf{s}} \frac{p_\theta(\mathbf{s}|\mathbf{x})}{p_\theta(\mathbf{t}|\mathbf{x})} \geq \sum_{\mathbf{s}} \frac{p_\theta(\mathbf{s}|\mathbf{x}')}{p_\theta(\mathbf{t}|\mathbf{x}')} \\ &\Rightarrow \quad \sum_{\mathbf{t}} \frac{p_\theta(\mathbf{t}|\mathbf{x})}{\sum_{\mathbf{s}} p_\theta(\mathbf{s}|\mathbf{x})} \leq \sum_{\mathbf{t}} \frac{p_\theta(\mathbf{t}|\mathbf{x}')}{\sum_{\mathbf{s}} p_\theta(\mathbf{s}|\mathbf{x}')} \Rightarrow \quad \frac{\alpha_{t_j}(\mathbf{x})}{\alpha_{s_j}(\mathbf{x})} \leq \frac{\alpha_{t_j}(\mathbf{x}')}{\alpha_{s_j}(\mathbf{x}')} \qquad (27)\end{aligned}$$

where the summations are over all label sequences $\mathbf{s}, \mathbf{t}$ having fixed j-components s_j, t_j. See Proposition 1 for a definition of $\alpha_{t_j}, \alpha_{s_j}$.

Due to the conditional independencies expressed in the graphical structure of CRF, the α_r functions satisfy

$$\begin{aligned}\alpha_r(\mathbf{x})/Z(\mathbf{x}) &= \sum_{\mathbf{y}:y_j=r} p_\theta(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{y}:y_j=r} \frac{p(\mathbf{y},\mathbf{x})}{p(\mathbf{x})} = \sum_{y_{-j}} \frac{p(y_{-j}, y_j = r, \mathbf{x})}{p(\mathbf{x})} \\ &= \frac{p(y_j = r, x_j, x_{-j})}{p(\mathbf{x})} = \frac{p(x_j|y_j = r)p(y_j = r|x_{-j})p(x_{-j})}{p(\mathbf{x})} \qquad (28)\end{aligned}$$

where $y_{-j} = \{y_1, \ldots, y_n\} \setminus \{y_j\}$ and $x_{-j} = \{x_1, \ldots, x_n\} \setminus \{x_j\}$.

Substituting (28) in (27) and using the fact that for all $r \in \mathcal{Y}$, $p(y_j = r|x_{-j}) = p(y_j = r|x'_{-j})$ we get

$$\begin{aligned}(27) \quad &\Rightarrow \quad \frac{p(x_j|t_j)}{p(x_j|s_j)} \leq \frac{p(x'_j|t_j)}{p(x'_j|s_j)} \quad \Rightarrow \quad \frac{p(x_j|t_j)}{p(x'_j|t_j)} \leq \frac{p(x_j|s_j)}{p(x'_j|s_j)} \\ &\Rightarrow \quad \frac{p\left(\cap_{k\in I} A_k|t_j\right)}{p\left(\left(\cap_{k\in I} A_k\right) \cap A_v|t_j\right)} \leq \frac{p\left(\cap_{k\in I} A_k|s_j\right)}{p\left(\left(\cap_{k\in I} A_k\right) \cap A_v|s_j\right)} \\ &\Rightarrow \quad p(A_v|t_j) \geq p(A_v|s_j)\end{aligned}$$

where the last implication comes from the conditional independence assumption (25).

□

A similar result holds for two-way ordering whose proof is omitted.

Proposition 5. *Under the same conditions as Proposition 4, if the CRF model satisfies* (20) *then*

$$\frac{p(A_v|t_j)}{p(A_w|t_j)} \leq \frac{p(A_v|s_j)}{p(A_w|s_j)} \quad \Rightarrow \quad \frac{p(t_j|A_v)}{p(t_j|A_w)} \leq \frac{p(s_j|A_v)}{p(s_j|A_w)}. \qquad (29)$$

Constraints such as (26) or the equivalent (but sometimes easier to estimate)

$$\frac{p(t_j|A_v)}{p(t_j)} \geq \frac{p(s_j|A_v)}{p(s_j)} \tag{30}$$

can be obtained from auxiliary data based on hypothesis tests. More specifically, Equations (29),(30) can be written as $\psi \geq 1$, where ψ is estimated by the odds ratio of a 2×2 continency table obtained from the co-occurrence of a label (s_j or t_j) and a set (A_v or A_w). This can be achieved by a test of independence in a 2×2 table, such as an asymptotic test based on the test statistic $(\log \hat{\psi} - \log \psi)/\mathrm{se}(\log \hat{\psi}) \approx \mathcal{N}(0,1)$ where se is the standard error [99].

For a large number of constraints, a collection of hypothesis tests can be performed offline. Ideally, we would like to control the familywise error rate (FWER). One way to retain a prescribed FWER is to test each hypothesis at a higher significant level, as is done in the Bonferroni correction for multiple hypothesis testing. However, the number of hypotheses considered in our case is hundreds or even thousands, while the field of multiple hypothesis testing for this problem size is still largely under development. Therefore we resort to single hypothesis testing and adopt a heuristic approach, i.e. selecting a certain value as threshold we order the hypothesis by their p-values and select the ones whose p-values are less than the threshold.

The derivations above are based on the conditional independence assumption (25) which may be too restrictive for arbitrary feature sets. However, in our experiments we found that the constraints identified automatically by hypothesis tests normally overlap with those returned by domain experts. Moreover, even if domain experts are available and human elicitation is taking place, the automatic elicitation described above can substantially reduce human intervention as it can be used to pre-filter a large set of unnecessary features.

Conflicting constraints are rare, but possible. In general, we expect the probability of observing conflict will be low for high dimensional problems where the number of constraints are relatively small compared with the number of features. Detecting conflicting constraints for one-way ordering is relatively easy. Note in this case the problem can be decomposed into a collection of subproblems each detecting conflicts for some feature A_v. In particular, we construct a directed graph for each feature A_v as follows: let the vertices of the graph correspond to labels from $\mathcal{Y}$. For identified constraint $\mu_{\langle t_j, A_v\rangle} \geq \mu_{\langle s_j, A_v\rangle}$, we add an edge to the graph that points from s_j to t_j. Once all constraints that relate to A_v have been enumerated, we may detect whether any conflict exists by cycle detection algorithms such as a modification of depth-first search.

The case for two-way ordering is more complicated. Fix a pair of labels or a pair of observations, the problem is reduced to the one-way ordering case. To see this, for fixed label pair (t_j, s_j) we may construct a directed graph with each vertex corresponding to a feature A_v. Each identified constraint $\mu_{\langle t_j, A_w\rangle} - \mu_{\langle s_j, A_w\rangle} \geq \mu_{\langle t_j, A_v\rangle} - \mu_{\langle s_j, A_v\rangle}$ is therefore represented as an edge in the graph that points from A_v to A_w. Alternatively, when fixing features A_w and A_v, the constraint is equivalent to $\mu_{\langle t_j, A_w\rangle} - \mu_{\langle t_j, A_v\rangle} \geq \mu_{\langle s_j, A_w\rangle} - \mu_{\langle s_j, A_v\rangle}$ which can be represented by an edge pointing from s_j to t_j. We may repeat the procedure by alternating between label pairs and observation pairs. The resulting constraint set is very unlikely to contain conflicting constraints, though still possible.

Whether the two-way ordering constraint set is contradiction free or not may also be answered by the following linear programming problem

$$\begin{aligned} \min \quad & \sum_{i=1}^{l} \xi_i \\ \text{s.t.} \quad & \mu_{\langle t_j^i, A_{w^i}\rangle} - \mu_{\langle s_j^i, A_{w^i}\rangle} - \mu_{\langle t_j^i, A_{v^i}\rangle} + \mu_{\langle s_j^i, A_{v^i}\rangle} + \xi_i \geq 0 \\ & \xi_i \geq 0, \quad i = 1, \ldots, l \end{aligned}$$

where l is the number of constraints and ξ_i is the slack variable introduced for the i-th constraint. The statement that the constraint set is contradiction free is equivalent to the condition where the optimal value for ξ_i are all zero. The size of the problem is proportional to the number of constraints, which in general varies from hundreds to thousands. The linear programming can be solved rather efficiently for this problem size. Moreover, this approach is quite general and applies to arbitrary linear constraint set, such as a mixture of one-way and two-way ordering constraints.

3.5 Sentiment Prediction

Many documents, such as reviews and blogs, are written with the purpose of conveying a particular opinion or sentiment. Other documents may not be written with the purpose of conveying an opinion, but nevertheless they contain one. Opinions, or sentiments, may be considered in several ways, the simplest of which is varying from positive opinion, through neutral, to negative opinion.

We distinguish between the tasks of global sentiment prediction and local sentiment prediction. Global sentiment prediction is the task of predicting the sentiment of the document based on the word sequence. Local sentiment prediction [69] is the task of predicting a sequence of sentiments $\mathbf{y} = (y_1, \ldots, y_n), y_i \in \mathcal{Y}$ based on a sequence of sentences $\mathbf{x} = (x_1, \ldots, x_n)$. In this case, each sentiment measures the local sentiment of the sentence x_i in the document.

Previous research on sentiment prediction has generally focused on predicting the sentiment of the entire document. A commonly used application is the task of predicting the number of stars assigned to a movie, based on a review text. Typically, the problem is considered as standard multiclass classification or regression using the bag of words representation.

In addition to the sentiment of the entire document, which we call global sentiment, we define the concept of local sentiment as the sentiment associated with a particular part of the text. It is reasonable to assume that the global sentiment of a document is a function of the local sentiment and that estimating the local sentiment is a key step in predicting the global sentiment. Moreover, the concept of local sentiment is useful in a wide range of text analysis applications including document summarization and visualization.

Formally, we view local sentiment as a function on the sentences in a document taking values in a finite partially ordered set, or a poset, $(\mathcal{Y}, \leq)$. To determine the local sentiment at a particular word, it is necessary to take context into account. For example, due to context the local sentiment at each of the following words `this is a horrible product` is low. Since sentences are natural components for segmenting document semantics, we view local sentiment as a piecewise constant function on sentences. Occasionally we encounter a sentence that violates this rule and conveys opposing sentiments in two different parts. In this situation we break the sentence into two parts and consider them as two sentences. We therefore formalize the problem as predicting a sequence of sentiments $\mathbf{y} = (y_1, \ldots, y_n), y_i \in \mathcal{Y}$ based on a sequence of sentences $\mathbf{x} = (x_1, \ldots, x_n)$ where we consider each sentence as a bag of words $x_i = \{w_{i1}, \ldots, w_{il_i}\}$.

We examine the performance of the CRF model in the local sentiment task and the benefit arising from incorporating parameter constraints through auxiliary data and domain knowledge. The CRF is based on Equation (10) with the feature functions $A_k(\mathbf{x}, i) = 1_{\{w_k \in x_i\}}$ that measure the appearance of vocabulary words in each sentence. The dataset that we use contains 249 movie reviews, randomly selected from the Cornell sentence polarity dataset v1.0[2], all written by the same author. The local sentiment labeling was performed manually by the author by associating with each sentence one of the following sentiment values $\mathcal{Y} = \{-2, -1, 0, 1, 2\}$ where 2 corresponds to highly praised, 1 corresponds to something good, 0 corresponds to objective description, -1 corresponds to something that needs improvement, and -2 corresponds to strong aversion.

[2] Available at `http://www.cs.cornell.edu/People/pabo/movie-review-data`

3.5.1 Sentence Level Sentiment Prediction

Figure 2 displays the testing accuracy of local sentiment prediction both as a function of the training data size and as a function of the number of constrained words averaged over 40 train-test splits. In all cases, limited memory BFGS was used to train the CRF. The constraints were enforced using the barrier method. The objective function was the regularized MLE with a Gaussian prior on θ with variance 10.

The dataset presents one particular difficulty where more than 75% of the sentences are labeled objective (or 0). As a result, the prediction accuracy for objective sentences is over-emphasized. To correct for this fact, we report our results by averaging the test-set performance for each individual label. Note that since there are 5 labels, random guessing yields a baseline of 0.2 accuracy.

As described in Section 3.4, for one-way ordering, we obtained 500 words from an auxiliary data set that received the smallest p values in a test of (30) to set the constraints (16). The auxiliary data set is the additional 201 movie reviews from a second author described in 3.5.3. Table 1 displays the top 15 positive and negative words. The constraint set is contradiction free when positive and negative words form two disjoint sets.

Similarly, we may apply a test of (29) on the auxiliary data set to get pairs of words for setting constraints of (20). Figure 3 shows a portion of the graph by connecting a pair of ordered words with a line where the arrow points to the higher ordered word. Detecting conflicting constraints is equivalent to detecting cycles in this directed graph. A total of 400 pairs of words are selected for two-way ordering constraints in Figure 2 (bottom left).

The results in Figure 2 indicate that by incorporating either one-way or two-way ordering information, the generalized isotonic CRF perform consistently better than regular CRF. The advantage of incorporating sequential information in sentiment

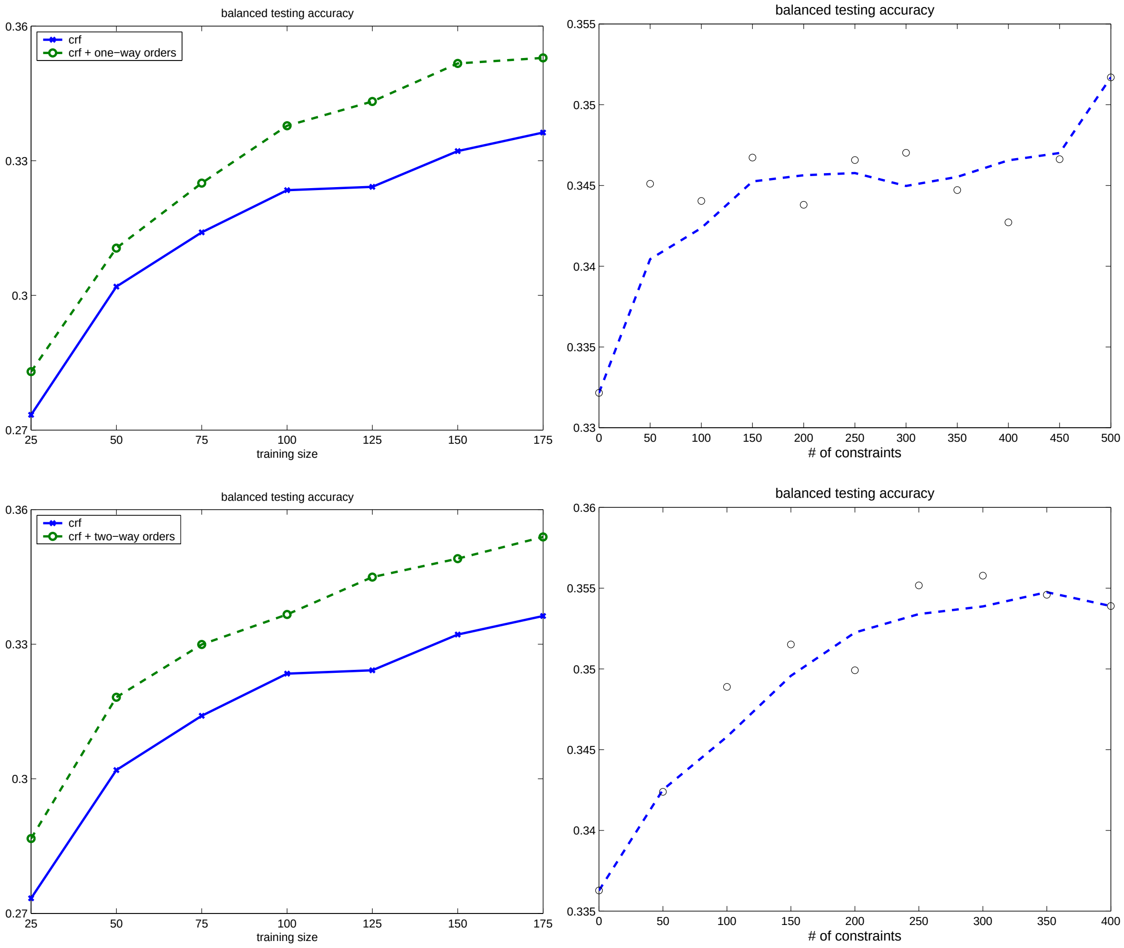


Figure 2: Balanced test accuracy for local sentiment prediction both as a function of training size (left column) and as a function of number of constrained words (right column). 500 words that received the smallest p values in a test of (30) are subject to one-way ordering (top row). 400 pairs of words that received the smallest p values in a test of (29) are subject to two-way ordering (bottom row). Blue lines in the right column are obtained by smoothing the data (represented by black circles). In this case, the training size is fixed to be 150.

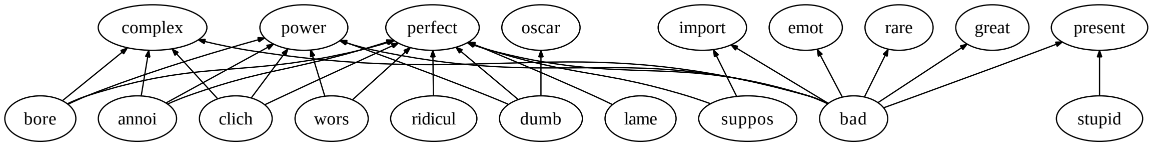


Figure 3: Ordering of stemmed words with respect to the positive sentiment. The words with higher order are drawn at the top.

Table 1: Lists of first 15 positive or negative stemmed words with the smallest p values.

great	perfect	power	love	complex
import	emot	present	fascin	rare
oscar	true	simpl	polit	beauti
bad	suppos	bore	stupid	wors
dumb	minut	tediou	annoi	wrong
bland	ridicul	worst	lifeless	lame

prediction has already been demonstrated in [69] and we therefore omit results comparing CRF and isotonic CRF with non-sequential models such as naive Bayes or SVM here.

We note that the information provided by one-way and two-way ordering are somewhat overlapping. For example, setting the words `great` and `bad` for one-way ordering automatically implies that the word pair (`great`, `bad`) satisfies the two-way ordering. We therefore avoid considering generalized isotonic CRF with mixed constraint types.

3.5.2 Global Sentiment Prediction

We also evaluated the contribution of the local sentiment analysis in helping to predict the global sentiment of documents. The sentence-based definition of sentiment flow is problematic when we want to fit a model that uses sentiment flows from multiple documents. Different documents have different number of sentences and it is not clear how to compare them or how to build a model from a collection of discrete flows of different lengths. We therefore convert the sentence-based flow to a smooth length-normalized flow that can meaningfully relate to other flows.

In order to account for different lengths, we consider the sentiment flow as a function $h : [0, 1] \to \mathcal{Y} \subset \mathbb{R}$ that is piecewise constant on the intervals $[0, l), [l, 2l), \ldots, [(k-1)l, 1]$ where k is the number of sentences in the document and $l = 1/k$. Each of the

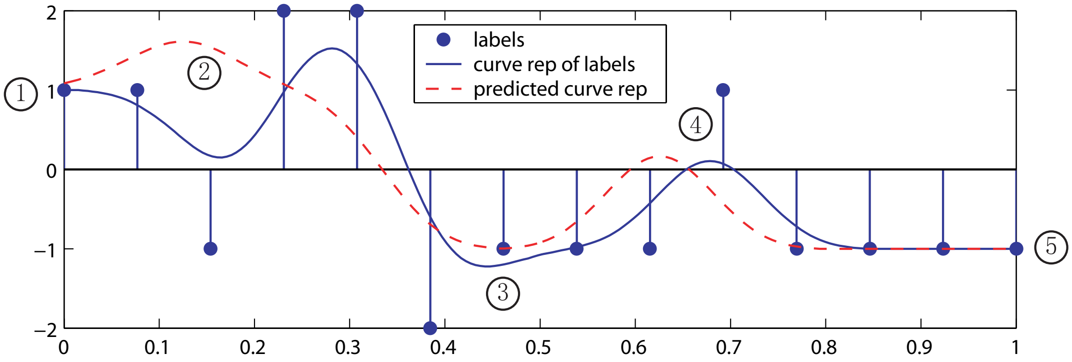


Figure 4: Sentiment flow and its smoothed curve representation. The blue circles indicate the labeled sentiment of each sentence. The blue solid curve and red dashed curve are smoothed representations of the labeled and predicted sentiment flows. Only non-objective labels are kept in generating the two curves. The numberings correspond to sentences displayed in Section 3.5.4.

intervals represents a sentence and the function value on it is its sentiment.

To create a more robust representation we smooth out the discontinuous function by convolving it with a smoothing kernel. The resulting sentiment flow is a smooth curve $f : [0,1] \to \mathbb{R}$ that can be easily related or compared to similar sentiment flows of other documents (see Figure 4 for an example). We can then define natural distances between two flows, for example the L_p distance

$$d_p(f_1, f_2) = \left(\int_0^1 |f_1(r) - f_2(r)|^p \, dr \right)^{1/p} \tag{31}$$

for use in a distance based classifier that predicts the global sentiment.

We compared a nearest neighbor classifier for the global sentiment, where the representation varied from bag of words to smoothed length-normalized local sentiment representation (with and without objective sentences). The smoothing kernel was a bounded Gaussian density (truncated and renormalized) with $\sigma^2 = 0.2$. Figure 4 displays discrete and smoothed local sentiment labels, and the smoothed sentiment flow predicted by isotonic CRF.

Figure 5 and Table 2 display test-set accuracy of global sentiment prediction as a function of the train set size. The distance in the nearest neighbor classifier was either L_1 or L_2 for the bag of words representation or their continuous version

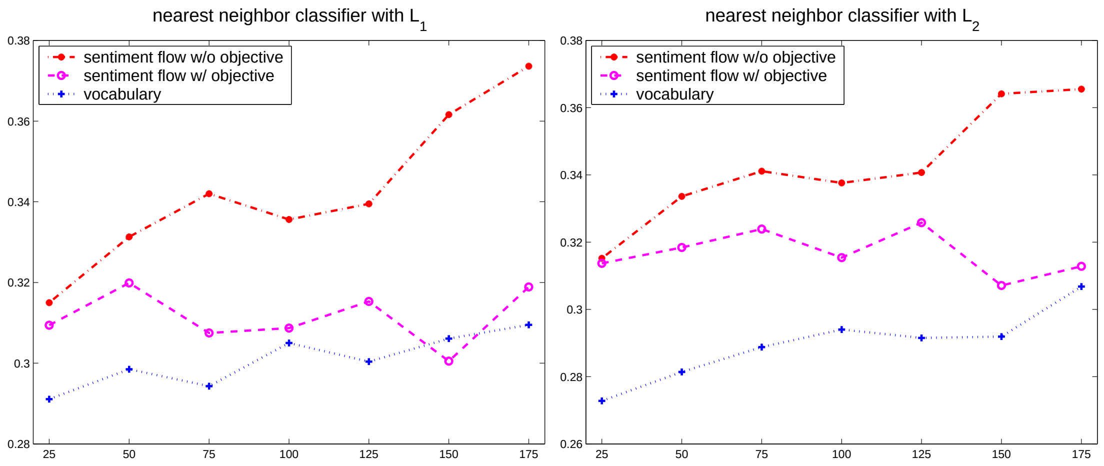


Figure 5: Accuracy of global sentiment prediction (4-class labeling) as a function of train set size.

Table 2: Accuracy results and relative improvement when training size equals 175.

	L_1		L_2	
vocabulary	0.3095		0.3068	
sentiment flow with objective sentences	0.3189	3.0%	0.3128	1.95%
sentiment flow without objective sentences	0.3736	20.7%	0.3655	19.1%

(31) for the smoothed sentiment curve representation. The results indicate that the classification performance of the local sentiment representation is better than the bag of words representation. In accordance with the conclusion of [80], removing objective sentences (that correspond to sentiment 0) increased the local sentiment analysis performance by 20.7%. We can thus conclude that for the purpose of global sentiment prediction, the local sentiment flow of the non-objective sentences holds most of the relevant information.

3.5.3 Measuring the rate of sentiment change

Thus far, we have ignored the dependency of the labeling model $p_\theta(\mathbf{y}|\mathbf{x})$ on the author, denoted here by the variable a. We now turn to account for different sentiment-authoring styles by incorporating this variable into the model. The word emissions

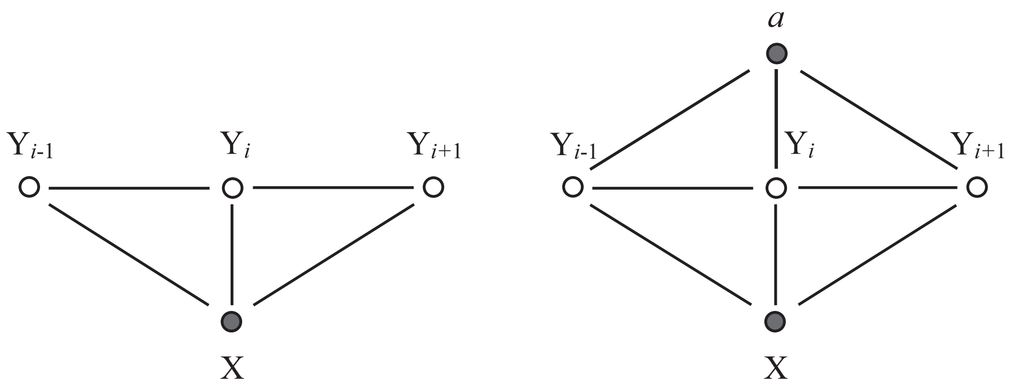


Figure 6: Graphical models corresponding to CRF (left) and author-dependent CRF (right).

$y_i \rightarrow w_i$ in the CRF structure are not expected to vary much across different authors. The sentiment transitions $y_{i-1} \rightarrow y_i$, on the other hand, typically vary across different authors as a consequence of their individual styles. For example, the review of an author who sticks to a list of self-ranked evaluation criteria is prone to strong sentiment variations. In contrast, the review of an author who likes to enumerate pros before he gets to cons (or vice versa) is likely to exhibit more local homogeneity in sentiment.

Accounting for author-specific sentiment transition style leads to the graphical model in Figure 6.

The corresponding author-dependent CRF model

$$p_\theta(\mathbf{y}|\mathbf{x},a) = \frac{1}{Z(\mathbf{x},a)} \exp\left(\sum_{i,a'} \sum_{\sigma,\tau} \left(\lambda_{\langle\sigma,\tau\rangle} + \lambda_{\langle\sigma,\tau,a'\rangle}\right) f_{\langle\sigma,\tau,a'\rangle}(y_{i-1}, y_i, a) \right.$$
$$\left. + \sum_i \sum_{\sigma,k} \mu_{\langle\sigma,A_k\rangle} g_{\langle\sigma,A_k\rangle}(y_i, \mathbf{x}, i) \right)$$

uses features $f_{\langle\sigma,\tau,a'\rangle}(y_{i-1}, y_i, a) = f_{\langle\sigma,\tau\rangle}(y_{i-1}, y_i)\delta_{a,a'}$ and transition parameters that are author-dependent $\lambda_{\langle\sigma,\tau,a\rangle}$ as well as author-independent $\lambda_{\langle\sigma,\tau\rangle}$. Setting $\lambda_{\langle\sigma,\tau,a\rangle} = 0$ reduces the model to the standard CRF model. The author-independent parameters $\lambda_{\langle\sigma,\tau\rangle}$ allow parameter sharing across multiple authors in case the training data is too scarce for proper estimation of $\lambda_{\langle\sigma,\tau,a\rangle}$. For simplicity, the above ideas are described in the context of non-isotonic CRF. However, it is straightforward to combine author-specific models with generalized isotonic restrictions.

We examine the rate of sentiment change as a characterization of the author's writing style using the isotonic author-dependent model. We assume that the CRF process is a discrete sampling of a corresponding continuous time Markov jump process. A consequence of this assumption is that the time T the author stays in sentiment σ before leaving is modeled by the exponential distribution $p_\sigma(T > t) = e^{-q_\sigma(t-1)}, t > 1$. Here, we assume $T > 1$ and q_σ is interpreted as the rate of change of the sentiment $\sigma \in \mathcal{Y}$: the larger the value, the more likely the author will switch to other sentiments in the near future.

To estimate the rate of change q_σ of an author we need to compute $p_\sigma(T > t)$ based on the marginal probabilities $p(\mathbf{s}|a)$ of sentiment sequences $\mathbf{s}$ of length l. The probability $p(\mathbf{s}|a)$ may be approximated by

$$p(\mathbf{s}|a) = \sum_{\mathbf{x}} p(\mathbf{x}|a) p_\theta(\mathbf{s}|\mathbf{x}, a) \approx \sum_{\mathbf{x}} \frac{\tilde{p}'(\mathbf{x}|a)}{n - l + 1} \times \left(\sum_i \frac{\alpha_i(s_1|\mathbf{x}, a) \prod_{j=i+1}^{i+(l-1)} M_j(s_{j-i}, s_{j-i+1}|\mathbf{x}, a) \beta_{i+(l-1)}(s_l|\mathbf{x}, a)}{Z(\mathbf{x}, a)} \right)$$

where $\tilde{p}'$ is the empirical probability function $\tilde{p}'(\mathbf{x}|a) = \frac{1}{|C|} \sum_{\mathbf{x}' \in C} \delta_{\mathbf{x},\mathbf{x}'}$ for the set C of documents written by author a of length no less than l. α, M, β are the forward, transition and backward probabilities analogous to the dynamic programming method in [59].

Using the model $p(\mathbf{s}|a)$ we can compute $p_\sigma(T > t)$ for different authors at integer values of t which would lead to the quantity q_σ associated with each author. However, since (32) is based on an approximation, the calculated values of $p_\sigma(T > t)$ will be noisy resulting in slightly different values of q_σ for different time points t and cross validation iterations. A linear regression fit for q_σ based on the approximated values of $p_\sigma(T > t)$ for two authors using 10-fold cross validation is displayed in Figure 7. The data was the 249 movie reviews from the previous experiments written by one author, and additional 201 movie reviews from a second author. Interestingly, the author associated with the red dashed line has a consistent lower q_σ value in all those figures, and thus is considered as more "static" and less prone to quick sentiment variations.

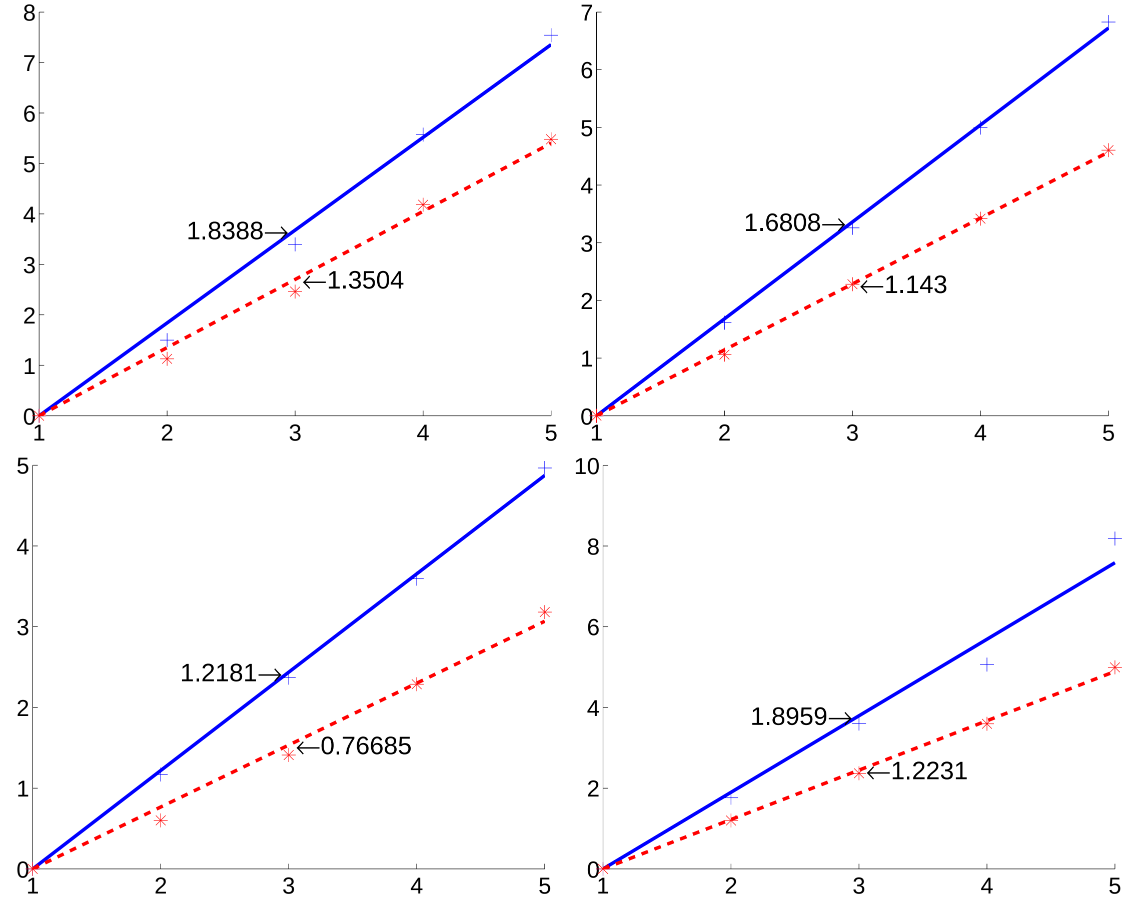


Figure 7: Linear regression fit for q_σ, $\sigma = 2, 1, -1, -2$ (left to right) based on approximated values of $p_\sigma(T > t)$ for two different authors. X-axis: time t; Y-axis: negative log-probability of $T > t$.

3.5.4 Text Summarization

We demonstrate the potential usage of sentiment flow for text summarization with a very simple example. The text below shows the result of summarizing the movie review in Figure 4 by keeping only sentences associated with the start, the end, the top, and the bottom of the predicted sentiment curve. The number before each sentence relates to the circled number in Figure 4.

> 1 What makes this film mesmerizing, is not the plot, but the virtuoso performance of Lucy Berliner (Ally Sheedy), as a wily photographer, retired from her professional duties for the last ten years and living with a has-been German actress, Greta (Clarkson). 2 The less interesting story line involves the ambitions of an attractive, baby-faced assistant editor at the magazine, Syd (Radha Mitchell), who lives with a boyfriend (Mann) in an emotionally chilling relationship. 3 We just lost interest in the characters, the film began to look like a commercial for a magazine that wouldn't stop and get to the main article. 4 Which left the film only somewhat satisfying; it did create a proper atmosphere for us to view these lost characters, and it did have something to say about how their lives are being emotionally torn apart. 5 It would have been wiser to develop more depth for the main characters and show them to be more than the superficial beings they seemed to be on screen.

3.5.5 Elicitation of Constraints from Domain Experts

In all previous experiments, the probability ordering constraints are obtained by testing hypotheses such as (30) or (29) on the auxiliary data set. We now demonstrate that we may achieve similar or even better results by eliciting constraints from domain experts.

During the experiment, one of the authors was presented with the vocabulary of the sentiment data set, and was asked to pick a subset of words from it which they thought would indicate either positive or negative sentiment. A total of 402 words were picked, and a subset of them starting with 'a' are listed in Table 3.

This set of words are then used to define one-way ordering constraints for CRF corresponding to a full ordering on the labels $\mathcal{Y}$. Figure 8 shows the test-set performance as a function of training size averaged over 40 cross validations. Compared with Figure 2 (top left), applying domain knowledge directly achieves similar, or even

Table 3: Stemmed words starting with 'a' that are chosen manually to conveying positive or negative sentiment.

acclaim	activ	admir	ador	aesthet
aliv	allure	amaz	amus	appeal
appreci	apt	artfully	artifice	astonish
attract	authent	awe	award	
abruptly	absurd	adolesc	ambigu	annoy
arrog	awkward	arti		

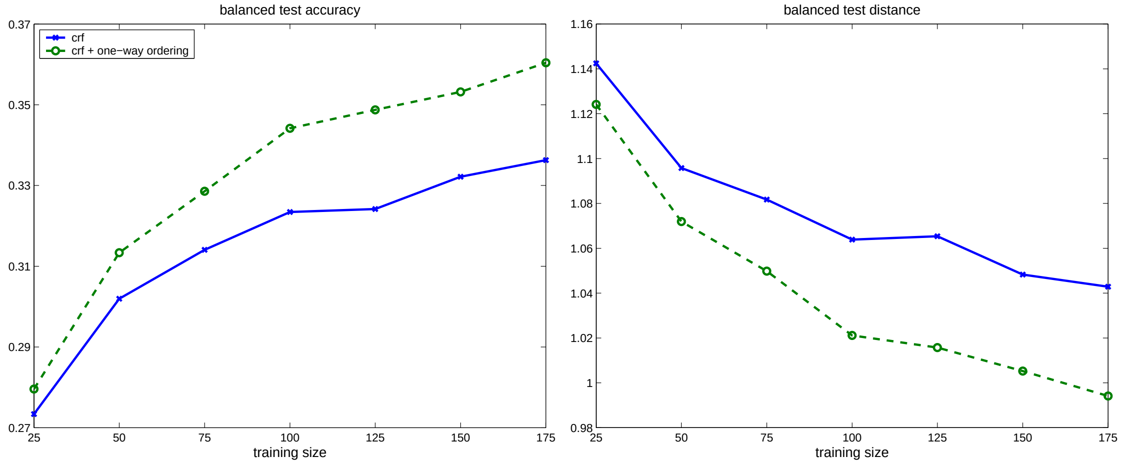


Figure 8: Balanced test-set accuracy (left) and distance of predicted sentiment from true sentiment (right) as a function of training size average over 40 cross validations. One-way ordering constraints are elicited from a domain expert without the use of auxiliary data set.

higher accuracy. This demonstrates the flexibility of our framework in the sense that domain knowledge may come from multiple sources, including domain experts and auxiliary data sets.

3.6 Information Extraction

The idea of generalized isotonic CRF can also be applied to information extraction in natural language processing. In contrast to the case of local sentiment prediction, the set of labels $\mathcal{Y}$ in information extraction is categorical and there is no natural order on it. The sequences $\mathbf{x}$ corresponds to a sentence or a document with x_i being vocabulary words.

We use a CRF model (10) with a set of features $A = \{A_v(\mathbf{x}, i) = 1_{\{x_i = v\}}\}$ that measure the appearance of word v at the current position. We consider isotonic constraints that define a partial order on the $\mu_{\langle \sigma, A_v \rangle}$ as follows. For each word v, we determine the most likely tag $\sigma \in \mathcal{Y}$ and if deemed significant we enforce

$$\mu_{\langle \sigma, A_v \rangle} \geq \mu_{\langle \tau, A_v \rangle} \quad \forall \tau \in \mathcal{Y}, \tau \neq \sigma. \tag{32}$$

We conducted our experiments on the advertisements data for apartment rentals[3] which contains 302 documents labeled with 12 fields, including `size`, `rent`, `restrictions`, etc. During each iteration, 100 documents are randomly selected for testing and the remaining documents are used for training. As previously noted, we use limited memory BFGS for L_2 regularized likelihood with the barrier method enforcing constraints.

As before, one of the authors was presented with the vocabulary of the advertisements data, and was asked to pick a subset of words from it which he thought would be indicative of some field. As a part of the elicitation, he was allowed to observe a few labeled documents (≤ 5) from the data set before the actual selection of words. Table 4 lists the picked words and the field column gives the highest ranked label σ for each word v on the right.

We also use features that model the local context, including

$$B = \{B_v^-(\mathbf{x}, i) = 1_{\{x_{i-1} = v\}}, B_v^+(\mathbf{x}, i) = 1_{\{x_{i+1} = v\}}\}$$

which consider words appearing before and after the current position, and

$$C = \{C_{u,v}^-(\mathbf{x}, i) = 1_{\{x_{i-1} = u\}} 1_{\{x_i = v\}}, C_{u,v}^+(\mathbf{x}, i) = 1_{\{x_i = u\}} 1_{\{x_{i+1} = v\}}\}$$

which consider bigrams containing the current word. Table 5 lists a set of bigrams that are deemed indicative of some field. Since each word or bigram appears only once in Table 4 and 5, the constraint set is contradiction free.

[3] Available at: http://nlp.stanford.edu/∼grenager/data/unsupie.tgz

Table 4: Words selected for one-way ordering in generalized isotonic CRF. The label on the left is determined to be the most likely label corresponding to the words on the right. Words between two asterisks, e.g. *EMAIL*, represent tokens that match the given regular expressions. Words with parentheses denote a group of similar words, e.g. image(s) is used to represent both image and images.

Field	Words
contact	*EMAIL* *PHONE* *TIME* today monday tuesday wednesday friday sat saturday sunday weekend(s) am pm appointment visit reply contact email fax tel schedule questions information details interested @
size	ft feet sq sqft
neighborhood	airport restaurant(s) safeway school(s) shop(s) shopping store(s) station(s) theater(s) transit transportation freeway(s) grocery hwy(s) highway(s) expressway near nearby close mall park banks churches bars cafes
rent	*MONEY* term(s) yearly yr lease(s) contract deposit year month
available	immediately available june july aug august
restrictions	smoke smoker(s) smoking pet(s) cat(s) dog(s) preferred
address	ave avenue blvd
features	backyard balcony(-ies) basement dishwasher(s) dryer(s) furniture fridge garage(s) jacuzzi kitchen(s) kitchenette laundry lndry oven(s) parking pool(s) refrig refrigerator(s) sauna(s) sink(s) spa storage stove(s) swimming tub(s) washer(s) lobby
photos	image(s) photo(s) picture(s)
utilities	utility utilities utils electricity pays
roommates	roommate student

Table 5: Bigrams selected for one-way ordering in generalized isotonic CRF. The label on the left is determined to be the most likely label corresponding to the bigrams on the right.

Field	Words
size	single-family *NUMBER*-story *NUMBER*-bedroom(s) one-bedroom one-bath *NUMBER*-bath(s) one-bathroom two-bedroom(s) *NUMBER*-bathroom *NUMBER*-br square-feet sq-feet sq-ft
neighborhood	walking-distance easy-access convenient-to close-to access-to distance-to block(s)-to away-from located-near block(s)-from block(s)-away minutes-to(away,from) mile-from
features	lots-of plenty-of living-room dining-room gas-stove street-parking
contact	open-house set-up stop-by
rent	*NUMBER*-month application-fee security-deposit per-month /-month /-mo a(one,first,last)-month
address	located-at
restrictions	at-least may-be

Table 6: Labeling accuracy and macro-averaged $F_{1.0}$ for various training size N. Models are trained using the set of features A (left) as well as $A \cup B$ (right) subject to one-way ordering induced by Table 4. An asterisk (*) indicates that the difference is not statistically significant according to the paired t test at the 0.05 level.

N	accuracy		$F_{1.0}$		accuracy		$F_{1.0}$	
	CRF	iso-CRF	CRF	iso-CRF	CRF	iso-CRF	CRF	iso-CRF
10	0.5765	**0.5862***	0.2804	**0.3264**	0.5942	**0.6255**	0.3153	**0.3923**
15	0.6265	**0.6578**	0.3479	**0.4002**	0.6294	**0.6614**	0.3703	**0.4503**
20	0.6354	**0.6750**	0.3760	**0.4433**	0.6553	**0.6931**	0.4110	**0.5090**
25	0.6760	**0.6968**	0.4257	**0.4687**	0.6712	**0.7100**	0.4412	**0.5320**
50	0.7062	**0.7491**	0.5064	**0.5734**	0.7187	**0.7409**	0.5226	**0.5818**
75	0.7533	**0.7658**	0.5716	**0.6038**	0.7391	**0.7528**	0.5594	**0.6061**
100	0.7696	**0.7814**	0.5992	**0.6287**	0.7514	**0.7628**	0.5857	**0.6256**
200	0.7910	**0.8012***	0.6348	**0.6691**	0.7810	**0.7859**	0.6294	**0.6540**

Table 7: Labeling accuracy and macro-averaged $F_{1.0}$ for various training size N. Models are trained using the set of features $A \cup B \cup C$ subject to one-way ordering induced by both Table 4 and 5. We omit the results for one-way ordering induced by Table 4 only, which are almost identical to those reported for iso-CRF. An asterisk (*) indicates that the difference is not statistically significant according to the paired t test at the 0.05 level.

	accuracy		$F_{1.0}$	
	CRF	iso-CRF	CRF	iso-CRF
10	0.5760	**0.5902**	0.2745	**0.2954***
15	0.6146	**0.6322**	0.3310	**0.3560**
20	0.6439	**0.6508**	0.3685	**0.3880**
25	0.6610	**0.6883**	0.4043	**0.4495**
50	0.7190	**0.7370**	0.5043	**0.5503**
75	0.7428	**0.7576**	0.5488	**0.5902**
100	0.7615	**0.7727**	0.5796	**0.6122**
200	0.7921	**0.7999**	0.6405	**0.6667**

Table 6 and 7 display the prediction accuracy (which equals micro-averaged $F_{1.0}$) and macro-averaged $F_{1.0}$ for test data subject to one-way ordering induced by Table 4 and 5. The results are averaged over 20 cross-validation iterations. In all cases, generalized isotonic CRFs consistently outperform the CRF.

3.7 Discussion

Regularized maximum likelihood estimation is one of the most popular estimation techniques in statistical learning. A natural way to incorporate domain knowledge into this framework is through the use of an informative or subjective prior. Assuming the prior is uniform over an admissible area the maximum posterior estimate becomes the constrained version of the maximum likelihood.

An informative prior or frequentist constraints are usually specified on the parameter space Θ. Unfortunately, it is highly non-trivial to obtain a statistical interpretation of the informative prior in terms of the underlying probabilities. This is especially true for conditional random fields which is perhaps the most popular model for structured prediction.

We argue that domain knowledge, whether elicited from a domain expert or from auxiliary data, is best specified directly in terms of probability constraints. Such constraints have a clear interpretation in terms of probability of certain events. We define several types of probability constraints that lead directly to simple parameter constraints thereby facilitating their use as a subjective prior in the statistical learning process. Moreover, the probability constraints can be described in terms of simple queries corresponding to the increase of the probability of a label t_j as a result of a local perturbation of the input sequence $\mathbf{x} \mapsto \mathbf{x}'$. The increase in probability is then compared to the increase in probability of another label s_j. Since it incorporates relative judgement corresponding to an ordering of probability ratios, it is more likely to be accurately elicited than specific probability values.

We present a general framework for incorporating several types of constraints into a simple informative prior consisting of partial ordering constraints on the model parameters. The framework applies to a wide range of applications and leads to efficient computational procedure for solving the constrained regularized maximum likelihood. We demonstrate its applicability to the problems of local sentiment analysis and predicting syntactic and morphological tags in natural language processing.

Our experiments indicate that incorporating the constraints leads to a consistent improvement in prediction accuracy over the regularized CRF model which is considered the state-of-the-art for sentiment prediction and information extraction. In our experiments we study both elicitation from a domain expert and from auxiliary data. In the latter case, we develop an effective mechanism for automatically deriving constraints based on hypothesis testing.

The developed framework applies directly to CRF but could be modified to other structured prediction models such as max-margin discriminative networks. With some simple modifications it applies also to other conditional models such as multinomial logistic regression and in general other forms of conditional graphical models.

Ideally, as the number of constraints increases, we expect the performance first increases, then decreases after some point. If we view the procedure of adding constraints as progressively restricting the feasible set of parameters, the reduced parametric set $\theta \setminus N$ usually leads to an improved model before it becomes too restrictive for the problem.

It is interesting to relate the effective number of constraints (defined as the maximum number of constraints added before the performance drops) with the number of features. Assume the ideal case where there is no overfitting, and each feature is useful. The feasible set $\theta \setminus N$ formed from the effective number of constraints in low dimensional space may not be considered restrictive in high dimensional space, which results from adding more features to the original feature space. As a consequence, we expect the effective number of constraints to be higher for the high dimensional case than for the low dimensional one. That is to say, the number of parameters to be considered is a roughly monotonic function of the number of features.

Following the same argument as above, for fixed number of parameter constraints, we expect as the number of features increases, the performance curve also increases, before it flattens (again we assume no overfitting). This makes our framework attractive especially in the case of high dimensional data.

CHAPTER IV

DOMAIN KNOWLEDGE UNCERTAINTY AND PROBABILISTIC PARAMETER CONSTRAINTS

A fundamental difficulty with incorporating domain knowledge is that as it is provided by humans, it often holds with some degree of uncertainty. For example in sentiment prediction, the presence of the word `good` corresponds usually, but not always to a positive opinion. While this difficulty applies to explicitly formulated domain knowledge, it is even more pronounced when the domain knowledge is obtained implicitly by interpreting user feedback. For example in web search, clickthrough data or the time a user spent in a site are usually interpreted as indicating high relevance. This interpretation is correct in many but not all cases.

In this chapter, we propose to explicitly model domain knowledge uncertainty by specifying the probability with which it is expected to hold. Specifically, we consider the case of a hierarchical prior over the parameter space with additional parameter constraints holding with certain probabilities. Thus in the case of $x \sim p(\cdot|\theta)$, $\theta \sim p(\cdot|\alpha)$ we enforce probabilistic parameter constraint $P(\theta \in A) \geq \eta$ where A is a set corresponding to the domain knowledge and η corresponds to the uncertainty or confidence level. We derive an equivalence between the probabilistic constraint $P(\theta \in A) \geq \eta$ and certain hard constraint over the hyperparameters α. Inference can then proceed on the equivalent model using standard techniques such as empirical Bayes or maximum posterior estimate.

Our proposed framework applies to a large class of practical models. We focus on generative and conditional modeling where the parameters are assigned a Dirichlet or Gaussian prior. This includes the popular cases of ridge regression, mixture of Gaussians, regularized logistic regression, naive Bayes and smoothed n-gram estimation. We show that in these cases the framework translates into well defined and computable hyperparameter constraints and discuss computational schemes for performing Bayesian inference.

From a Bayesian perspective, our framework derives a prior consistent with uncertain domain knowledge and thus may be considered a form of prior elicitation. Its practical significance is that it enables the use of a large quantity of somewhat inaccurate knowledge which is otherwise problematic to use.

4.1 Probabilistic Constraints in Hierarchical Bayes

We consider situations in which the model is a hierarchical Bayes model

$$\begin{aligned} z &\sim f(\cdot|\theta) \qquad \theta \in \mathbb{R}^n \\ \theta &\sim g(\cdot|\alpha) \\ \alpha &\sim h(\cdot) \end{aligned} \tag{33}$$

where f, g are distributions parameterized by θ, α and h is a hyperprior for α. Abusing notation slightly, we consider the distribution f to be over $z = x$ in the generative case i.e., $f(x|\theta)$, or a conditional model over $z = y|x$ in a discriminative setting i.e., $f(y|x, \theta)$. Model (33) is fairly standard and contains a wide variety of popular generative and conditional models such as regularized logistic regression, ridge regression and lasso, mixture of Gaussians, etc. In some cases the distribution $h(\alpha)$ is uniform or an uninformative prior. In other cases it is replaced with a fixed value altogether.

We introduce domain knowledge into the model by identifying sets $A_i, i = 1, \ldots, l$ which are expected to contain the parameters $\theta \in A_i$ with some degree of confidence. A simple case is linear constraints

$$A_i = \{\theta : a_i^\top \theta \leq b_i\} \quad a_i \in \mathbb{R}^n, b_i \in \mathbb{R} \tag{34}$$

which despite its simplicity is general enough to account for many practical situations. Some useful special cases that are achievable using (34) are

$$\theta_{\pi(1)} \leq \cdots \leq \theta_{\pi(k)} \tag{35}$$

$$b \leq \theta_i \leq c \tag{36}$$

$$b \leq |\theta_i - \theta_j| \leq c \tag{37}$$

$$b \leq \sum \theta_i \leq c. \tag{38}$$

Equation (35) represents a case where we know some parameters are likely to be larger than others (π is a permutation over n letters and $k < n$). Equation (36) represents a case where we know the parameter values are bounded, for example in logistic regression we might know that some parameters are positive $\theta_i \geq 0$ (contributing to positive class label) and some are negative $\theta_i \leq 0$ (contributing to negative class label). Equation (37) represents knowledge that two parameters are similar in value and Equation (38) determines that the total parameter value is somehow bounded.

The constraints $\theta \in A_i$ are assigned confidence values η_i and incorporated into the model by pairing (33) with

$$\int_{A_i} g(\theta|\alpha)\, d\theta \geq \eta_i \quad i = 1, \ldots, l. \tag{39}$$

It is important to note that the constraints (39) may or may not be satisfied depending on the value of α. If α is a fixed parameter the constrained problem is trivial - either (39) is satisfied or not. In the former case we can proceed with normal Bayesian inference and in the latter case we need to modify either the constraints (39) or the model (33). However, the situation gets more interesting when α is a random variable. In this case, standard Bayesian inference is modified to account for the constraints, effectively introducing the domain knowledge into the modeling process.

Proposition 6. *The model* (33) *subject to the constraints* (39) *is equivalent to the*

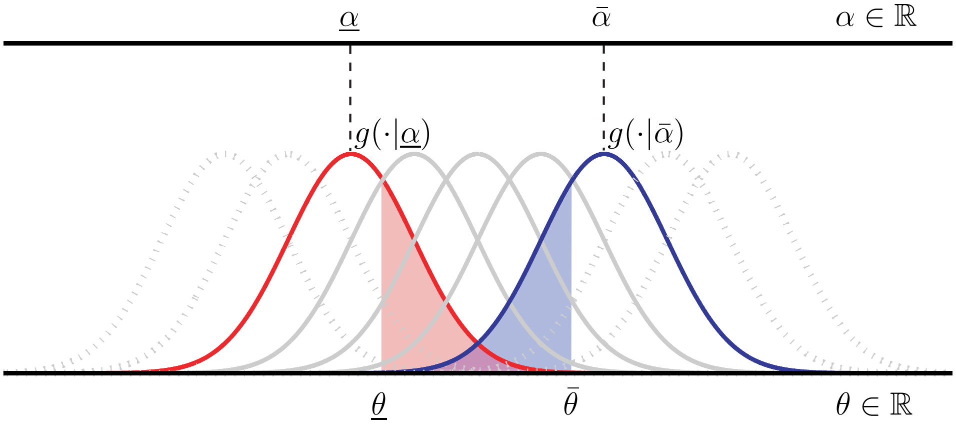


Figure 9: Illustration of proof for Proposition 6. A_i is chosen to be $[\underline{\theta},\ \bar{\theta}]$. For $\alpha \in [\underline{\alpha},\ \bar{\alpha}]$, $\int_{A_i} g(\theta|\alpha)\, d\theta \geq \eta_i$, which implies $B_i = [\underline{\alpha},\ \bar{\alpha}]$. Solid lines represent $g(\cdot|\alpha)$ for $\alpha \in B_i$ while dashed lines represent $\alpha \notin B_i$.

following Bayes model

$$\begin{aligned} z &\sim f(\cdot|\theta) \\ \theta &\sim g(\cdot|\alpha) \qquad (40) \\ \alpha &\sim c\, h(\cdot)\, 1_{\{\alpha \in B_1 \cap \cdots \cap B_l\}} \end{aligned}$$

where c ensures normalization and

$$B_i = \left\{ \alpha : \int_{A_i} g(\theta|\alpha)\, d\theta \geq \eta_i \right\}.$$

Proof. The equivalence follows from considering separately the cases when the constraints are satisfied and when they are not (see Figure 4.1). □

The equivalence derived in Proposition 6 is useful as (40) is an unconstrained Bayesian model on which inference can proceed as usual, assuming the sets $B_1, \ldots, B_l$ are determined and $\cap_i B_i \neq \emptyset$. Specifically, assuming a dataset $\mathcal{D} = \{z^{(1)}, \ldots, z^{(m)}\}$, the full Bayesian treatment suggests integrating over the posterior to obtain expectations of interest. We focus on two alternatives due to computational consideration: empirical Bayes and maximum posterior.

In the case of empirical Bayes (EB) we obtain a point estimate for α by maximizing the posterior $p(\alpha|\mathcal{D})$

$$\begin{aligned} \alpha^* = &\arg\max_{\alpha} \; h(\alpha) \int f(\mathcal{D}|\theta) g(\theta|\alpha) \, d\theta \\ &\text{subject to } \alpha \in B_1 \cap \cdots \cap B_l \end{aligned} \tag{41}$$

and use α^* to compute probabilities of interest. For example, we can classify a new example x by maximizing the predictive distribution implied by the posterior distribution [90] defined as

$$\hat{y} = \arg\max_{y} \int_{\theta} f(y|x,\theta) p(\theta|\mathcal{D},\alpha^*) \, d\theta \tag{42}$$

where $p(\theta|\mathcal{D},\alpha^*)$ is the posterior distribution over θ given by

$$p(\theta|\mathcal{D},\alpha^*) \propto f(\mathcal{D}|\theta) g(\theta|\alpha^*).$$

A second alternative that may be used when the integration (41) is computationally intractable is maximum posterior (MAP) where $p(\alpha,\theta|\mathcal{D})$ is maximized to obtain point estimates for both α, θ

$$\begin{aligned} (\alpha^*, \theta^*) = &\arg\max_{\alpha,\theta} \; f(\mathcal{D}|\theta) g(\theta|\alpha) h(\alpha) \\ &\quad \text{subject to } \alpha \in B_1 \cap \cdots \cap B_l. \end{aligned} \tag{43}$$

In this case new examples may be classified as

$$\hat{y} = \arg\max_{y} f(y|x,\theta^*).$$

In general, it is often hard to invert the constraints (39) and obtain the sets $B_1, \ldots, B_l$ in Proposition 6. In the next two sections we derive the inversion for the case of linear constraints with either a Dirichlet or a Gaussian prior. The maximization problems (41), (43) may be solved using standard interior point optimization.

4.2 Dirichlet Prior

Dirichlet prior $g(\theta|\alpha)$ applies to a variety of models $f(z|\theta)$ whose parameters take values in the simplex

$$\theta \in \mathbb{P}_{n-1} = \left\{\theta \in \mathbb{R}^n : \theta_i \geq 0, \sum \theta_i = 1\right\}. \tag{44}$$

In particular, it is often used in conjunction with a multinomial $f(z|\theta)$ modeling the appearance of words or short phrases called n-grams. The MAP estimate for $f(z|\theta) = \text{Mult}(\theta)$, $g(\theta|\alpha) = \text{Dir}(\alpha)$ modifies the observed word counts by adding α_i to the count of word i in the text and re-normalizing the modified count vector to form a probability distribution. Such models serve a key role in a wide variety of text processing tasks including language modeling, topic analysis, text classification, and syntactic parsing.

Since each dimension in the parameter vector θ corresponds directly to the probability that a certain word or phrase appears, it is easy to construct constraints $\theta \in A_i$ that correspond to linguistic knowledge. In the generative case, such knowledge may correspond to the identification of words that are more popular than others. For example, the following constraint may correspond to plausible linguistic knowledge

$$\theta_i \geq \theta_j \text{ if word } i \text{ is much shorter than word } j. \tag{45}$$

Such a statement may often hold as very long words tend to be uncommon and very short words tend to be common. However, as (45) is not always true it is best to enforce it with some confidence $\eta_i < 1$ in order to prevent poor estimation quality.

As a second example consider the conditional case where different multinomial models with Dirichlet priors are built separately for different class labels y. In this case domain knowledge may reflect the relationship between the class label and the words, in addition to the relationship among the words as in (45). For example, consider the case where the label y corresponds to spam or not spam. It is relatively easy to come up with a list of keywords affiliated with spam emails (`free, information, $`) and constrain the corresponding θ_i to be large if the label y equals spam and small otherwise. Such domain knowledge, while plausible, may not hold always and enforcing it categorically may result in poor estimation quality. On the other hand, enforcing the constraints with confidence $\eta_i < 1$ will allow the model to use the constraints when they apply and avoid them when they do not.

As mentioned in the previous section we focus in this work on linear constraints (34). Such constraints are relatively flexible and they are able to capture ordered and axis aligned constraints (35)-(36) which include the two examples presented above as well as additional special cases such as (37)-(38).

The key to inverting the linear constraints (39) and identifying the sets B_i in the case of a Dirichlet prior is the observation that if $X_j \sim \chi^2_{d_j}, j = 1, \dots, n$ ($\chi^2_{d_j}$ represent independent chi-squared variables with d_j degrees of freedom) then

$$\left(\frac{X_1}{\sum X_i}, \dots, \frac{X_n}{\sum X_i}\right) \sim \text{Dir}\left(\frac{d_1}{2}, \dots, \frac{d_n}{2}\right).$$

It follows that if $\theta \sim \text{Dir}\left(\alpha_1, \dots, \alpha_n\right)$, we may construct independent random variables $Y_j \sim \chi^2_{2\alpha_j}$ so that

$$P\left(\sum_{j=1}^{n} a_j\theta_j \le b\right) = P\left(\frac{\sum_j a_j Y_j}{\sum_j Y_j} \le b\right) = P\left(\sum_j \left(a_j - b\right) Y_j \le 0\right). \tag{46}$$

If $\lambda_1, \dots, \lambda_u$ are u distinct non-zero values of $a_j - b$, $j = 1, \dots, n$, and $T_k \sim \chi^2_{r_k}$ with $r_k \overset{\text{def}}{=} 2\sum_j \alpha_j \delta(a_j - b, \lambda_k)$, $k = 1, \dots, u$, (46) becomes equivalent to

$$P\left(\sum_{k=1}^{u} \lambda_k T_k \le 0\right) = \frac{1}{2} - \frac{1}{\pi}\int_0^\infty \frac{\sin\left(\frac{1}{2}\sum_{k=1}^{u} r_k \tan^{-1}\left(\lambda_k t\right)\right)}{t\prod_{k=1}^{u}\left(1+\lambda_k^2 t^2\right)^{r_k/4}} dt \tag{47}$$

which is a function of $r_1, \dots, r_u$ and thus of α [84].

Solving (47) is a difficult problem since it involves integration over a complex expression of r_k which in turn depend on α. We suggest to use the Edgeworth expansion to approximate (47). The Edgeworth expansion states that if X is a random variable with finite moments, mean zero and variance one, then its density function f can be approximated as either (48) or (49)

$$\frac{f(x)}{\phi(x)} \approx 1 + H_3(x)\frac{\kappa_3}{6} \tag{48}$$

$$\frac{f(x)}{\phi(x)} \approx 1 + H_3(x)\frac{\kappa_3}{6} + H_4(x)\frac{\kappa_4}{24} + H_6(x)\frac{\kappa_3^2}{72}. \tag{49}$$

Above, κ_j is the j-th order cumulant, $\phi(x)$ is the pdf of a standard normal distribution, and H_k are the Hermite polynomials defined as

$$\begin{aligned} H_3(x) &= x^3 - 3x \\ H_4(x) &= x^4 - 6x^2 + 3 \\ H_6(x) &= x^6 - 15x^4 + 45x^2 - 15. \end{aligned}$$

Note for arbitrary random variable Y, we can always define X to be $\frac{Y-E[Y]}{\sqrt{Var[Y]}}$ so that Edgeworth expansion can be applied. See Appendix A or [23] for more details on the Edgeworth expansion.

The first four cumulants for the random variable $\sum_{k=1}^{u} \lambda_k T_k$ in (47) can be computed rather easily. Since $T_k \sim \chi^2_{r_k}$, $k = 1, \ldots, u$ we have

$$\begin{aligned} \kappa_1 &= \mathrm{E}\left[\sum_{k=1}^{u} \lambda_k T_k\right] = \sum_{k=1}^{u} \lambda_k r_k \\ \kappa_2 &= \mathrm{Var}\left[\sum_{k=1}^{u} \lambda_k T_k\right] = \sum_{k=1}^{u} \lambda_k^2 \mathrm{Var}\left[T_k\right] = \sum_{k=1}^{u} 2\lambda_k^2 r_k \\ \kappa_3 &= \sum_{k=1}^{u} 8\lambda_k^3 r_k \\ \kappa_4 &= \sum_{k=1}^{u} 48\lambda_k^4 r_k. \end{aligned}$$

The use of the approximation (48) leads to the following inversion of the probabilistic constraint $P(a^\top \theta \leq b) \geq \eta$

$$B = \left\{\alpha : \Phi\left(\frac{-\kappa_1}{\sqrt{\kappa_2}}\right) - \frac{\kappa_3}{6} H_2\left(\frac{-\kappa_1}{\sqrt{\kappa_2}}\right) \phi\left(\frac{-\kappa_1}{\sqrt{\kappa_2}}\right) \geq \eta\right\}$$

where Φ is the cumulative density function (cdf) of a standard normal distribution. The derivation follows from the fact that $\phi^{(n)}(x) = (-1)^n H_n(x)\phi(x)$.

In theory, function (47) can be approximated to arbitrary precision by using higher order cumulants in the Edgeworth expansion. For random variable $\sum_{k=1}^{u} \lambda_k T_k$ in (47), its higher order cumulants have simple forms which again depend on $r_1, \ldots, r_u$. This implies that the set B can be approximated arbitrarily closely at very little computational cost. In practice, approximations such as (48) and (49) that use only the first four cumulants are often considered adequate and usually work well.

4.3 Gaussian Prior

The most popular prior for continuous unbounded parameters $\theta \in \mathbb{R}^n$ is the Gaussian distribution. It is often used in conjunction with a Gaussian model $f(z|\theta) = \mathcal{N}(\theta, \Upsilon)$, $g(\theta|\mu, \Sigma) = \mathcal{N}(\mu, \Sigma)$ where the posterior $p(\theta|D)$ is Gaussian as well. In this case the posterior and various integrals over it have a close form.

In the conditional or discriminative setting, a Gaussian prior is often used in conjunction with linear regression

$$f(y|x, \theta) = \mathcal{N}(\theta^\top x, \sigma^2) \quad y \in \mathbb{R} \tag{50}$$

or logistic regression

$$f(y|x, \theta) = \left(1 + e^{-y\theta^\top x}\right)^{-1} \quad y \in \{-1, +1\}. \tag{51}$$

In both cases (50)-(51) a Gaussian prior over θ is the most popular means of introducing domain knowledge or regularizing the model.

Specifying domain knowledge by constraining θ is relatively easy as $\theta_1, \ldots, \theta_n$ correspond directly to the expected values of the data dimensions $z_1, \ldots, z_n$. For example, consider modeling a physical population quantity using a mixture of Gaussians. There may be reasons to believe that some mixture components correspond to specific groups in the population, enabling the use of domain knowledge to constrain the parameters of the mixture components. If the constraints are uncertain, introducing probabilistic rather than hard constraints will be more robust in the event of their failure.

In the conditional case, constraints on θ may reflect the relationship among the data and the predictor variable y. For example in a logistic regression model for classifying document topics, we may enforce $|\theta_i| \leq c$ for some i corresponding to stop-words or non-content words. The assumption that non-content words such as `the` or `of` do not contribute to the topic is a reasonable one. However, there are cases in which the constraints may not hold which motivate $\eta < 1$.

We turn now to inverting the constraints (34) and identifying the sets $B_1, \ldots, B_n$ if $\theta \sim \mathcal{N}(\mu, \Sigma)$. We have $u \overset{\text{def}}{=} a_i^\top \theta \sim \mathcal{N}(\bar{u}, \sigma^2)$ where $\bar{u} = a_i^\top \mu$, $\sigma^2 = a_i^\top \Sigma a_i$ and

$$\begin{aligned} P\left(a_i^\top \theta \leq b_i\right) \geq \eta_i &\Leftrightarrow P\left(\frac{u-\bar{u}}{\sigma} \leq \frac{b_i - \bar{u}}{\sigma}\right) \geq \eta_i \\ &\Leftrightarrow \frac{b_i - \bar{u}}{\sigma} \geq \Phi^{-1}(\eta_i) \\ &\Leftrightarrow a_i^\top \mu + \Phi^{-1}(\eta_i)\sqrt{a_i^\top \Sigma a_i} \leq b_i \end{aligned} \tag{52}$$

(Φ is the standard normal cdf). Further details concerning this derivation may be found in [8].

Depending on the problem structure, we may assume the hyperparameter α to be (μ, Σ) or just μ (Σ is considered fixed in this case). One difficulty is that the MAP or EB optimization problem is specified in terms of Σ^{-1} while the inverted constraints (52) are specified in terms of Σ. This difficulty is not substantial if Σ is diagonal as $\Sigma^{-1} = \text{diag}(1/\Sigma_{11}, \ldots, 1/\Sigma_{nn})$.

In situations when Σ is not a diagonal matrix obtaining the EB or MAP estimator subject to the inverted constraints is highly non-trivial from an optimization perspective. For the problem of obtaining the MAP estimator, we propose instead to optimize a surrogate objective function based on the method of Bregman divergences.

We make a standard assumption regarding the hyperprior $h(\alpha)$

$$h\left(\mu, \Sigma^{-1}\right) \propto \exp\left(-\frac{1}{2}\mathrm{tr}\left(\Sigma^{-1}\Lambda\right)\right) \tag{53}$$

(Λ is a positive definite matrix) which is equivalent to stating that $\Sigma^{-1} \sim \mathrm{Wishart}_{n+1}$ (Λ^{-1}), $\mu|\Sigma^{-1}$ is uniform. Note that the techniques introduced below apply to arbitrary $f(\cdot|\theta)$ as no assumptions are made regarding the particular choice of f.

We propose to use an iterative optimization technique, and during the step of optimizing Σ with fixed θ and μ, maximize instead a surrogate objective function based on the method of Bregman projection. Specifically, we solve the following problem to obtain the point estimator for Σ for fixed θ and μ

$$\begin{aligned} \min_{\Sigma} \quad & D_{\mathrm{LogDet}}\Big(\Sigma, \Lambda + (\theta-\mu)(\theta-\mu)^\top\Big) \\ \text{s.t.} \quad & \mathrm{tr}\Big(\Sigma a_i a_i^\top\Big) \le \left(\frac{b_i - a_i^\top\mu}{\Phi^{-1}(\eta_i)}\right)^2, \quad i = 1, \ldots, l. \end{aligned} \tag{54}$$

The divergence above is the LogDet Bregman divergence between matrices [10] (see also Appendix C for a brief introduction of Bregman divergence). The hyperparameter Σ estimated by (54) is then used when we subsequently optimize over θ or μ.

The problem (54) has the same constraints as the original subproblem but a different objective function which is originally $D_{\mathrm{LogDet}}(\Lambda + (\theta-\mu)(\theta-\mu)^\top, \Sigma)$. By switching the arguments of the objective function, we are able to solve the problem using the method of Bregman projections [10] which achieves the optimal solution by sequentially projecting to different convex regions defined by the corresponding constraints.

A useful property of Bregman projection is that it can be used to ensure the positive definiteness of Σ^{-1} and Σ when starting from a positive definite matrix Λ. This results immediately from the fact that each update of Σ^{-1} by projecting the matrix divergence onto the convex region defined by $\mathrm{tr}\Big(\Sigma a_i a_i^\top\Big) \le z_i$ takes the following form

$$\Sigma^{-1} = \Big(\Lambda + (\theta - \mu)(\theta - \mu)^\top\Big)^{-1} + \nu a_i a_i^\top$$

where

$$\nu = \max\left\{0, \frac{a_i^\top(\Lambda + (\theta-\mu)(\theta-\mu)^\top)a_i - z_i}{z_i a_i^\top(\Lambda + (\theta-\mu)(\theta-\mu)^\top)a_i)}\right\} \geq 0.$$

The problem size that can usually be handled in our framework is limited by the problem size that can be solved by the quadratic programming solver. In case the parameter constraints are sparse, i.e. a_i are sparse vectors, a couple of thousand parameters is durable. For extremely large problems, feature selection or dimensionality reduction is required as a processing step.

4.4 Approximation of Empirical Bayes

In this section, we discuss the case of empirical Bayes (41) when $\theta \sim g(\cdot|\alpha)$ is not a conjugate prior for the likelihood function. In particular, we consider binary classification problems modeled by the generalized linear model

$$f(y|x, \theta) = \Phi(y\theta^\top x)$$

where y takes values in $\{0, 1\}$, θ is a vector of regression parameters and $\Phi(\cdot)$ is some link function. Two most commonly used link functions are the logistic function $\Phi(z) = \frac{e^z}{1+e^z}$ and the probit function $\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$. The resulting models are termed logistic regression and probit regression accordingly.

A common practice is to choose a Guassian prior for the regression parameter θ. In this case, the integral $\int_\theta f(\mathcal{D}|\theta) g(\theta|\mu, \Sigma) d\theta$ is analytically intractable, and may be approximated using sampling methods. Alternatively, we consider here techniques based on an analytical approximation. The idea is to approximate the likelihood function as an unnormalized Gaussian so that we can compute the integral analytically. We begin with the following two lemmas which are essential for carrying out the computation.

Lemma 1. *Let* $\mathsf{A} \in \mathbb{R}^{n\times n}$ *be a symmetric positive definite matrix and* $\mathsf{b} \in \mathbb{R}^n$ *is some arbitrary vector, we have*

$$\int \exp\left(-\frac{1}{2}x^\top \mathsf{A}x + x^\top \mathsf{b}\right) d^n x = \sqrt{\frac{(2\pi)^n}{\det \mathsf{A}}} \exp\left(\frac{1}{2}\mathsf{b}^\top \mathsf{A}^{-1}\mathsf{b}\right). \tag{55}$$

Proof.

$$\begin{aligned} & \int \exp\left(-\frac{1}{2}x^\top \mathsf{A}x + x^\top \mathsf{b}\right) d^n x \\ = \; & \frac{(2\pi)^{\frac{n}{2}}}{|\mathsf{A}|^{\frac{1}{2}}} \exp\left(\frac{1}{2}\mathsf{b}^\top \mathsf{A}^{-1}\mathsf{b}\right) \int \frac{1}{(2\pi)^{\frac{n}{2}}|\mathsf{A}^{-1}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mathsf{A}^{-1}\mathsf{b})^\top \mathsf{A}(x-\mathsf{A}^{-1}\mathsf{b})\right) d^n x \\ = \; & \sqrt{\frac{(2\pi)^n}{\det \mathsf{A}}} \exp\left(\frac{1}{2}\mathsf{b}^\top \mathsf{A}^{-1}\mathsf{b}\right) \end{aligned}$$

where the last equality holds since the function inside the integral is the probability density function of a multivariate normal distribution with mean $\mathsf{A}^{-1}\mathsf{b}$ and covariance matrix A^{-1}. □

Lemma 2. *The Woodbury matrix identity (see e.g. [42])*

$$(A + CBC^\top)^{-1} = A^{-1} - A^{-1}C(B^{-1} + C^\top A^{-1}C)^{-1}C^\top A^{-1}. \tag{56}$$

When B is a one-by-one matrix, the Woodbury matrix identity (56) reduces to the Sherman Morrison inverse formula, which states that

$$(A + uv^\top)^{-1} = A^{-1} - \frac{1}{1 + v^\top A^{-1}u}A^{-1}uv^\top A^{-1}.$$

Given a set of data $\mathcal{D} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$, we start by approximating $\log f(\mathcal{D}|\theta)$ as a quadratic function in θ

$$\log f(\mathcal{D}|\theta) \cong -\frac{1}{2}\theta^\top \mathsf{A}\theta + \theta^\top \mathsf{b} + \mathsf{c}$$

where the expressions for A, b and c are given in Appendix D. The integral $\int f(\mathcal{D}|\theta)$

$g(\theta|\mu,\Sigma)d\theta$ is then computed to be

$$\begin{aligned}
& \int f(\mathcal{D}|\theta) g(\theta|\mu,\Sigma) d\theta \\
= \; & \int \exp\left(-\frac{1}{2}\theta^\top \mathsf{A}\theta + \theta^\top \mathsf{b} + \mathsf{c}\right) \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\theta-\mu)^\top\Sigma^{-1}(\theta-\mu)\right) d\theta \\
= \; & \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(\mathsf{c} - \frac{1}{2}\mu^\top\Sigma^{-1}\mu\right) \int \exp\left(-\frac{1}{2}\theta^\top\left(\mathsf{A}+\Sigma^{-1}\right)\theta + \theta^\top\left(\mathsf{b}+\Sigma^{-1}\mu\right)\right) d\theta \\
= \; & \frac{1}{|I+\Sigma\mathsf{A}|^{\frac{1}{2}}} \exp\left(\mathsf{c} - \frac{1}{2}\mu^\top\Sigma^{-1}\mu + \frac{1}{2}\left(\mathsf{b}+\Sigma^{-1}\mu\right)^\top\left(\mathsf{A}+\Sigma^{-1}\right)^{-1}\left(\mathsf{b}+\Sigma^{-1}\mu\right)\right)
\end{aligned}$$

where the last equality is obtained by applying Lemma 1.

We make a standard assumption regarding the hyperprior $h(\mu,\Sigma)$

$$h(\mu,\Sigma) \propto |\Sigma|^{\frac{p}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\Sigma\Lambda_0\right)\right)$$

for $p \geq 1$ which is equivalent to stating that Σ is distributed as $\text{Wishart}_{n+1+p}(\Lambda_0^{-1})$ [44] for some positive definite matrix Λ_0 and $\mu|\Sigma$ is uniform. The objective function of (41) now becomes

$$\left(\mathsf{b}+\Sigma^{-1}\mu\right)^\top\left(\mathsf{A}+\Sigma^{-1}\right)^{-1}\left(\mathsf{b}+\Sigma^{-1}\mu\right) - \mu^\top\Sigma^{-1}\mu - \log|I+\Sigma\mathsf{A}| - \mathrm{tr}\left(\Sigma\Lambda_0\right) + p\log|\Sigma|.$$

To solve for optimal hyperparameters, we iteratively optimize over μ and Σ. When Σ is fixed, the problem (41) reduces to a quadratic programming problem

$$\begin{aligned}
\min_{\mu} \quad & \frac{1}{2}\mu^\top\mathsf{Q}\mu + \mathsf{q}^\top\mu \\
\text{s.t.} \quad & a_i^\top\mu \leq b_i - \Phi^{-1}(\eta_i)\sqrt{a_i^\top\Sigma a_i} \qquad i = 1,\ldots,l
\end{aligned}$$

where

$$\begin{aligned}
& \mathsf{Q} = \Sigma^{-1} - (\Sigma\mathsf{A}\Sigma+\Sigma)^{-1} \overset{(a)}{=} \Sigma^{-1} - \left(\Sigma^{-1} - \left(\mathsf{A}^{-1}+\Sigma\right)^{-1}\right) = (\mathsf{A}^{-1}+\Sigma)^{-1} \\
& \mathsf{q} = -(\mathsf{A}\Sigma + I)^{-1}\mathsf{b}.
\end{aligned}$$

Step (a) holds by Lemma 2.

To optimize Σ when μ is fixed, let $\mathsf{X} = \mathsf{A}+\Sigma^{-1}$, $\mathsf{y} = \mathsf{b} - \mathsf{A}\mu$, we have

$$\Sigma^{-1}\mu + \mathsf{b} = \left(\mathsf{X}-\mathsf{A}\right)\mu + \mathsf{b} = \mathsf{X}\mu + \mathsf{y}.$$

Since A is symmetric and Σ is assumed to be symmetric positive definite, matrix X is therefore symmetric, yielding

$$\begin{aligned}
& \left(\mathsf{b}+\Sigma^{-1}\mu\right)^\top\left(\mathsf{A}+\Sigma^{-1}\right)^{-1}\left(\mathsf{b}+\Sigma^{-1}\mu\right) \\
= \quad & \left(\mathsf{X}\mu+\mathsf{y}\right)^\top \mathsf{X}^{-1}\left(\mathsf{X}\mu+\mathsf{y}\right) \\
= \quad & \mu^\top\mathsf{X}\mathsf{X}^{-1}\mathsf{X}\mu+\mathsf{y}^\top\mathsf{X}^{-1}\mathsf{y}+\mathsf{y}^\top\mathsf{X}^{-1}\mathsf{X}\mu+\mu^\top\mathsf{X}\mathsf{X}^{-1}\mathsf{y} \\
= \quad & \mu^\top\left(\mathsf{A}+\Sigma^{-1}\right)\mu+\mathsf{y}^\top\mathsf{X}^{-1}\mathsf{y}+2\mu^\top\mathsf{y} \\
= \quad & \mu^\top\Sigma^{-1}\mu+\mathsf{y}^\top\mathsf{X}^{-1}\mathsf{y}+\text{const.}
\end{aligned}$$

Without loss of generality, matrix A is assumed to be symmetric positive definite. Let $UDU^\top$ be the singular value decomposition of A, we define $\mathsf{A}^{\frac{1}{2}}$ to be $UD^{\frac{1}{2}}U^\top$ where $D^{\frac{1}{2}}$ is computed by taking square root of D element-wise, and denote $\left(\mathsf{A}^{\frac{1}{2}}\right)^{-1}$ as $\mathsf{A}^{-\frac{1}{2}}$. Let $\Omega=\mathsf{A}^{\frac{1}{2}}\Sigma\mathsf{A}^{\frac{1}{2}}$, $\Upsilon=I+\Omega$, $\Lambda=\mathsf{A}^{-\frac{1}{2}}\Lambda_0\mathsf{A}^{-\frac{1}{2}}$, $\mathsf{z}=\mathsf{A}^{-\frac{1}{2}}\mathsf{b}-\mathsf{A}^{\frac{1}{2}}\mu=\mathsf{A}^{-\frac{1}{2}}\mathsf{y}$, the objective function becomes

$$\begin{aligned}
& \mathsf{y}^\top(\mathsf{A}+\Sigma^{-1})^{-1}\mathsf{y}-\log|I+\Sigma\mathsf{A}|-\operatorname{tr}(\Sigma\Lambda_0)+p\log|\Sigma| \\
\overset{(a)}{=} \quad & \mathsf{y}^\top(\mathsf{A}+\mathsf{A}^{\frac{1}{2}}\Omega^{-1}\mathsf{A}^{\frac{1}{2}})^{-1}\mathsf{y}-\log|I+\mathsf{A}^{-\frac{1}{2}}\Omega\mathsf{A}^{-\frac{1}{2}}\mathsf{A}|-\operatorname{tr}\left(\mathsf{A}^{-\frac{1}{2}}\Omega\mathsf{A}^{-\frac{1}{2}}\Lambda_0\right) \\
& +p\log|\mathsf{A}^{-\frac{1}{2}}\Omega\mathsf{A}^{-\frac{1}{2}}| \\
= \quad & \left(\mathsf{A}^{-\frac{1}{2}}\mathsf{y}\right)^\top(I+\Omega^{-1})^{-1}\left(\mathsf{A}^{-\frac{1}{2}}\mathsf{y}\right)-\log|\mathsf{A}^{-\frac{1}{2}}||I+\Omega||\mathsf{A}^{\frac{1}{2}}|-\operatorname{tr}\left(\Omega\mathsf{A}^{-\frac{1}{2}}\Lambda_0\mathsf{A}^{-\frac{1}{2}}\right) \\
& +p\log|\Omega|+\text{const} \\
= \quad & \mathsf{z}^\top(I+\Omega^{-1})^{-1}\mathsf{z}-\log|I+\Omega|-\operatorname{tr}(\Omega\Lambda)+p\log|\Omega|+\text{const} \\
\overset{(b)}{=} \quad & \mathsf{z}^\top(I-\Upsilon^{-1})\mathsf{z}-\log|\Upsilon|-\operatorname{tr}(\Upsilon\Lambda)+p\log|\Upsilon-I|+\text{const} \\
= \quad & -\mathsf{z}^\top\Upsilon^{-1}\mathsf{z}-\log|\Upsilon|-\operatorname{tr}(\Upsilon\Lambda)+p\log|\Upsilon-I|+\text{const} \\
= \quad & -J(\Upsilon)
\end{aligned}$$

where (a) is based on the fact that $\Sigma=\mathsf{A}^{-\frac{1}{2}}\Omega\mathsf{A}^{-\frac{1}{2}}$ and $\Sigma^{-1}=\mathsf{A}^{\frac{1}{2}}\Omega^{-1}\mathsf{A}^{\frac{1}{2}}$, and (b) holds since

$$(I+\Omega^{-1})^{-1}=(I+(\Upsilon-I)^{-1})^{-1}=(\Upsilon-I)(\Upsilon-I+I)^{-1}=I-\Upsilon^{-1}.$$

Instead of minimizing $J(\Upsilon)$ directly, we choose to minimize an upper bound of $J(\Upsilon)$ which substantially simplifies the optimization problem. Let λ_i and $\mathbf{u}_i$ be the set of eigenvalues and eigenvectors of Υ, i.e. $\Upsilon = \sum_i \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$. Since $\Upsilon \succ I$ by assumption, we have $\lambda_i > 1\ \forall i$, and consequently

$$\mathsf{z}^\top \Upsilon^{-1} \mathsf{z} = \sum_i \frac{1}{\lambda_i} \left(\mathsf{u}_i^\top \mathsf{z}\right)^2 \leq \sum_i \lambda_i \left(\mathsf{u}_i^\top \mathsf{z}\right)^2 = \mathsf{z}^\top \Upsilon \mathsf{z}$$

and

$$\log|\Upsilon| = \log \prod_i \lambda_i = \sum_i \log \lambda_i \leq \sum_i (\lambda_i - 1) = \mathrm{tr}(\Upsilon - I).$$

This leads to the following upper bound on $J(\Upsilon)$

$$\begin{aligned} J(\Upsilon) &\leq \mathsf{z}^\top \Upsilon \mathsf{z} + \mathrm{tr}(\Upsilon - I) + \mathrm{tr}\left(\Upsilon \Lambda\right) - p \log|\Upsilon - I| + \mathrm{const} \\ &= p\left(\frac{1}{p}\mathrm{tr}\left((\Upsilon - I)\left(I + \Lambda + \mathsf{z}\mathsf{z}^\top\right)\right) - \log\left|(\Upsilon - I)\left(I + \Lambda + \mathsf{z}\mathsf{z}^\top\right)\right|\right) + \mathrm{const} \end{aligned}$$

which is essentially proportional to the LogDet divergence between matrices $\Upsilon - I$ and $p(I + \Lambda + \mathsf{z}\mathsf{z}^\top)^{-1}$.

Since the constraints are simplified to be

$$C_i = \left\{ \Upsilon : \mathsf{v}_i^\top (\Upsilon - I) \mathsf{v}_i \leq \left(\frac{b_i - a_i^\top \mu}{\Phi^{-1}(\eta_i)} \right)^2 \right\}, \quad i = 1, \ldots, l$$

where $\mathsf{v}_i = \mathsf{A}^{-\frac{1}{2}} a_i$, we can solve for optimal Υ (therefore Σ) by sequential projections onto each convex set C_i. As with the case of Gaussian prior before, each projection can be computed efficiently and $\Upsilon - I$ is guaranteed to be positive definite at every step.

To classify a new example x by (42), we note that it has the same form as the Bayesian logistic regression or the Bayesian probit regression, but with the prior being defined by the learned optimal hyperparameters. Following Chapter 4.5 of [6], if we apply the Laplace approximation to obtain a Gaussian representation for the posterior distribution $p(\theta|\mathcal{D}, \alpha^*)$, the decision boundary obtained will be the same as the MAP value $\theta^* = \arg\max_\theta p(\theta|\mathcal{D}, \alpha^*)$.

4.5 Experiments

We demonstrate our framework using experiments on synthetic and real-world data. The synthetic data experiments test the applicability of the framework to the multinomial, Gaussian, and linear regression cases. The real world experiments test the applicability of the framework to two NLP tasks: sentiment prediction and readability prediction, both using linear regression.

4.5.1 Synthetic Data Experiments

We start by evaluating the framework on the problem of estimating multinomial parameters under ordering constraints. We sampled data from Mult(θ) for $\theta = \left(\frac{1}{12}, \frac{1}{6}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3}\right)$ and enforced the probabilistic constraints $A = \{\theta_i \leq \theta_j,\ i = 1,2,3$ and $j = 4,5\}$ (Figure 10, top left) and $B = \{\theta_i \geq \theta_j,\ i = 1,2,3$ and $j = 4,5\}$ (Figure 10, top right). We used in this and other experiments (unless noted otherwise) a confidence value of $\eta_i = 0.95$. We assumed a Dirichlet prior for $\theta \sim \text{Dir}(\alpha)$, and a uniform hyperprior for α.

In the Gaussian case we generated data from three normal distributions $\mathcal{N}(\theta_1, 1)$, $\mathcal{N}(\theta_2, 1), \mathcal{N}(\theta_3, 1)$ for $\theta = (\theta_1, \theta_2, \theta_3) = (0, 1/2, 1)$ and enforced the probabilistic constraints $C = \{\theta_1 \leq \theta_2, \theta_2 \leq \theta_3, \theta_1 \geq 0, \theta_3 \leq 1\}$ (Figure 10, middle left) and $D = \{\theta_1 \geq \theta_2, \theta_2 \geq \theta_3\}$ (Figure 10, middle right).

In the case of linear regression, the samples were drawn from the model $y \sim \mathcal{N}(\beta^\top x, 1)$ where β is a 10 dimensional randomly generated vector whose first and last 5 components are uniformly distributed on $(-1, 0)$ and $(0, 1)$ respectively. We enforced the probabilistic constraints $E = \{\{\beta_1, \beta_3, \beta_5\} \leq 0, \{\beta_6, \beta_8, \beta_{10}\} \geq 0, \{\beta_2, \beta_4\} \leq \{\beta_7, \beta_9\}\}$ (Figure 10, bottom left) and $F = \{\{\beta_1, \beta_3, \beta_5\} \geq 0, \{\beta_6, \beta_8, \beta_{10}\} \leq 0, \{\beta_2, \beta_4\} \geq \{\beta_7, \beta_9\}\}$ (Figure 10, bottom right). In this case we applied ridge regularization to both the MLE and the constrained MLE and report the best results from the following set of ridge parameters $\{0.001, 0.01, 0.1, 0.2, 0.5, 1, 2, 5, 10, 100\}$.

For logistic regression, the samples were drawn from the following distributions $p(x|y = 1) \sim \mathcal{N}(\beta_+, I)$ and $p(x|y = -1) \sim \mathcal{N}(\beta_-, I)$ where β_+ is a 10 dimensional vector whose components are set to 0.3, and $\beta_- = -\beta_+$. The hyperplane that separates the positive examples from the negative examples is almost perpendicular to the line connecting β_+ and β_-. For regression parameters w, we enforced the probabilistic constraints $E = \{w_i \geq 0, \forall\, i\}$ (Figure 11, top) and $F = \{w_i \leq 0, \forall\, i\}$ (Figure 11, bottom). As in the case of linear regression, we applied L_2 regularization to both the MLE and the constrained MLE and report the best results from the regularization parameter set $\{0.001, 0.01, 0.1, 0.2, 0.5, 1, 2, 5, 10, 100\}$.

In all four cases we observe similar results. When the constraints are correct incorporating them via constrained MLE (hard constraints) or MAP, EB (probabilistic constraints) provides higher estimation accuracy over the non-constrained MLE.

However, when some of the constraints are inaccurate, incorporating them as hard constraints hurts performance substantially and results in much poorer estimation as compared to the unconstrained MLE. This is to be expected as hard inaccurate constraints force the estimator away from the true parameters. On the other hand, incorporating inaccurate probabilistic constraints using MAP or EB performs remarkably well with almost equal performance to the unconstrained MLE. The inaccurate constraints don't hurt the estimator as the constraints are simply ignored due to their clash with the information embedded in the data. Note that this holds even for high confidence values such as $\eta = 0.95$ (our choice for these experiments).

4.5.2 Sentiment and Readability Prediction

To test the validity of the framework on real world data we experimented with two NLP tasks: sentiment and readability prediction where the underlying model is linear regression.

For sentiment prediction, we randomly chose 2 out of 4 movie critics from the

Cornell sentiment scale datasets, which results in collections of 1027 and 1307 documents respectively, with 4 sentiment levels ranging from 1 (very bad) to 4 (very good). For readability prediction, we used the weekly reader dataset, obtained by crawling the Weekly Reader[1] commercial website after receiving special permission. The readability dataset contains a total of 1780 documents, with 4 readability levels ranging from 2 to 5 indicating the school grade levels of the intended audience. Preprocessing includes lower-casing, stop word removal, stemming, and selecting 1000 top features based on document frequency. The predictor variable is also centered for ease of applying parameter constraints.

The probabilistic constraints for the sentiment prediction experiment were developed by one of the authors after being presented with the vocabulary of the dataset. The author was asked to pick two subsets of the vocabulary - one associated with positive sentiment and one with negative sentiment. A total of 190 and 154 words were chosen for the two critics. A subset of these words starting with 'a' are listed in Table 8. For words that are deemed indicative of positive sentiment, we enforce $\theta_i \geq b$ for some nonnegative number b as the parameter constraints. Similarly, we enforce $\theta_i \leq -b$ for the negative words.

In the case of readability prediction, we assume that the appearance of longer words implies higher readability level than the appearance of shorter words. To this end, we randomly chose 600 pairs of words of different length, and required that the parameters corresponding to the longer words have a higher value than the parameters corresponding to the shorter words. Note that in both the sentiment and readability cases the constraints represent reasonable domain knowledge but may not be entirely accurate.

Figure 12 compares ridge regression, constrained ridge regression and MAP with a full covariance matrix. We chose to use a full rather than a diagonal matrix due to

[1] www.wrtoolkit.com

Table 8: Words chosen for parameter constraints for sentiment prediction. Superscript numbers indicate the movie critic. Italics blue words indicate positive sentiment while non-italics black words indicate negative sentiment.

appeal^{1,2}	*award*^{1,2}	*accomplish*^{1,2}	*attract*^{1,2}
amus^{1,2}	annoi^{1,2}	*appar*^{1}	avoid^{1}
adequ^{1}	*amaz*^{1}	aw^{1}	*appreci*^{1}
awkward^{2}	absurd^{2}	*achiev*^{2}	artifici^{2}
art^{2}	arti^{2}	*admir*^{2}	

the correlation between the regression parameters. The ridge parameter was chosen from the set $\{0.2, 0.5, 1, 2, 5\}$. Variance σ^2 of the linear regression model (50) is assumed to lie in $\{0.5, 1, 2\}$ and Λ in (53) takes the form of τI where τ is chosen from $\{0.01, 0.05, 0.1, 0.2, 0.3\}$. The parameter value bound b in sentiment prediction is set to be 0.

The results shown in Figure 12 illustrate that the probabilistic constraints help improve accuracy over the unconstrained MLE. More impressive is the fact that they result in a substantial improvement in modeling accuracy also over the (hard) constrained MLE. This is due to the uncertain nature of the constraints and the fact that some of the constraints do not hold. This underscores the main point of the chapter that domain knowledge is often uncertain and better enforced using probabilistic parameter constraints rather than hard ones.

It is worth mentioning that the framework is not sensitive to the choice of η for a broad region of possible η. For sentiment prediction, we have experimented with η being equal to 0.75, 0.85 and 0.95, and parameter value bound b being equal to 0, 0.1 and 0.5. For all those combinations of parameters, we found that the graphs are quite similar to Figure 12 except for individual values of likelihood or accuracy. This is indeed a desirable property for real-world applications when the confident level expressed by a domain expert may be subject to uncertainty.

4.6 Discussion

Incorporating knowledge into the learning process has been studied extensively by the statistics community. Frequentists use it to define the model and constrain the parameter space. Bayesians use it to define the model and the prior over the parameter space. In the Bayesian case, uncertainty is usually handled by using hierarchical models with diffuse hyperpriors [4]. Obtaining domain knowledge is addressed by prior elicitation in the subjective Bayes community [41].

Our work differs from the standard prior elicitation approach in that we do not elicit the prior directly. Rather we elicit parameter constraints and confidence values which are used in turn to derive an equivalent prior in a hierarchical Bayes setting via Proposition 6. Standard Bayesian inference can then proceed on the equivalent model in the usual manner.

The advantage of doing so is that it is much easier for domain experts to specify constraints and confidence values. Directly specifying a prior is considerably less intuitive as it makes it hard to discern the confidence with which specific assertions are made. Thus, our contribution is in nicely separating the domain assertions and their confidence values in a simple and intuitive way.

Using experiments on synthetic and real world data we show that uncertain domain knowledge can be effectively incorporated in practice. The use of uncertain constraints leads to high modeling accuracy when the constraints are accurate. In case the constraints are inaccurate, the uncertainty prevents the model from performing poorly which stands in contrast to hard constraints that push the parameters away from their true values.

Specifying domain knowledge in real world situations is sometimes bound to be inaccurate. Our approach enables the use of a large number of domain statements without worrying too much about the validity of each specific statement. Our experiments indicate that it works well for natural language problems where domain knowledge is relatively easy to specify. It is likely that the framework performs similarly well in other areas where domain knowledge is available and the underlying model has a Dirichlet or Gaussian prior.

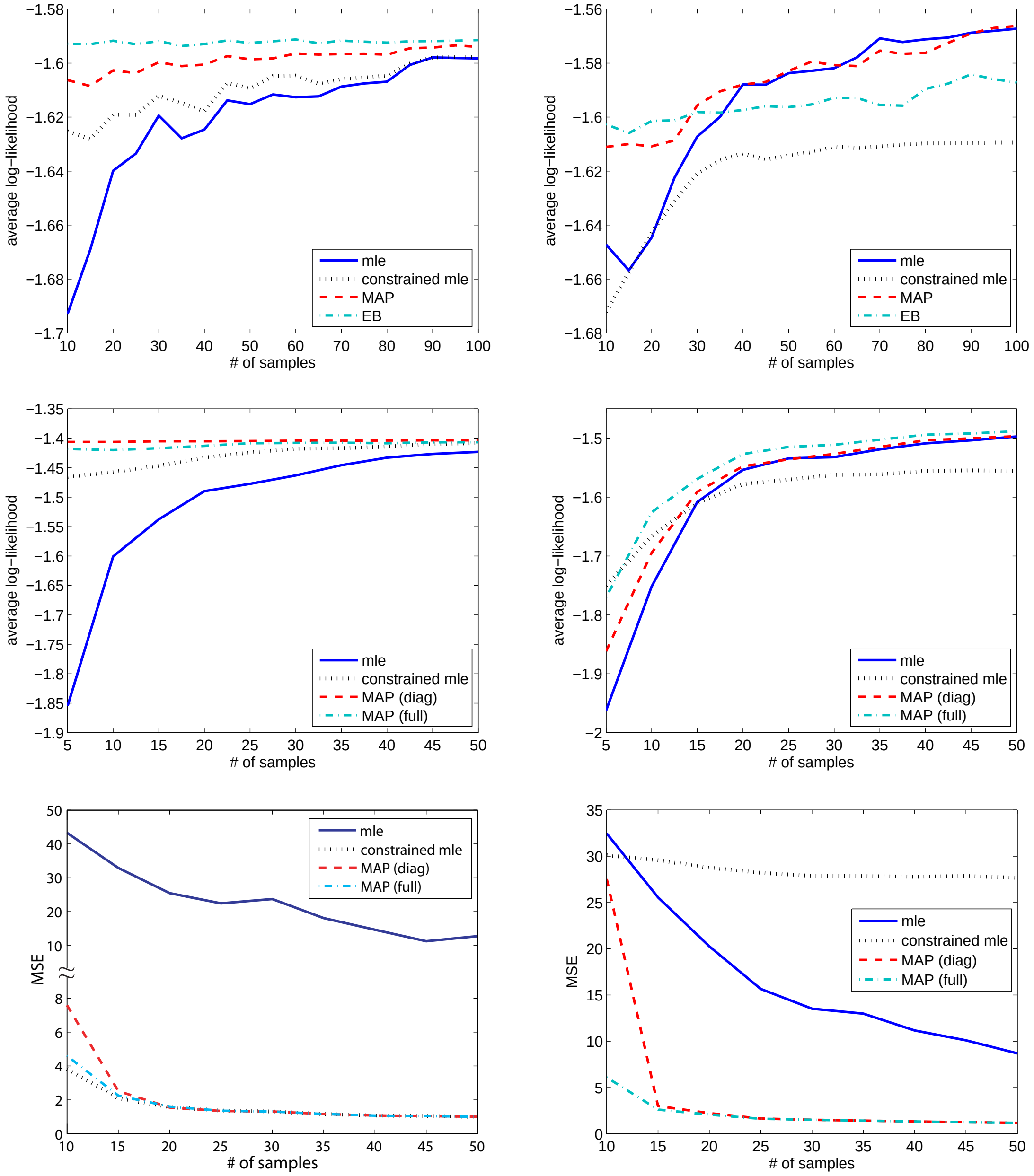


Figure 10: Average test set performance for multinomial (top), Gaussian (middle) and linear regression (bottom) over 20 random train/test splits. Multinomial parameters are estimated by MLE, (hard) constrained MLE, and probabilistic constraint EB and MAP. Gaussian means and regression parameters are estimated by MLE, (hard) constrained MLE and probabilistic constraint MAP with either diagonal or full covariance matrix. Ridge regularization is applied to both MLE and constrained MLE for linear regression. In all three cases, the left column corresponds to correct constraints while the right column corresponds to incorrect constraints.

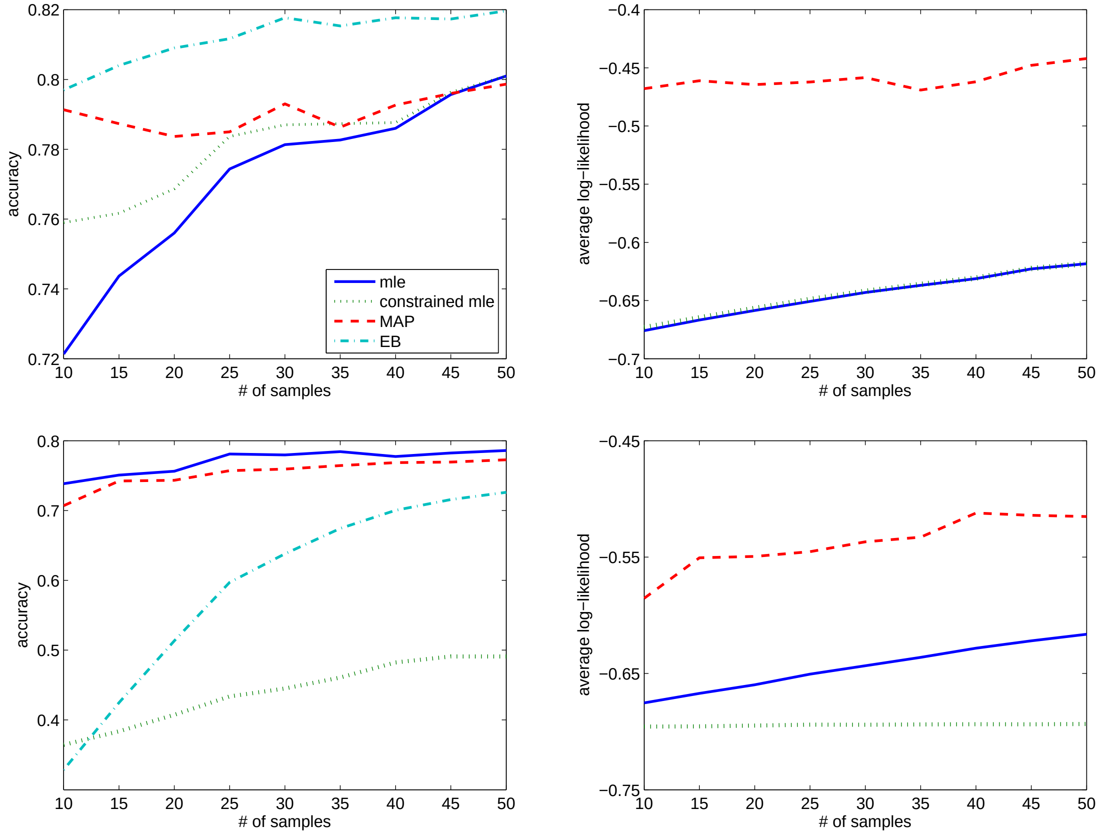


Figure 11: Average test set performance for logistic regression over 20 random train/test splits. Parameters are estimated by MLE, (hard) constrained MLE, and probabilistic constraint EB and MAP with full covariance matrix. L_2 regularization is applied to both MLE and constrained MLE. The top row corresponds to correct constraints while the bottom row corresponds to incorrect constraints. Average log-likelihood is not reported for EB since it can only be estimated approximately.

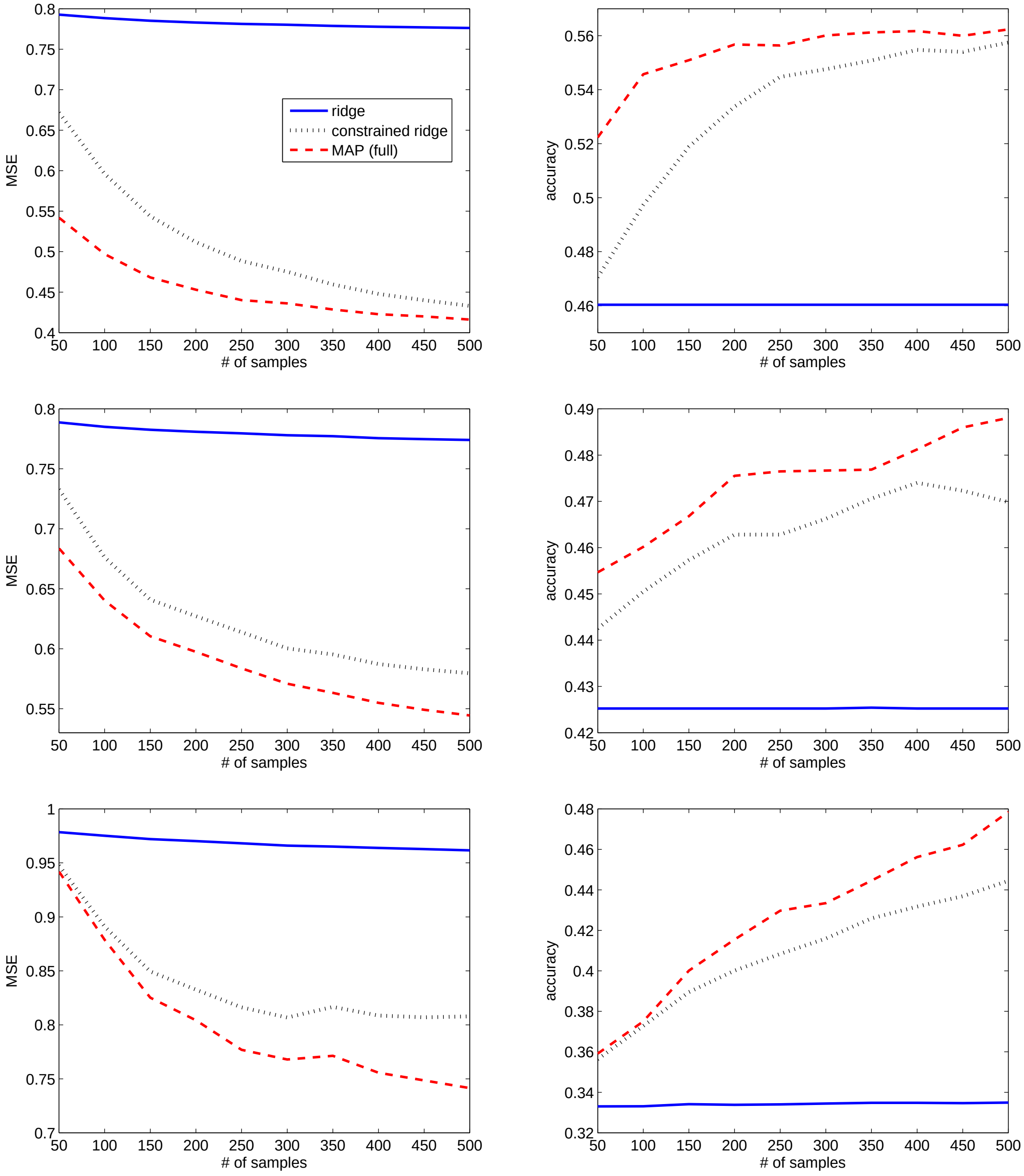


Figure 12: Test-set mean square error (MSE, left) and accuracy rates (right) over 10 iterations for sentiment prediction (top, middle corresponding to the two critics) and readability prediction (bottom). Regression parameters are estimated by ridge, (hard) constrained ridge and probabilistic constraint MAP with full covariance matrix.

CHAPTER V

LINGUISTIC KNOWLEDGE FOR METRIC LEARNING, WITH APPLICATION TO TEXT VISUALIZATION

Visual document analysis systems such as IN-SPIRE have demonstrated their applicability in managing large text corpora, identifying topics within a document and quickly identifying a set of relevant documents by visual exploration. The success of such systems depends on several factors with the most important one being the quality of the dimensionality reduction. This is obvious as visual exploration can be made possible only when the dimensionality reduction preserves the structure of the original space, i.e., documents that convey similar topics are mapped to nearby regions in the low dimensional 2D or 3D space.

Standard dimensionality reduction methods such as principal component analysis (PCA), locally linear embedding (LLE) [92], or t-distributed stochastic neighbor embedding (t-SNE) [112] take as input a set of feature vectors such as bag of words. An obvious drawback is that such methods ignore the textual nature of documents and instead consider the vocabulary words $v_1, \dots, v_n$ as abstract orthogonal dimensions.

In this chapter we introduce a framework for incorporating domain knowledge into dimensionality reduction for text documents. Our technique does not require any labeled data, therefore is completely unsupervised. In addition, it applies to a wide variety of domain knowledge.

We focus on the following type of non-Euclidean geometry where the distance between document x and y is defined as

$$d_T(x, y) = \sqrt{(x - y)^\top T(x - y)}. \tag{57}$$

Here $T \in \mathbb{R}^{n \times n}$ is a symmetric positive semidefinite matrix, and we assume that documents x, y are represented as term-frequency (tf) column vectors. Since T can always be written as $H^\top H$ for some matrix $H \in \mathbb{R}^{n \times n}$, an equivalent but sometimes more intuitive interpretation of (57) is to compose the mapping $x \mapsto Hx$ with the Euclidean geometry

$$d_T(x, y) = d_I(Hx, Hy) = \|Hx - Hy\|_2. \tag{58}$$

We can view T as encoding the semantic similarity between pairs of words and H as smoothing the tf vector by mapping observed words to related but unobserved words. Therefore, the geometry realized by (57) or (58) may be used to derive novel dimensionality reduction methods that are customized to text in general and to specific text domains in particular. The main challenge is to obtain the matrices H or T that describe the relationship among vocabulary words appropriately.

We consider three general ways of obtaining H or T using domain knowledge. The first corresponds to manually specifying H or T based on the semantic relationship among words (determined by domain expert). The second corresponds to constructing H or T by analyzing relationships between different words using corpus statistics. The third is based on knowledge obtained from linguistic resources. Whether to specify H directly or indirectly by specifying $T = H^\top H$ depends on the knowledge type and is discussed in detail in Section 5.2.

We investigate the performance of the proposed dimensionality reduction methods for three text domains: sentiment visualization for movie reviews, topic visualization for newsgroup discussion articles, and visual exploration of ACL papers. In each of these domains we evaluate the dimensionality reduction using several different quantitative measures. All the techniques mentioned in this chapter are unsupervised, making use of labels only for evaluation purposes.

Our conclusion is that all three approaches mentioned above improves dimensionality reduction for text upon standard embedding ($H = I$). Furthermore, geometries obtained from corpus statistics are superior to manually constructed geometries and to geometries derived from standard linguistic resources such as Word-Net. Combining heterogenous types of knowledge provides the best results.

5.1 Non-Euclidean Geometries

Dimensionality reduction methods often assume, either explicitly or implicitly, Euclidean geometry. For example, PCA minimizes the reconstruction error for a family of Euclidean projections. LLE uses the Euclidean geometry as a local metric. t-SNE is based on a neighborhood structure, determined again by the Euclidean geometry. The generic nature of the Euclidean geometry makes it somewhat unsuitable for visualizing text documents as the relationship between words conflicts with Euclidean orthogonality. We consider in this chapter several alternative geometries of the form (57) or (58) which are more suited for text and compare their effectiveness in visualizing documents.

As mentioned before, H smooths the tf vector x by mapping the observed words into observed and non-observed (but related) words. In case H is nonnegative, it can be further decomposed into a product of a non-negative column normalized matrix $R \in \mathbb{R}^{n\times n}$ and a non-negative diagonal matrix $D \in \mathbb{R}^{n\times n}$. The decomposition $H = RD$ shows that H has two key roles. It smooths related vocabulary words (realized by R) and it emphasizes some words over others (realized by D). Setting R_{ij} to a high value if w_i, w_j are similar and 0 if they are unrelated maps an observed word to a probability vector over related words in the vocabulary. The value D_{ii} captures the importance of v_i and therefore should be higher for important content words than for less important words or stop-words[1].

It is instructive to examine the matrices R and D in the case where the vocabulary words cluster in some meaningful way. Figure 13 gives an example where vocabulary

[1] The nonnegativity assumption of H is useful when constructing H by domain experts such as the method A in Section 5.2. In general, H needs not to be nonnegative for dimensionality reduction as in (58).

$$\begin{pmatrix}0.8 & 0.1 & 0.1 & 0 & 0\\ 0.1 & 0.8 & 0.1 & 0 & 0\\ 0.1 & 0.1 & 0.8 & 0 & 0\\ 0 & 0 & 0 & 0.9 & 0.1\\ 0 & 0 & 0 & 0.1 & 0.9\end{pmatrix}\begin{pmatrix}5 & 0 & 0 & 0 & 0\\ 0 & 5 & 0 & 0 & 0\\ 0 & 0 & 5 & 0 & 0\\ 0 & 0 & 0 & 3 & 0\\ 0 & 0 & 0 & 0 & 3\end{pmatrix}$$

Figure 13: An example of a decomposition $H = RD$ in the case of two word clusters $\{v_1, v_2, v_3\}$, $\{v_4, v_5\}$. The block diagonal elements in R represent the fact that words are mostly mapped to themselves, but sometimes are mapped to other words in the same cluster. The diagonal matrix indicates that the first cluster is more important than the second cluster for the purposes of dimensionality reduction.

words form two clusters. The matrix R may become block-diagonal with non-zero elements occupying diagonal blocks representing within-cluster word blending, i.e., words within each cluster are interchangeable to some degree. The diagonal matrix D represents the importance of different clusters. The word clusters are formed with respect to the visualization task at hand. For example, in the case of visualizing the sentiment content of reviews we may have word clusters labeled as “positive sentiment words”, “negative sentiment words” and “objective words”. In the case of visualizing news stories we may have word clusters representing news topics such as “politics”, “business” and “science and technology”.

In general, the matrices R, D may be defined based on the language or may be specific to document domain and visualization purpose. It is reasonable to expect that the words emphasized for visualizing topics in news stories might be different than the words emphasized for visualizing writing styles or sentiment content.

Applying the geometry (57) or (58) to dimensionality reduction is easily accomplished by first mapping document tf vectors $x \mapsto Hx$ and proceeding with standard dimensionality reduction techniques such as PCA or t-SNE. The resulting dimensionality reduction is Euclidean in the transformed space but non-Euclidean in the original space. In many cases, the vocabulary contains tens of thousands of words or more making the specification of T or H a complicated and error prone task. We describe in the next section several techniques for specifying these matrices in practice.

5.2 Linguistic Knowledge

5.2.1 A: Human Elicitation

In this method, a domain expert manually specifies $H = RD$ by specifying (R, D) based on the perceived relationship among the vocabulary words. More specifically, the user first constructs a hierarchical word clustering that may depend on the current text domain, and then specifies the matrices (R, D) based on the clustering.

Denoting the clusters by $C_1, \ldots, C_r$ (a partition of $\{v_1, \ldots, v_n\}$), R is set to

$$R_{ij} \propto \begin{cases} \rho_a, & i = j, v_i \in C_a \\ \rho_{ab}, & i \neq j, v_i \in C_a, v_j \in C_b \end{cases} .$$

The values $\rho_{ab}, a \neq b$ capture the semantic similarity between two clusters and the value ρ_{aa} captures the similarity of two different words within the cluster a. These values may be set manually by domain expert or automatically computed based on the clustering hierarchy (for example ρ_{ab} can be the inverse of the minimal number of tree edges traversed in moving from a to b). To maintain a probabilistic interpretation, the matrix R should be normalized so that its columns sum to 1. The diagonal matrix D is specified by setting the values

$$D_{ii} = d_a, \quad v_i \in C_a$$

according to the importance of word cluster C_a to the current visualization task.

We emphasize that as with the rest of the methods in this chapter, the manual specification is done without access to labeled data. Since manual clustering assumes some form of human intervention, it is reasonable to also consider cases where the user specifies H or T in an interactive manner. For example, the expert specifies an initial clustering of words and values for (R, D), views the resulting embeddings and adjusts the selection interactively until reaching a satisfactory embedding.

5.2.2 B: Contextual Diffusion

An alternative to manually specifying $T = DR^\top RD$ is to construct it based on similarity between the contextual distributions of the vocabulary words. The contextual distribution of word v is defined as

$$q_v(w) = p(w \text{ appears in } x | v \text{ appears in } x) \tag{59}$$

where x is a randomly drawn document. In other words q_v is the distribution governing the words appearing in the context of word v.

A natural similarity measure between distributions is the Fisher diffusion kernel proposed by [58]. Applied to contextual distributions as in [33] we arrive at the following similarity matrix

$$T(u,v) = \exp\left(-c \arccos^2\left(\sum_w \sqrt{q_u(w)q_v(w)}\right)\right).$$

where $c > 0$. Intuitively, the word u will be diffused into v depending on the geometric diffusion between the distributions of likely contexts.

We use the following formula to estimate the contextual distribution from a corpus

$$\begin{aligned} q_v(w) &= \sum_{x'} p(w, x'|v) = \sum_{x'} p(w|x', v)p(x'|v) \\ &= \sum_{x'} \text{tf}(w, x') \frac{\text{tf}(v, x')}{\sum_{x''} \text{tf}(v, x'')} \\ &= \left(\frac{1}{\sum_{x'} \text{tf}(v, x')}\right)\left(\sum_{x'} \text{tf}(w, x')\text{tf}(v, x')\right) \end{aligned} \tag{60}$$

where $\text{tf}(w, x)$ is the number of times word w appears in document x divided by the length of the document x. The contextual distribution q_v or diffusion matrix T above may be computed in an unsupervised manner without labels.

5.2.3 C: Web n-Grams

In method B the contextual distribution is computed using a large external corpus that is similar to the text being analyzed. An alternative that is especially useful when such a corpus is not easily available is to use generic resources to estimate the contextual distribution (59)-(60). One option is to use the publicly available Google n-gram dataset [9] which contains n-gram counts ($n \leq 5$) obtained from Google based on processing over a trillion words of running text to estimate T. More specifically, we compute the contextual distribution by considering the proportion of times two words appear together within the n-grams e.g., for $n = 2$ we have

$$q_v(w) = \frac{\text{\# of bigrams containing both } w \text{ and } v}{\text{\# of bigrams containing } v}.$$

5.2.4 D: Word-Net

In the last method, we consider using Word-Net, a standard linguistic resource, to specify T. This is similar to manual specification (method A) in that it builds upon experts' knowledge rather than corpus statistics. In contrast to method A, however, Word-Net is a carefully built resource containing more accurate and comprehensive linguistic information such as synonyms, hyponyms and holonyms. On the other hand, its generality puts it at a disadvantage as method A may be adapted to a specific text domain.

We follow [11] who compared five similarity measures between words based on Word-Net. In our experiments we use the measure of [54] (see also [57])

$$T(u, v) = \log \frac{p(u)p(v)}{2p(\text{lcs}(u, v))}$$

as it was shown to outperform the others. Above, lcs stands for the lowest common subsumer, i.e., the lowest node in the hierarchy that subsumes (is a hypernym of) both u and v. The quantity $p(u)$ is the probability that a randomly selected word in a corpus is an instance of the synonym set that contains word u.

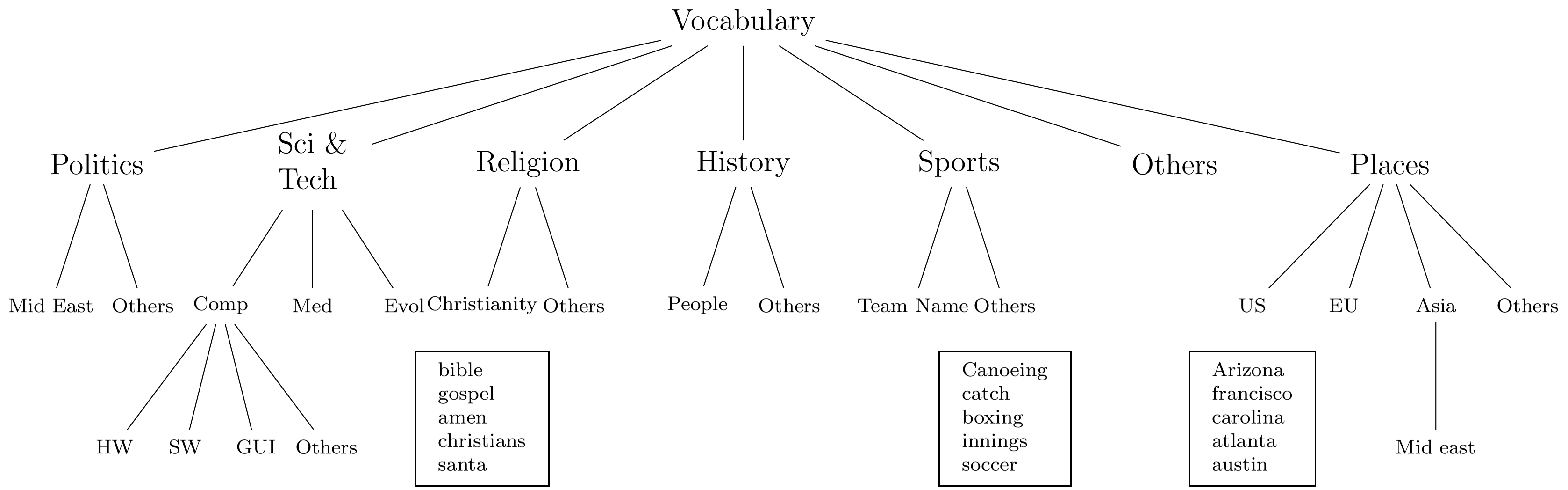


Figure 14: Manually specified hierarchical word clustering for the 20 newsgroup domain. The words in the frames are examples of words belonging to several bottom level clusters.

5.2.5 Convex Combination

In addition to individual methods we also consider their convex combinations

$$H^* = \sum_i \alpha_i H_i \quad \text{s.t.} \quad \alpha_i \geq 0, \sum_i \alpha_i = 1 \tag{61}$$

where H_i are matrices from methods A-D (obtained implicitly by specifying R and D for method A and T for methods B-D). Doing so allows us to combine heterogeneous types of domain knowledge including experts' knowledge and corpus statistics, leverage their diverse nature and potentially achieve better performance than any of the methods on its own.

5.2.6 Online Update via Bregman Projection

The requirement of creating word clusters in method A is often not an easy task. In the following, we consider domain knowledge in the form of triplets of words. Each triplet (u, v, w) states that word u is considered more similar to word v than word w, or equivalently $T(u, v) \geq T(u, w)$. We choose the triplet representation since it is relatively easy to specify, and exhibits less ambiguity than specifying relationship for a pair of words.

The constraint $T(u, v) \geq T(u, w)$ can also be motivated from the following example of classifying a document according to whether it conveys positive or negative sentiment with distance being defined by (57). Document A contains one word `excellent`, document B contains one word `good` and document C contains one word `dull`. The distance between A and B is computed to be

$$\begin{aligned} d^2_{AB} &= T(\text{‘}\texttt{excellent}\text{'}, \text{‘}\texttt{excellent}\text{'}) + T(\text{‘}\texttt{good}\text{'}, \text{‘}\texttt{good}\text{'}) - \\ & \quad T(\text{‘}\texttt{excellent}\text{'}, \text{‘}\texttt{good}\text{'}) - T(\text{‘}\texttt{good}\text{'}, \text{‘}\texttt{excellent}\text{'}). \end{aligned}$$

Similarly, we have

$$\begin{aligned} d^2_{AC} &= T(\text{‘}\texttt{excellent}\text{’}, \text{‘}\texttt{excellent}\text{’}) + T(\text{‘}\texttt{dull}\text{’}, \text{‘}\texttt{dull}\text{’}) - \\ & \quad T(\text{‘}\texttt{excellent}\text{’}, \text{‘}\texttt{dull}\text{’}) - T(\text{‘}\texttt{dull}\text{’}, \text{‘}\texttt{excellent}\text{’}). \end{aligned}$$

The distance d_{AB} is expected to be no larger than the distance d_{AC}. Since the matrix T is symmetric, if, in addition, the diagonal elements of T contain the same value, we have $d_{AB} \leq d_{AC}$ implies that $T(\text{‘}\texttt{excellent}\text{’}, \text{‘}\texttt{good}\text{’}) \geq T(\text{‘}\texttt{excellent}\text{’}, \text{‘}\texttt{dull}\text{’})$. This matches our intuition that the word `excellent` is more similar to `good` than to `dull` for the task of sentiment prediction.

The constraint $T(u, v) \leq T(u, w)$ can be written equivalently as

$$\begin{aligned} T(u,v) \leq T(u,w) \quad &\Longleftrightarrow \quad T(u,v) + T(v,u) - T(u,w) - T(w,u) \leq 0 \\ &\Longleftrightarrow \quad e_u^\top K e_v + e_v^\top K e_u - e_u^\top K e_w - e_w^\top K e_u \leq 0 \\ &\Longleftrightarrow \quad \text{tr}\left[K\left(e_v e_u^\top + e_u e_v^\top - e_w e_u^\top - e_u e_w^\top\right)\right] \leq 0 \qquad (62) \end{aligned}$$

where e_u denotes a unit vector with the u-th element being 1 and 0 otherwise. Note the rank of matrix $A = e_u(e_v - e_w)^\top + (e_v - e_w)e_u^\top$ is two.

Given triplets of words, we consider finding a new matrix T so that T satisfies all constraints and stays close to the initial matrix T_0. While there may be several ways to define closeness between two matrices, we choose to use the LogDet divergence (see e.g. Appendix C) which can be derived from the differential relative entropy between two multivariate Gaussians with the same mean vector μ, and covariance matrix T and T_0 respectively. To see this, note that the differential relative entropy is defined as $D(f\|g) = \int f \log f - f \log g$. The first term $\int f \log f$ is simply the negative of the differential entropy of $\mathcal{N}(\mu, T)$, which is computed to be $-\frac{1}{2}\log\left((2\pi)^n|T|\right) - \frac{n}{2}$, and the second term $\int f \log g$ is

$$
\begin{aligned}
& \int N(x|\mu, T) \log N(x|\mu, T_0) \\
= \quad & \int N(x|\mu, T) \left(-\frac{1}{2}(x-\mu)^\top T_0^{-1}(x-\mu) - \log\left((2\pi)^{\frac{n}{2}}|T_0|^{\frac{1}{2}}\right)\right) \\
= \quad & -\frac{1}{2}\int N(x|\mu, T)\mathrm{tr}\Big(T_0^{-1}(x-\mu)(x-\mu)^\top\Big) - \int N(x|\mu, T)\log\left((2\pi)^{\frac{n}{2}}|T_0|^{\frac{1}{2}}\right) \\
= \quad & -\frac{1}{2}\mathrm{tr}\Big(T_0^{-1}\mathbb{E}\left[(x-\mu)(x-\mu)^\top\right]\Big) - \log\left((2\pi)^{\frac{n}{2}}|T_0|^{\frac{1}{2}}\right) \\
= \quad & -\frac{1}{2}\mathrm{tr}\left(T_0^{-1}T\right) - \log\left((2\pi)^{\frac{n}{2}}|T_0|^{\frac{1}{2}}\right).
\end{aligned}
$$

Each projection into the convex set (62) defined by the word triplet can be computed in closed form. Detailed derivation is given in Appendix E. Note, the projection preserves the symmetry of the matrix. While the diagonal elements of T created from method B – D contain the same value, they no longer remain the same during projections.

5.3 Experiments

We evaluate the proposed methods by experimenting on two text datasets where domain knowledge is relatively easy to obtain (especially for method A and B). Preprocessing includes lower-casing, stop words removal, stemming, and selecting the most frequent 2000 words for both datasets.

The first is the Cornell sentiment scale dataset of movie reviews from 4 critics [80]. The visualization in this case focuses on the sentiment quantity of either 1 (very bad) or 4 (very good) [82]. For method A, we use the General Inquirer resource[2] to partition the vocabulary into three clusters conveying positive, negative or neutral sentiment. While visualizing documents from one particular author, the rest of the reviews from other three authors can be used as an estimate of contextual distribution for method B.

[2] http://www.wjh.harvard.edu/∼inquirer/

The second text dataset is the 20 newsgroups. It consists of newsgroup articles from 20 distinct newsgroups and is meant to demonstrate topic visualization. In this case one of the authors designed a hierarchical clustering of the vocabulary words based on general knowledge of English language (see Figure 14 for a partial clustering hierarchy) without access to labels. The contextual distribution for method B is estimated from the Reuters RCV1 dataset [64] which consists of news articles from Reuters.com in the year 1996 and 1997.

Method C uses Google n-gram which provides a massive scale resource for estimating the contextual distribution. In the case of Word-Net (method D) we used Pedersen's implementation of Jiang and Conrath's similarity measure[3]. Note, for these two methods, the obtained information is not domain specific but rather represents general semantic relationships between words.

In our experiments below we focused on two dimensionality reduction methods: PCA and t-SNE. PCA is a well known classical method while t-SNE [112] is a recent dimensionality reduction technique for visualization purposes. The use of t-SNE is motivated by the fact that it was shown to outperform LLE, CCA, MVU, Isomap, and Laplacian eigenmaps when the dimensionality of the data is reduced to two or three.

To measure the dimensionality reduction quality, we visualize the data as a scatter plot with different data groups (topics, sentiments) displayed with different markers and colors. Our quantitative evaluation of the visualization is based on the fact that documents belonging to different groups (topics, sentiments) should be spatially separated in the 2-D space. Specifically, we used the following indices:

(i) The weighted intra-inter criteria is a standard clustering quality index that is invariant to non-singular linear transformations of the embedded data. It equals $\mathrm{tr}(S_T^{-1} S_W)$ where S_W is the within-cluster scatter matrix, $S_T = S_W + S_B$ is the

[3] http://wn-similarity.sourceforge.net/

total scatter matrix, and S_B is the between-cluster scatter matrix [36].

(ii) The Davies Bouldin index is an alternative to (i) that is similarly based on the ratio of within-cluster scatter to between-cluster scatter [29].

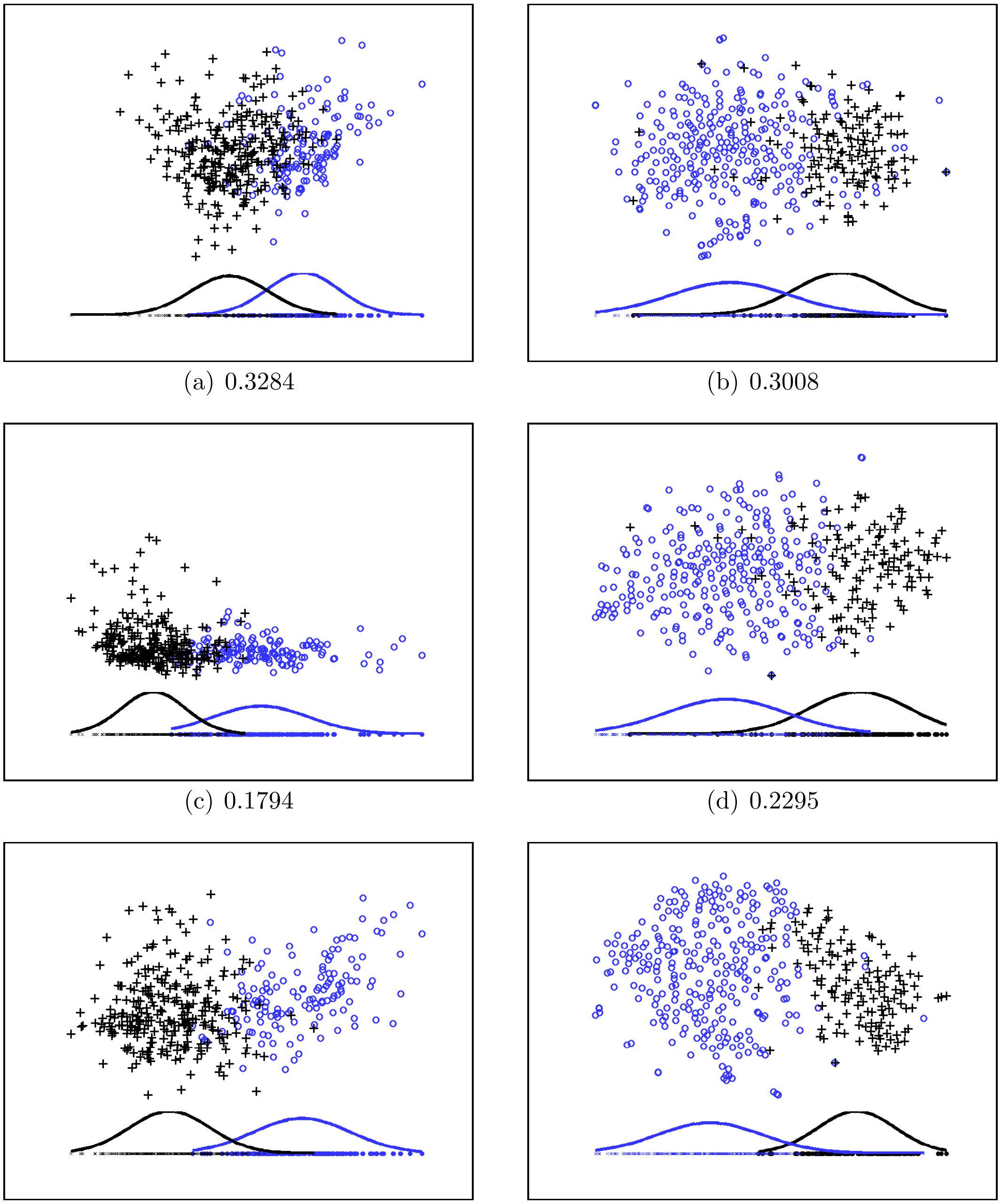
(iii) Classification error rate of a k-NN classifier that applies to data groups in the 2-D embedded space. Despite the fact that we are not interested in classification per se (otherwise we would classify in the original high dimensional space), it is an intuitive and interpretable measure of cluster separation.

(iv) An alternative to (iii) is to project the embedded data onto a line which is the direction returned by applying Fisher's linear discriminant analysis to the embedded data. The projected data from each group is fitted to a Gaussian whose separation is used as a proxy for visualization quality. In particular, we summarize the separation of the two Gaussians by measuring the overlap area. While (iii) corresponds to the performance of a k-NN classifier, method (iv) corresponds to the performance of Fisher's LDA classifier.

Labeled data is not used during the dimensionality reduction stage but it is used in each of the above measures for evaluation purposes.

Figure 15-16 display both qualitative and quantitative evaluation of PCA and t-SNE for the sentiment and newsgroup domains for $H = I$ (top row), manual specification (middle row) and contextual distribution (bottom column). In general for both domains, methods A and B perform better both qualitatively and quantitatively (indicating by the numbers in the top two rows) than the original dimensionality reduction with method B outperforming method A.

Tables 9-10 compare evaluation measures (i) and (iii) for different types of domain knowledge. Table 9 corresponds to the sentiment domain where we conducted separate experiments for four movie critics. Table 10 corresponds to the newsgroup domain where two tasks were considered. The first involves three newsgroups



(a) 0.3284

(b) 0.3008

(c) 0.1794

(d) 0.2295

(e) 0.1385

(f) 0.1093

Figure 15: Qualitative evaluation of dimensionality reduction for the sentiment domain. The left column displays PCA reduction while the right column displays t-SNE. The top row corresponds to no domain knowledge ($H = I$) reverting PCA and t-SNE to their original form. The middle row corresponds to manual specification (method A). The bottom row corresponds to contextual diffusion (method B). Different sentiment labels are marked with different colors and marks.
The graphs were rotated such that the direction returned by applying Fisher linear discriminant onto the projected 2D coordinates aligns with the positive x-axis. The bell curves are Gaussian distributions fitted from the x-coordinates of the projected data points (after rotation). The numbers displayed in each sub-figure are computed from measure (iv).

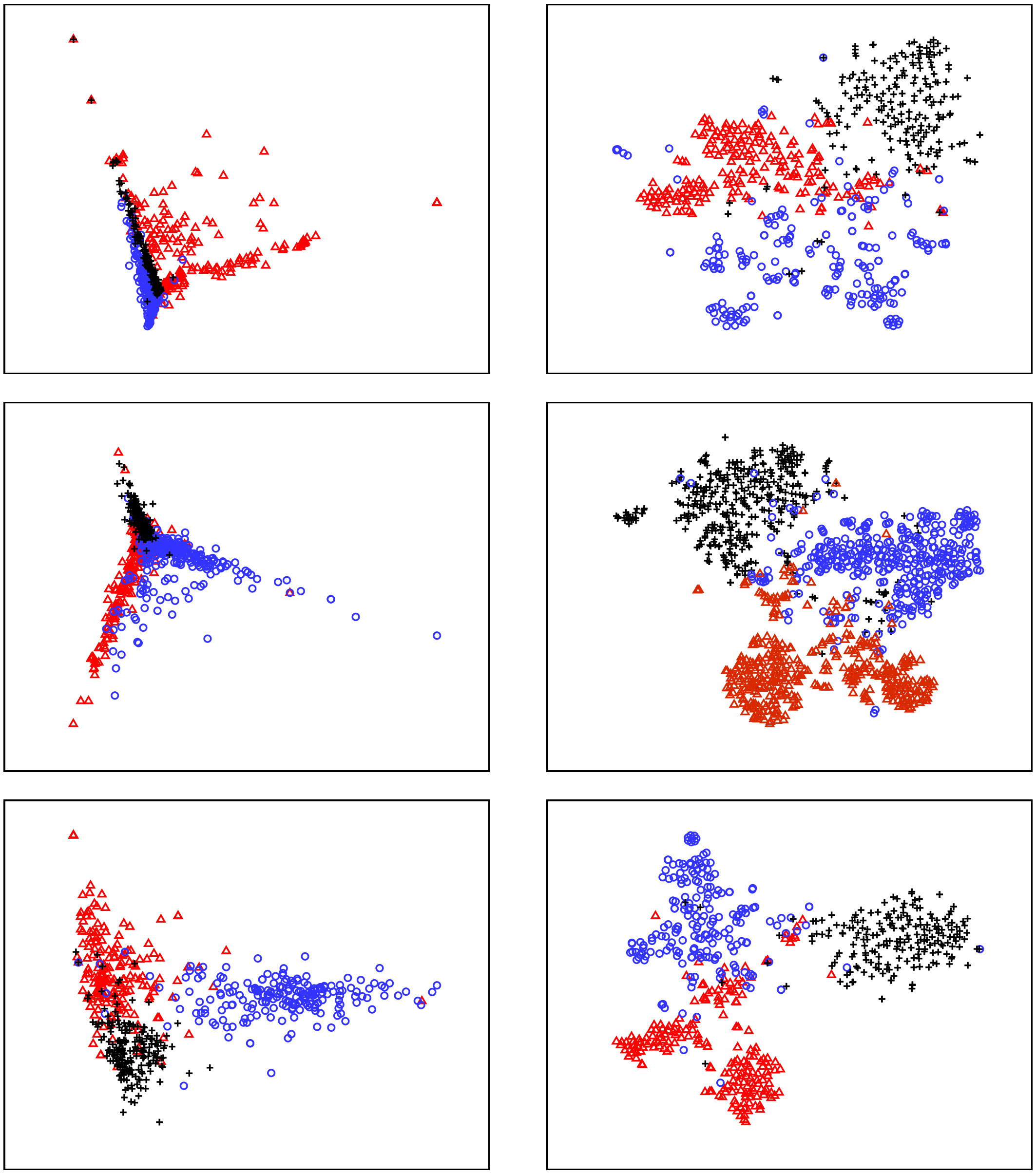


Figure 16: Qualitative evaluation of dimensionality reduction for the newsgroup domain. The left column displays PCA reduction while the right column displays t-SNE. The top row corresponds to no domain knowledge ($H = I$) reverting PCA and t-SNE to their original form. The middle row corresponds to manual specification (method A). The bottom row corresponds to contextual diffusion (method B). Different newsgroup labels are marked with different colors and marks.

(comp.sys.mac.hardware vs. rec.sports.hockey vs. talk.politics.mideast) and the second involves four newsgroups (rec.autos vs. rec.motorcycles vs. rec.sports.baseball vs. rec.sports.hockey). It is clear from these two tables that the contextual diffusion, Google n-gram, and Word-Net generally outperform the original $H = I$ matrix. The best method varies from task to task but the contextual diffusion and Google n-gram in general result in good performance.

Table 9: Quantitative evaluation of dimensionality reduction for visualization in the sentiment domain. Each of the four columns corresponds to a different movie critic from the Cornell dataset (see text). The top five rows correspond to measure (i) (lower is better) and the bottom five rows correspond to measure (iii) ($k = 5$, higher is better). Results were averaged over 40 cross validation iterations. We conclude that all methods outperform the original $H = I$ with the contextual diffusion and manual specification generally outperforming the others.

	Dennis Schwartz		James Berardinelli		Scott Renshaw		Steve Rhodes	
	PCA	t-SNE	PCA	t-SNE	PCA	t-SNE	PCA	t-SNE
$H = I$	1.8625	1.8781	1.4704	1.5909	1.8047	1.9453	1.8013	1.8415
A	1.8474	1.7909	1.3292	1.4406	1.6520	1.8166	**1.4844**	1.6610
B	**1.4254**	**1.5809**	**1.3140**	**1.3276**	**1.5133**	**1.6097**	1.5053	**1.6145**
C	1.6868	1.7766	1.3813	1.4371	1.7200	1.8605	1.7750	1.7979
$H = I$	0.6404	0.7465	0.8481	0.8496	0.6559	0.6821	0.6680	0.7410
A	0.6011	0.7779	**0.9224**	0.8966	0.7424	0.7411	**0.8350**	0.8513
B	**0.8831**	**0.8554**	0.9188	**0.9377**	**0.8215**	**0.8332**	0.8124	**0.8324**
C	0.7238	0.7981	0.8871	0.9093	0.6897	0.7151	0.6724	0.7726

We also examined convex combinations

$$\alpha_1 H_A + \alpha_2 H_B + \alpha_3 H_C + \alpha_4 H_D \tag{63}$$

with $\sum \alpha_i = 1$ and $\alpha_i \geq 0$. Table 11 displays quantitative results using evaluation measures (i), (ii) and (iii) where k is chosen to be 5 for (iii). The first four rows correspond to method A, B, C and D and the bottom row corresponds to a convex combination found which minimizes the unsupervised evaluation measure (ii) (i.e. the search for the optimal combination is based on (ii) that does not require labeled data). Note that the convex combination also outperforms method A, B, C, and D for

Table 10: Quantitative evaluation of dimensionality reduction for visualization for two tasks in the news article domain. The numbers in the top five rows correspond to measure (i) (lower is better), and the numbers in the bottom five rows correspond to measure (iii) ($k = 5$) (higher is better). We conclude that contextual diffusion (B), Google n-gram (C), and Word-Net (D) tend to outperform the original $H = I$.

	PCA (1)	PCA (2)	t-SNE (1)	t-SNE (2)
$H = I$	1.5391	1.4085	1.1649	1.1206
B	1.2570	**1.3036**	1.2182	1.2331
C	**1.2023**	1.3407	**0.7844**	**1.0723**
D	1.4475	1.3352	1.1762	1.1362

	PCA (1)	PCA (2)	t-SNE (1)	t-SNE (2)
$H = I$	0.8461	0.5630	0.9056	0.7281
B	0.7381	**0.6815**	0.9110	0.6724
C	0.8420	0.5898	**0.9323**	0.7359
D	**0.8532**	0.5868	0.9013	**0.7728**

measure (i) and more impressively for measure (iii) which is a supervised measure that uses labeled data. In general, by combining heterogeneous types of domain knowledge, we may further improve the quality of dimensionality reduction for visualization, and the search for such a combination may be accomplished without the use of labeled data.

For online update, we experimented with the sentiment datasets, and reported the accuracy of predicting positive or negative sentiments using k-nearest neighbor classifier. The constraints are derived by randomly picking triplets of words from the vocabulary. If two and only two of the words are labeled the same sentiment according to the General Inquirer resource, we add the triplet to the constraint pool. The initial matrix is computed from the contextual diffusion which often achieves the best performance among methods A – D.

Figure 17 reports the classification accuracy for $k = 5$. The black dotted lines correspond to the Euclidean geometry $T = I$ while the red lines correspond to modifying the geometry defined by the contextual diffusion with varying number of constraints. We observe substantial improvement over the contextual diffusion for two out of four

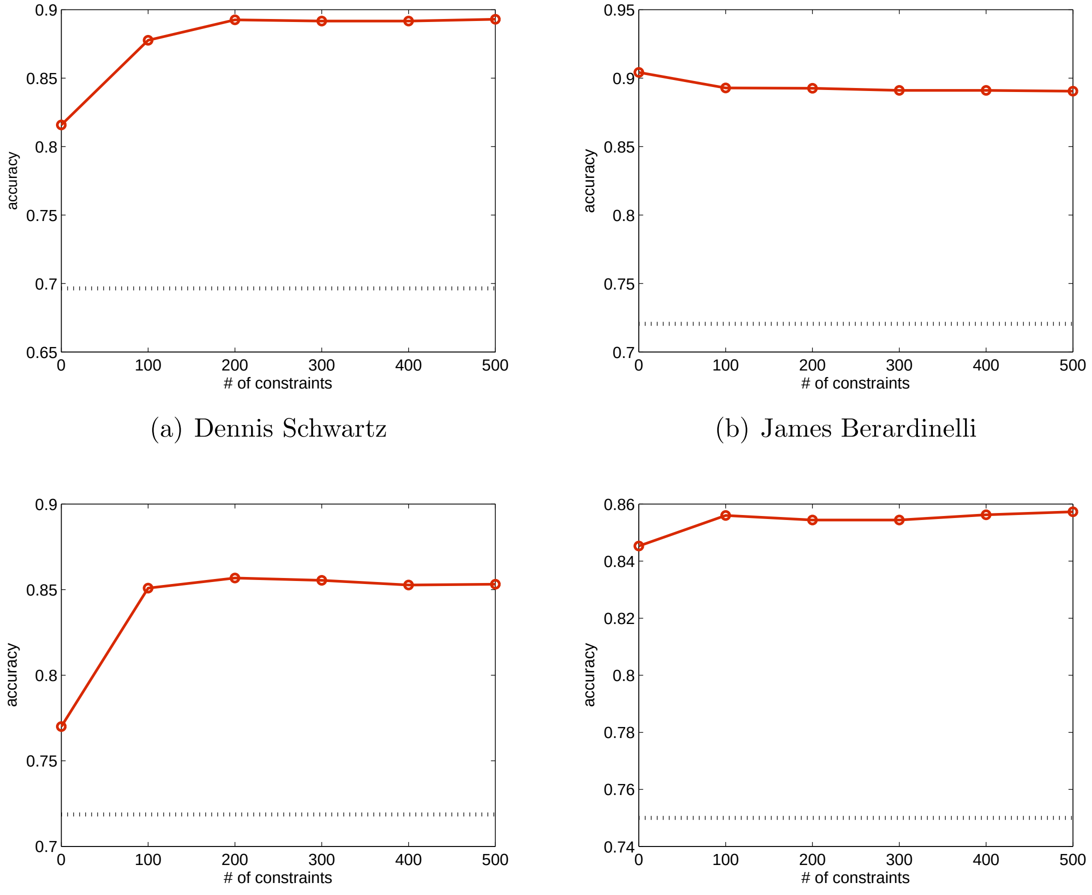
Table 11: Three evaluation measures (i), (ii), and (iii) (see the beginning of the section for description) for convex combinations (63) using different values of α. The first four rows represent methods A, B, C, and D. The bottom row represents a convex combination whose coefficients were obtained by searching for the minimizer of measure (ii). Interestingly the minimizer also performs well on measure (i) and more impressively on the labeled measure (iii).

$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$	(i)	(ii)	(iii) (k=5)
(1,0,0,0)	0.5756	-3.9334	0.7666
(0,1,0,0)	0.5645	-4.6966	0.7765
(0,0,1,0)	0.5155	-5.0154	0.8146
(0,0,0,1)	0.6035	-3.1154	0.8245
(0.3,0.4,0.1,0.2)	**0.4735**	**-5.1154**	**0.8976**

tasks, while all of them outperform the Euclidean geometry.

Finally, we demonstrate the effect of domain knowledge on a new dataset that consists of all oral papers appearing in ACL 2001 – 2009. For the purpose of manual specification, we obtain 1545 unique words from paper titles, and assign for each word relatedness scores for the following clusters: morphology/phonology, syntax/parsing, semantics, discourse/dialogue, generation/summarization, machine translation, retrieval/categorization and machine learning. The score takes value from 0 to 2, where 2 represents the most relevant. The score information is then used to generate the transformation matrix R. We also assign for each word an importance value ranging from 0 to 3 (larger the value, more important the word). This information is used to generate the diagonal matrix D.

Figure 18 shows the projection of all 2009 papers using t-SNE (papers from 2001 to 2008 are used to estimate contextual diffusion). Using Euclidean geometry $H = I$ (Figure 18 left) results in a Gaussian like distribution which does not provide much insight into the data. Using a manually specified H (Figure 18 left) we get two clear clusters, the smaller containing papers dealing with machine translation and multilingual tasks. Interestingly, the contextual diffusion results in a one-dimensional



(a) Dennis Schwartz

(b) James Berardinelli

(c) Scott Renshaw

(d) Steve Rhodes

Figure 17: Sentiment prediction using k nearest neighbor classifier for $k = 5$. The black dotted lines correspond to the Euclidean geometry $T = I$ while the red lines correspond to modifying the geometry defined by the contextual diffusion with varying number of constraints.

manifold. Investigating the papers along the curve (from bottom to top) we find that it starts with papers discussing semantics and discourse (south), continues to structured prediction and segmentation (east), continues to parsing and machine learning (north), and then moves to sentiment prediction, summarization and IR (west) before returning to the center. Another interesting insight that we can derive is the relative discontinuity between the bottom part (semantics and discourse) and the rest of the curve. It seems spatial separability is higher in that area than in the other areas where the curve nicely traverses different regions continuously.

5.4 Discussion

Despite having a long history, dimensionality reduction is still an active research area. Broadly speaking, dimensionality reduction methods may be classified as projective or manifold based [12]. The first projects data onto a linear subspace (e.g., PCA and canonical correlation analysis) while the second traces a low dimensional nonlinear manifold on which data lies (e.g., multidimensional scaling, isomap, Laplacian eigenmaps, LLE and t-SNE). The use of dimensionality reduction for text documents is surveyed by [108] who also describe current homeland security applications.

Dimensionality reduction is closely related to metric learning. [114] is one of the earliest papers that focus on learning metrics of the form (57). In particular they try to learn matrix T in an supervised way by expressing relationships between pairs of samples. A representative paper on unsupervised metric learning for text documents is [62] which learns a metric on the simplex based on the geometric volume of the data.

Incorporating domain knowledge into dimensionality reduction is a relatively unexplored direction. A recent exception is the work by [28] who incorporates domain knowledge by expressing relationships between pairs of samples. Doing so leads to a constrained clustering optimization problem. Our contribution, on the other hand, incorporates domain knowledge into the dimensions of the ambient space with a particular emphasis on text (where in the case of tf documents the dimensions correspond to words).

We focus in this chapter on visualizing a corpus of text documents using a 2-D scatter plot. While this is perhaps the most popular and practical text visualization technique, other methods such as [103], [48], [47], [79], [7], [68] exist. Techniques developed in this chapter may be ported to enhance these alternative visualization methods as well.

We introduce several ways of incorporating domain knowledge into dimensionality reduction for visualizing text documents. The proposed methods of manual specification, contextual distribution, Google n-grams and Word-Net. All outperform in general the baseline $H = I$, which is the one currently used in most text visualization systems.

The answer to the question of which method is best depends on both the domain and the task at hand. For small tasks with limited vocabulary, manual specification could achieve best results. A large vocabulary size makes manual specification less accurate and effective. In cases where we have access to a large external corpus that is similar to the one we are interested in visualizing, contextual diffusion is an excellent choice. Lacking such a domain specific dataset estimating the contextual distribution using the generic Google n-gram is a good substitute. Word-Net captures relationships (such as synonyms and hyponyms) other than occurrence statistics between vocabulary words, and could be useful for certain tasks. Finally, the effectiveness of dimensionality reduction methods can be increased further by carefully combining different types of domain knowledge ranging from semantic similarity to occurrence statistics.

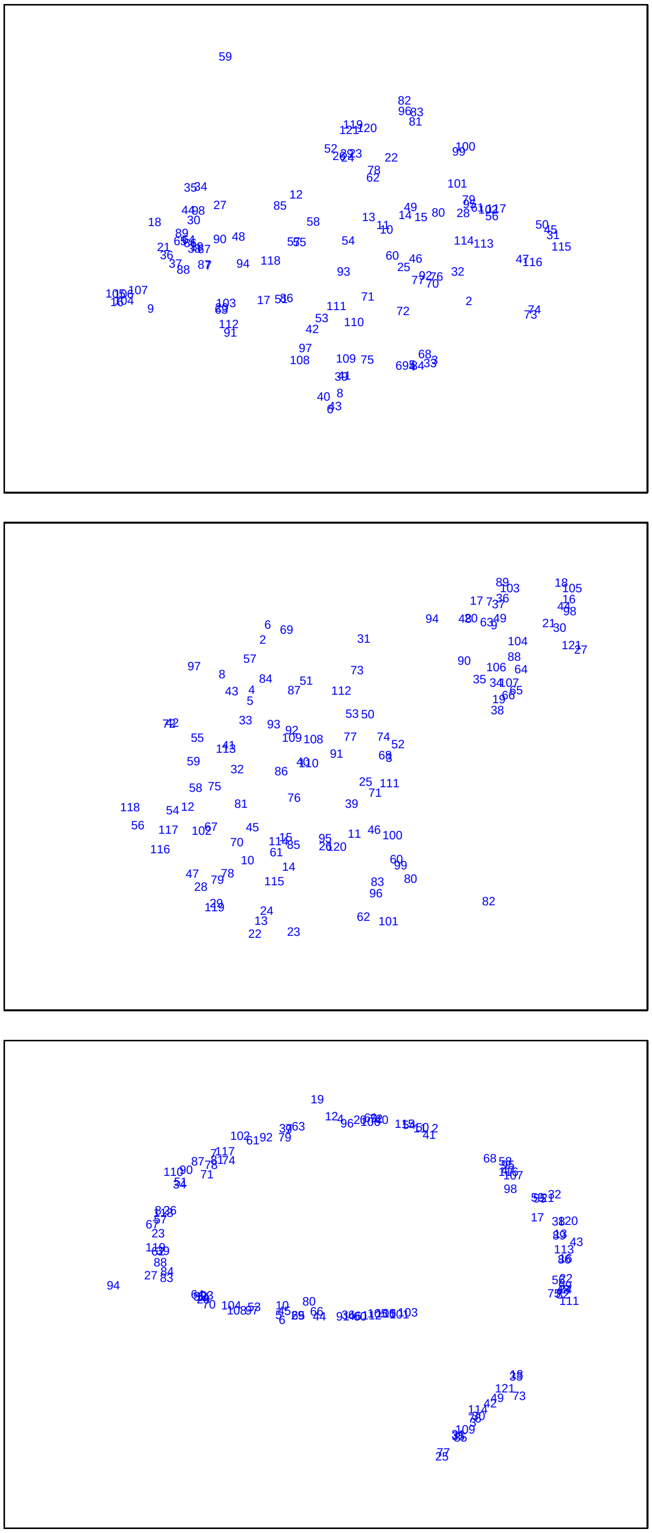


Figure 18: Qualitative evaluation of dimensionality reduction for the ACL dataset using t-SNE. Left: no domain knowledge ($H = I$); Middle: manual specification (method A); Right: contextual diffusion (method B). Each document is labeled by its assigned id from ACL anthology. See text for more details.

CHAPTER VI

CONCLUSION

In this thesis, we consider how domain knowledge in the form of probability constraints relates to the parameters of a probabilistic model. We address the issue in the case of conditional random fields, and develop the isotonic CRFs which are variants of CRFs with isotonic constraints over the parameter space. We apply the isotonic CRFs to sentiment prediction and information extraction, and show a consistent improvement in prediction accuracy over the regular CRFs.

With the observation that domain knowledge provided by humans often holds with some degree of uncertainty, we propose to explicitly model domain knowledge uncertainty by specifying the probability the knowledge is expected to hold, and aggregate both domain knowledge and its uncertainty into the learning process within a hierarchical Bayes framework. In contrast to hard parameter constraints, the approach is effective even when the domain knowledge is inaccurate and generally results in superior modelling accuracy.

Finally, we address the problem of incorporating general linguistic knowledge into the geometric assumptions made by learning algorithms for metric learning and text visualization. We show how to obtain knowledge from domain experts and corpus statistics, and provide a way so that users are not required to make their knowledge available immediately. We demonstrate the effort leads to an improved metric for documents, and foster better visual understanding of text corpus.

We identify three areas of future work:

- We are interested in extending the work to web search, online advertising and social networks. An example is the static ranking, which produces a query-independent ordering of web pages, and is of essential importance for problems such as index selection. Features that have shown to be useful for static ranking include PageRank score, popularity (described by the number of times the webpage has been visited by users over some period of time) and page statistics such as the frequency of the most common terms [89]. Therefore, it is natural to expect that a webpage should receive a higher ranking score than the other if they have the same feature values except that this one is more popular. In the case of a linear model, this is equivalent to enforcing the non-negativity constraint for the parameter corresponding to the popularity feature. However, modern search engineers usually employ nonlinear models such as decision tree, and it is not clear what parameter constraints this kind of knowledge corresponds to. An interesting problem therefore is whether we can learn a function that respects the above isotonic property, without resorting to postprocessing steps that may involve isotonic regression.

- We briefly address in Chapter 3 how to elicit constraints from some auxiliary dataset. The auxiliary dataset used there is very similar to the one used for training, which makes elicitation an easier task to accomplish. The question that follows consequently is whether we can elicit constraints from datasets that are dissimilar in nature but still relate to the target problem, and how uncertainty can be addressed in the case. Some preliminary work has been done in [16]. They observe that many structured prediction problems have a companion binary decision problem of predicting whether an input can produce a good structure or not, and that it is often very easy to obtain the answers for the companion problem. For example, a companion problem for the part-of-speech tagging is to ask whether a given sequence of words has a corresponding legitimate sequence of POS tags. The information from the companion problem is formulated as two conditions (or constraints) to be satisfied by the weight to be learned for the target problem. The condition corresponding to the negative examples from the companion problem is very similar to the one in [40] which requires to classify all points in the polyhedral set to the same class.

- We describe the problem of noisy labels in Chapter 2 as a result of the experts' limitations of expertise or lack of dedication. Repeated labeling has shown to improve the labeling quality when it can be applied. However, in some cases we may not have control over the assignment of examples to annotators. An example is the search engine where users are considered as teachers and webpage clicks as labels. This has been addressed in [32, 31] under an extreme assumption that the annotator can be either good or evil. A more realistic assumption would be that teachers are not perfect, but they are not evil either. How to remedy the labeling uncertainty in this case remains an open problem, and is of both theoretical and practical interest.

APPENDIX A

EDGEWORTH EXPANSION

Edgeworth expansion is a method to approximate a distribution in terms of its cumulants [23]. Consider random variables X, Y with cumulants κ_n, γ_n, $n \in \mathbb{N}$. The characteristic functions of X and Y satisfy

$$\mathrm{E}\left(e^{itX}\right) = \exp\left(\sum_{n=1}^{\infty} (\kappa_n - \gamma_n) \frac{(it)^n}{n!}\right) \mathrm{E}\left(e^{itY}\right). \tag{64}$$

If $\kappa_1 = 0$, $\kappa_2 = 1$, and $Y \sim N(0,1)$, Equation (64) becomes

$$\mathrm{E}\left(e^{itX}\right) = \exp\left(\sum_{n=3}^{\infty} \kappa_n \frac{(it)^n}{n!}\right) \exp\left(-\frac{t^2}{2}\right). \tag{65}$$

We define $P_n(u)$ as the coefficient of z^n in the expansion of the following expression

$$\exp\left(\sum_{n=1}^{\infty} \frac{\kappa_{n+2}}{(n+2)!} u^{n+2} z^n\right) = 1 + \sum_{n=1}^{\infty} P_n(u) z^n.$$

By setting $z = 1$ and $u = it$, Equation 65 becomes

$$\mathrm{E}\left(e^{itX}\right) = \left(1 + \sum_{n=1}^{\infty} P_n(it)\right) \exp\left(-\frac{t^2}{2}\right). \tag{66}$$

Since the characteristic function is the conjugate of the Fourier transform we may recover the pdf of X by applying the inverse Fourier transform to both sides of (66)

$$f(x) = \left(1 + \sum_{n=1}^{\infty} P_n(-D)\right) \phi(x)$$

where $\phi(x)$ is the pdf of a standard normal random variable. The operator D is the differential operator with respect to x since $(it)^n \exp\left(-\frac{t^2}{2}\right)$ is the Fourier transform of $(-1)^n D^n \phi(x)$.

The first two and three terms of the expansions

$$f(x) \approx \phi(x) - \frac{\kappa_3}{6}\phi^{(3)}(x)$$

$$f(x) \approx \phi(x) - \frac{\kappa_3}{6}\phi^{(3)}(x) + \frac{\kappa_4}{24}\phi^{(4)}(x) + \frac{\kappa_3^2}{72}\phi^{(6)}(x)$$

are equivalent to

$$\begin{aligned}\frac{f(x)}{\phi(x)} &\approx 1 + H_3(x)\frac{\kappa_3}{6} \\ \frac{f(x)}{\phi(x)} &\approx 1 + H_3(x)\frac{\kappa_3}{6} + H_4(x)\frac{\kappa_4}{24} + H_6(x)\frac{\kappa_3^2}{72}\end{aligned}$$

since $\phi^{(n)}(x) = (-1)^n H_n(x)\phi(x)$ where $H_n(x)$ are the Hermite polynomials.

APPENDIX B

WISHART AND INVERTED WISHART DISTRIBUTION

The inverted Wishart distribution is the matrix variate generalization of the inverted gamma distribution defined as follows:

Definition 1. A $p \times p$ random symmetric positive definite matrix V is said to be distributed as inverted Wishart with m $(> 2p)$ degrees of freedom and inverse scale matrix Ψ, if its pdf is given by

$$\frac{2^{-\frac{1}{2}(m-p-1)p}|\Psi|^{\frac{1}{2}(m-p-1)}}{\Gamma_p\left(\frac{1}{2}(m-p-1)\right)|V|^{\frac{1}{2}m}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(V^{-1}\Psi\right)\right)$$

where Ψ is a $p \times p$ positive definite matrix, and $\Gamma_p(\cdot)$ is the multivariate gamma function.

This distribution, denoted as $IW_p(m, \Psi)$, has been used as a conjugate prior for the covariance matrix in a normal distribution.

Closely related to inverted Wishart distribution is Wishart distribution, whose discovery has contributed enormously to the development of multivariate analysis.

Definition 2. A $p \times p$ random symmetric positive definite matrix S is said to have a Wishart distribution $W_p(n, \Sigma)$ with parameters p, n $(\geq p)$, and Σ ($p \times p$ positive definite matrix), if its pdf is given by

$$\frac{|S|^{\frac{1}{2}(n-p-1)}}{2^{\frac{1}{2}np}\Gamma_p\left(\frac{1}{2}n\right)|\Sigma|^{\frac{1}{2}n}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\Sigma^{-1}S\right)\right).$$

The relation between the Wishart and inverted Wishart distributions is given in the following theorem. More properties of these two distributions can be found in [44].

Theorem B.0.1. *Let $V \sim IW_p(m, \Psi)$, then $V^{-1} \sim W_p(m-p-1, \Psi^{-1})$.*

APPENDIX C

BREGMAN DIVERGENCE

Bregman divergence generalizes the Euclidean distance. Bregman projection projects a point onto a convex set with Bregman divergence as distance measure. It includes the classic metric projections (projections under Euclidean distance) as a special case. In the following, we give a brief introduction of Bregman divergence and Bregman projection. More details can be found in [10, 13].

Let S be a nonempty open convex set, such that its closure $\bar{S}$ is contained in $\Lambda \subseteq \mathbb{R}^n$. Function $f : \Lambda \to \mathbb{R}$ is assumed to be continuous and strictly convex on $\bar{S}$, and has continuous first partial derivatives at every $x \in S$ denote by $\nabla f(x)$. From $f(x)$ we construct the function $D_f : \bar{S} \times S \to R$ defined as

$$D_f(x, y) = f(x) - f(y) - \langle \nabla f(y), x - y \rangle.$$

This function is called Bregman divergence. It may be interpreted as $f(x) - h(x)$ where $h(z)$ represents the hyperplane that is tangent to the epigraph of f at the point $(y, f(y))$. See Figure 19 for an illustration.

Examples of Bregman divergence include the squared Euclidean distance ($D_f(x, y) = \frac{1}{2}\|x-y\|^2$), relative entropy or Kullback-Leibler cross entropy ($D_f(x, y) = \sum_i x_i \log \frac{x_i}{y_i} - \sum_i x_i + \sum_i y_i$) and the Itakura-Saito distance ($D_f(x, y) = \sum_i \frac{x_i}{y_i} - \log \frac{x_i}{y_i} - 1$). They are generated by functions $f(x) = \frac{1}{2}\|x\|^2$, $f(x) = \sum_i x_i \log x_i$, and $f(x) = -\sum_i \log x_i$ respectively.

Bregman divergence satisfies following properties:

1. $D_f(x, y) \geq 0$, and equals 0 if and only if $x = y$;

2. $D_f(x, y)$ is strictly convex in its first argument, but in general not convex in the

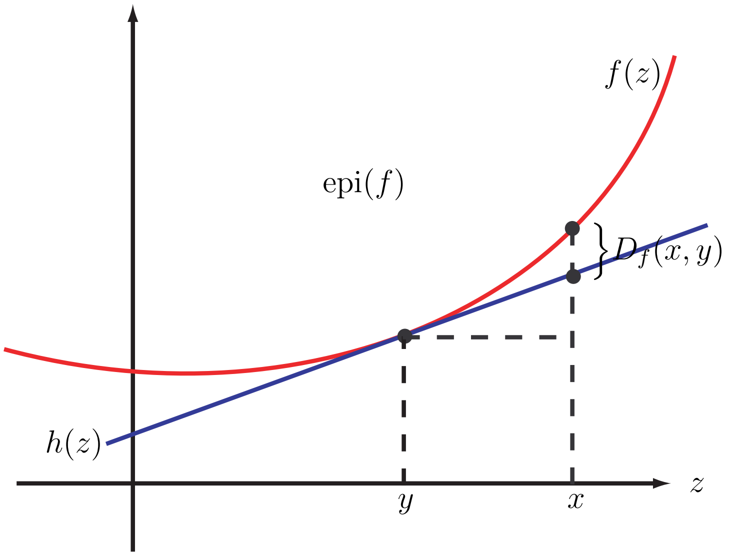


Figure 19: Geometric interpretation of Bregman divergence.

second argument;

3. $D_f(x,y)$ is usually not symmetric, i.e. $D_f(x,y) \neq D_f(y,x)$;

4. $D_f(x,y) = D_f(x,z) + D_f(z,y) - (x-z)^\top (\nabla f(y) - \nabla f(z))$.

The above definition of Bregman divergence applies to vectors. We can naturally extend this definition to real, symmetric $n \times n$ matrices, denoted by S^n. Given a strictly convex, differentiable function $f : S^n \to \mathbb{R}$, the Bregman matrix divergence is defined to be

$$D_f(X,Y) = f(X) - f(Y) - \operatorname{tr}\left(\nabla f(Y)^\top (X-Y)\right)$$

where $\operatorname{tr}(A)$ denotes the trace of a matrix A. An example of Bregman matrix divergence is the squared Frobenius norm $\|X-Y\|_F^2$, generated by function $f(X) = \|X\|_F^2$.

Bregman divergence over positive definite matrices (denoted by S_+^n) can be constructed from f which is a function of eigenvalues of a positive definite matrix. Specifically, let $\lambda_1, \ldots, \lambda_n$ be eigenvalues of X and $f(X) = \sum_i (\lambda_i \log \lambda_i - \lambda_i)$ with $0 \log 0 \overset{\text{def}}{=} 0$. The function $f(X)$ may be expressed as $\operatorname{tr}(X \log X - X)$ where $\log X$ denotes the matrix logarithm. This results in the von Neumann divergence

$$D_{vN}(X, Y) = \operatorname{tr}(X \log X - X \log Y - X + Y)$$

which is also called quantum relative entropy in quantum information theory. Another important matrix divergence, called the LogDet divergence

$$D_{ld}(X, Y) = \operatorname{tr}(XY^{-1}) - \log\det(XY^{-1}) - n$$

is generated by function $f(X) = -\sum_i \log \lambda_i$, or equivalently $f(X) = -\log\det X$.

Given a closed convex set $\Omega \subseteq \mathbb{R}^n$ such that $\Omega \cap \bar{S} \neq \varnothing$, and for $y \in S$, Bregman projection finds a point $x \in \Omega \cap \bar{S}$ such that

$$D_f(x, y) = \min_{z \in \Omega \cap \bar{S}} D_f(z, y).$$

The point x is denoted by $\mathcal{P}_\Omega(y)$ and is called a Bregman projection of the point y onto the set Ω. Projection x exists and is unique.

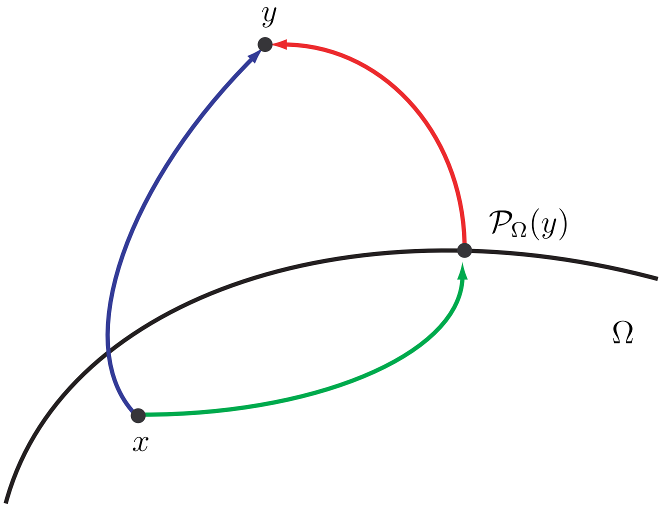


Figure 20: Geometric description for Theorem C.0.2.

Bregman projection satisfies the generalized Pythagorean theorem which can be stated as follows:

Theorem C.0.2. *Let Ω be a closed convex set such that $\Omega \cap \bar{S} \neq \varnothing$. Assume that $y \in S$ implies $\mathcal{P}_\Omega(y) \in S$. Let $x \in \Omega \cap \bar{S}$, then for any $y \in S$, the following inequality holds*

$$D_f(x, y) \geq D_f(x, \mathcal{P}_\Omega(y)) + D_f(\mathcal{P}_\Omega(y), y).$$

The theorem is illustrated in Figure 20.

APPENDIX D

QUADRATIC APPROXIMATION OF LOG-LIKELIHOOD FUNCTION FOR LOGISTIC AND PROBIT REGRESSION

Logistic regression and probit regression fall into the category of generalized linear model

$$f(y|x,\theta) = \Phi(y\theta^\top x)$$

where y takes values in $\{-1, 1\}$, θ is a vector of regression parameters and $\Phi(\cdot)$ is a link function. By choosing the link function $\Phi(\cdot)$ as the logistic function $\Phi(z) = \frac{e^z}{1+e^z}$, we get logistic regression. Similarly, probit regression is realized by the probit link function $\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$.

Given a set of data $\mathcal{D} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$, our goal is to approximate the log-likelihood $\log f(\mathcal{D}|\theta)$ as a quadratic function of the regression parameter θ

$$\begin{aligned}
\log f(\mathcal{D}|\theta) &= \sum_{i=1}^{m} \log \Phi\left(y^{(i)}\theta^\top x^{(i)}\right) \approx \sum_{i=1}^{m} a_i \left(y^{(i)}\theta^\top x^{(i)}\right)^2 + b_i \left(y^{(i)}\theta^\top x^{(i)}\right) + c_i \\
&= \theta^\top \left(\sum_{i=1}^{m} a_i x^{(i)} x^{(i)^\top}\right) \theta + \theta^\top \left(\sum_{i=1}^{m} b_i y^{(i)} x^{(i)}\right) + \sum_{i=1}^{m} c_i \\
&\overset{\text{def}}{=} -\frac{1}{2}\theta^\top \mathsf{A}\theta + \theta^\top \mathsf{b} + \mathsf{c}.
\end{aligned}$$

To compute a_i, b_i and c_i, we use Taylor expansion. For simplicity, let $z = y\theta^\top x$ and

$z_0 = y\theta_0^\top x$ where θ_0 is fixed, we have

$$\begin{aligned}
\log\Phi(z) \approx & \log\Phi(z_0) + \frac{\Phi'(z_0)}{\Phi(z_0)}(z-z_0) + \frac{1}{2}\left(\frac{\Phi''(z_0)}{\Phi(z_0)} - \left(\frac{\Phi'(z_0)}{\Phi(z_0)}\right)^2\right)(z-z_0)^2 \\
= & \underbrace{\frac{1}{2}\left(\frac{\Phi''(z_0)}{\Phi(z_0)} - \left(\frac{\Phi'(z_0)}{\Phi(z_0)}\right)^2\right)}_{a_i} z^2 \quad + \\
& \underbrace{\left(\frac{\Phi'(z_0)}{\Phi(z_0)} - \left(\frac{\Phi''(z_0)}{\Phi(z_0)} - \left(\frac{\Phi'(z_0)}{\Phi(z_0)}\right)^2\right) z_0\right)}_{b_i} z \quad + \\
& \underbrace{\log\Phi(z_0) + \frac{1}{2}\left(\frac{\Phi''(z_0)}{\Phi(z_0)} - \left(\frac{\Phi'(z_0)}{\Phi(z_0)}\right)^2\right) z_0^2\,.}_{c_i}
\end{aligned}$$

For logistic link function $\Phi(z) = \frac{e^z}{1+e^z}$, its first and second derivatives are computed to be

$$\begin{aligned}
\Phi'(z) &= \frac{e^z}{(1+e^z)^2} \\
\Phi''(z) &= \frac{e^z(1-e^z)}{(1+e^z)^3}.
\end{aligned}$$

For probit link function $\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$, we get

$$\begin{aligned}
\Phi'(z) &= \frac{1}{\sqrt{2\pi}} e^{-z^2/2} \\
\Phi''(z) &= \frac{-z}{\sqrt{2\pi}} e^{-z^2/2}.
\end{aligned}$$

In practice, θ_0 can be chosen as $\theta_0 = \arg\max_\theta \log f(\mathcal{D}|\theta) + \frac{\lambda}{2}\|\theta\|^2$, which is the maximum a posteriori estimate of θ with a prior distribution of θ given by $\mathcal{N}(0, \frac{1}{\lambda})$.

APPENDIX E

BREGMAN PROJECTION FOR LOGDET DIVERGENCE WITH RANK TWO CONSTRAINT MATRIX

We consider the following problem

$$\begin{aligned} \min_{X} \quad & \mathrm{tr}(XX_0^{-1}) - \log\det(XX_0^{-1}) - n \\ \text{s.t.} \quad & \mathrm{tr}(XA) \leq 0 \end{aligned} \tag{67}$$

where X_0 is an $n \times n$ positive semi-definite matrix, and $A = uv^\top + vu^\top$ for some $u, v \in \mathbb{R}^n$. Note, the objective function of (67) is the LogDet divergence between matrices X and X_0, and the rank of the matrix A is at most 2. Problem (67) is convex, therefore a local minimum of (67) is also a global minimum.

To solve for (67), we consider the Lagrange function

$$\mathcal{L}(X, \alpha) = \mathrm{tr}(XX_0^{-1}) - \log\det(XX_0^{-1}) - n + \alpha\mathrm{tr}(XA)$$

where the Lagrange multiplier α is constrained to be a non-negative number. At optimum X, the derivative of $\mathcal{L}$ with respect to X is zero, which results in

$$\left(X^\top\right)^{-1} = X_0^{-1} + \alpha A.$$

Since the inverse of a symmetric matrix is still symmetric, we have $X = X^\top$ and the above update rule is simplified to

$$X^{-1} = X_0^{-1} + \alpha\left(uv^\top + vu^\top\right). \tag{68}$$

The Karush-Kuhn-Tucker (KKT) conditions state that the following holds at the optimum (X, α): (i) $\alpha\mathrm{tr}(XA) = 0$; (ii) $\alpha \geq 0$. This implies that if $\alpha > 0$, then

$\mathrm{tr}(XA) = 0$. Note

$$\mathrm{tr}(XA) = \mathrm{tr}(Xuv^\top) + \mathrm{tr}(Xvu^\top) = v^\top Xu + u^\top Xv.$$

In the following, we derive a closed form solution of α based on these observations.

Let $Y = X_0^{-1} + \alpha uv^\top$, we have

$$\begin{aligned} X &= \left(X_0^{-1} + \alpha uv^\top + \alpha vu^\top\right)^{-1} = \left(Y + \alpha vu^\top\right)^{-1} \\ &= Y^{-1} - \frac{\alpha}{1 + \alpha u^\top Y^{-1} v} Y^{-1} vu^\top Y^{-1} \end{aligned} \tag{69}$$

where the second equality is obtained by applying the Sherman-Morrison inverse formula [53]:

$$(A + uv^\top)^{-1} = A^{-1} - \frac{1}{1 + v^\top A^{-1} u} A^{-1} uv^\top A^{-1}. \tag{70}$$

To simplify the computation, let

$$\begin{aligned} a &= u^\top X_0 u \\ b &= v^\top X_0 v \\ c &= u^\top X_0 v = v^\top X_0 u \\ \beta &= \frac{\alpha}{1 + \alpha c}, \end{aligned}$$

we have

$$\begin{aligned} u^\top Y^{-1} u &= u^\top \left(X_0 - \frac{\alpha}{1 + \alpha v^\top X_0 u} X_0 uv^\top X_0\right) u = u^\top \left(X_0 - \frac{\alpha}{1 + \alpha c} X_0 uv^\top X_0\right) u \\ &= u^\top X_0 u - \beta u^\top X_0 uv^\top X_0 u = a - \beta ac \end{aligned}$$

where the first equality is obtained by applying the Sherman-Morrison formula to Y^{-1}. Similarly,

$$\begin{aligned} v^\top Y^{-1} v &= b - \beta bc \\ u^\top Y^{-1} v &= c - \beta ab \\ v^\top Y^{-1} u &= c - \beta c^2. \end{aligned}$$

Since

$$
\begin{aligned}
u^\top X v &= u^\top \left(Y^{-1} - \frac{\alpha}{1+\alpha(c-\beta ab)} Y^{-1} v u^\top Y^{-1} \right) v \\
&= u^\top Y^{-1} v - \frac{1}{\frac{1}{\beta} - \beta ab} u^\top Y^{-1} v u^\top Y^{-1} v \\
&= c - \beta ab - \frac{1}{\frac{1}{\beta} - \beta ab} \left(c - \beta ab\right)^2 \\
v^\top X u &= v^\top \left(Y^{-1} - \frac{\alpha}{1+\alpha(c-\beta ab)} Y^{-1} v u^\top Y^{-1} \right) u \\
&= v^\top Y^{-1} u - \frac{1}{\frac{1}{\beta} - \beta ab} v^\top Y^{-1} v u^\top Y^{-1} u \\
&= c - \beta c^2 - \frac{1}{\frac{1}{\beta} - \beta ab} \left(b - \beta bc\right) \left(a - \beta ac\right),
\end{aligned}
$$

the value of β must satisfy the following equation

$$
u^\top X v + v^\top X u = 2c - \beta(c^2 + ab) - \frac{1}{\frac{1}{\beta} - \beta ab} \big((c - \beta ab)^2 + (b - \beta bc)(a - \beta ac) \big) = 0
$$

which is simplified to a quadratic equation

$$
\beta^2 abc - \beta(ab + c^2) + c = 0.
$$

The roots are given by the quadratic formula

$$
\beta = \frac{ab + c^2 \pm \sqrt{(ab+c^2)^2 - 4abc^2}}{2abc} = \frac{ab + c^2 \pm |ab - c^2|}{2abc}.
$$

For $\alpha > 0$, we have

$$
\beta = \frac{\alpha}{1 + \alpha c} = \frac{1}{\frac{1}{\alpha} + c} < \frac{1}{c},
$$

therefore β has a unique solution which is $\frac{c}{ab}$. Solving for α, we get

$$
\alpha = \frac{c}{ab - c^2}.
$$

The optimum X is computed as

$$
\begin{aligned}
X &= Y^{-1} - \frac{1}{\frac{1+\alpha c}{\alpha} - \frac{\alpha}{1+\alpha c} ab} Y^{-1} v u^\top Y^{-1} \\
&= X_0 - X_0 \Lambda X_0
\end{aligned}
$$

where

$$\Lambda = \frac{1}{\frac{1+\alpha c}{\alpha} - \frac{\alpha}{1+\alpha c}ab} \left(vu^\top + uv^\top - \frac{\alpha b}{1+\alpha c}uu^\top - \frac{\alpha a}{1+\alpha c}vv^\top \right)$$

and $\alpha = \max\{0, \frac{c}{ab-c^2}\}$.

REFERENCES

[1] Bakker, B. and Heskes, T., "Task clustering and gating for bayesian multitask learning," *Journal of Machine Learning Research*, vol. 4, pp. 83–99, 2003.

[2] Barlow, R. E., Bartholomew, D., Bremner, J. M., and Brunk, H. D., *Statistical inference under order restrictions (the theory and application of isotonic regression)*. John Wiley and Sons, Inc., 1972.

[3] Baxter, J., "A bayesian/information theoretic model of learning to learn via multiple task sampling," *Machine Learning*, vol. 28, no. 1, pp. 7–39, 1997.

[4] Berger, J. O., *Statistical Decision Theory and Bayesian Analysis*. Springer, 1985.

[5] Bi, J. and Zhang, T., "Support vector classification with input data uncertainty," in *Advances in Neural Information Processing Systems 17*, pp. 161–168, 2005.

[6] Bishop, C. M., *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] Blei, D., Ng, A., and Jordan, M., "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[8] Boyd, S. and Vandenberghe, L., *Convex Optimization*. Cambridge University Press, 2004.

[9] Brants, T. and Franz, A., "Web 1T 5-gram Version 1," 2006.

[10] Bregman, L. M., "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, pp. 200–217, 1967.

[11] Budanitsky, A. and Hirst, G., "Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures," in *NAACL Workshop on WordNet and other Lexical Resources*, 2001.

[12] Burges, C., "Dimension reduction: A guided tour," Tech. Rep. MSR-TR-2009-2013, Microsoft Research, 2009.

[13] Censor, Y. and Zenios, S. A., *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, 1998.

[14] Chang, J., Boyd-Graber, J., Wang, C., Gerrish, S., and Blei, D. M., "Reading tea leaves: How humans interpret topic models," in *Neural Information Processing Systems*, 2009.

[15] Chang, M.-W., Ratinov, L., and Roth, D., "Guiding semi-supervision with constraint-driven learning," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 280–287, 2007.

[16] Chang, M.-W., Srikumar, V., Goldwasser, D., and Roth, D., "Structured output learning with indirect supervision," in *Proc. of the International Conference on Machine Learning*, 2010.

[17] Chechik, G., Sharma, V., Shalit, U., and Bengio, S., "Large scale online learning of image similarity through ranking," *Journal of Machine Learning Research*, vol. 11, pp. 1109–1135, 2010.

[18] Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S., "Identifying sources of opinions with conditional random fields and extraction patterns," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005.

[19] Collins, M., "Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, 2002.

[20] Collins-Thompson, K. and Callan, J., "Predicting reading difficulty with statistical language models," *Journal of the American Society for Information Science and Technology*, vol. 56, no. 13, pp. 598–605, 2005.

[21] Collins-Thompson, K., "Reducing the risk of query expansion via robust constrained optimization," in *Proceeding of the 18th ACM conference on Information and knowledge management*, pp. 837–846, 2009.

[22] Cover, T. M. and Thomas, J. A., *Elements of Information Theory.* John Wiley & Sons, Inc., second ed., 2006.

[23] Cramer, H., *Mathematical Methods of Statistics.* Princeton University Press, 1957.

[24] Crammer, K., Dredze, M., and Pereira, F., "Exact convex confidence-weighted learning," in *Advances in Neural Information Processing Systems 21*, 2009.

[25] Crammer, K., Dredze, M., and Kulesza, A., "Multi-class confidence weighted algorithms," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 496–504, 2009.

[26] Crammer, K., Mohri, M., and Pereira, F., "Gaussian margin machines," in *Twelfth International Conference on Artificial Intelligence and Statistics*, pp. 105–112, 2009.

[27] Daumé III, H., Langford, J., and Marcu, D., "Search-based structured prediction," 2009.

[28] Davidson, I., "Knowledge driven dimension reduction for clustering," in *International Joint Conference on Artificial Intelligence*, 2009.

[29] Davies, D. L. and Bouldin, D. W., "A cluster separation measure.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 4, pp. 224–227, 2000.

[30] Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S., "Information-theoretic metric learning," in *Proceedings of the 24th international conference on Machine learning*, pp. 209–216, 2007.

[31] Dekel, O. and Shamir, O., "Good learners for evil teachers," in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 233–240, 2009.

[32] Dekel, O. and Shamir, O., "Vox populi: Collecting high-quality labels from a crowd," in *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

[33] Dillon, J., Mao, Y., Lebanon, G., and Zhang, J., "Statistical translation, heat kernels, and expected distances," in *Uncertainty in Artificial Intelligence*, pp. 93–100, AUAI Press, 2007.

[34] Dredze, M., Crammer, K., and Pereira, F., "Confidence-weighted linear classification," in *International Conference on Machine Learning*, 2008.

[35] Dredze, M., Kulesza, A., and Crammer, K., "Multi-domain learning by confidence-weighted parameter combination," *Maching Learning*, vol. 79, no. 1-2, pp. 123–149, 2010.

[36] Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern classification.* Wiley, 2001.

[37] Epshteyn, A. and DeJong, G., "Generative prior knowledge for discriminative classification," *Journal of Artificial Intelligence Research*, vol. 27, pp. 25–53, 2006.

[38] Fei-Fei, L., Fergus, R., and Perona, P., "A bayesian approach to unsupervised one-shot learning of object categories," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pp. 1134–1141, 2003.

[39] Fink, M. and Perona, P., "Mutual boosting for contextual inference," in *Advances in neural information processing systems*, 2004.

[40] Fung, G., Mangasarian, O. L., and Shavlik, J., "Knowledge-based support vector machine classifiers," in *In Advances in Neural Information Processing Systems 14*, pp. 01–09, 2002.

[41] Garthwaite, P. H., Kadane, J., and O'Hagan, A., "Statistical methods for eliciting probability distributions," *Journal of the American Statistical Association*, vol. 100, pp. 680–701, 2005.

[42] Golub, G. H. and van Loan, C. F., *Matrix Computations.* The Johns Hopkins University Press, 1996.

[43] Graça, J., Ganchev, K., and Taskar, B., "Expectation maximization and posterior constraints," in *Advances in neural information processing systems 20*, 2008.

[44] Gupta, A. K. and Nagar, D. K., *Matrix Variate Distributions.* Chapman and Hall/CRC, 1999.

[45] Haghighi, A. and Klein, D., "Prototype-driven learning for sequence models," in *Proceedings of the Human Language Technology Conference of the NAACL*, pp. 320–327, 2006.

[46] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning.* Springer-Verlag, New York, 2001.

[47] Havre, S., Hetzler, E., Whitney, P., and Nowell, L., "Themeriver: Visualizing thematic changes in large document collections," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 9–20, 2002.

[48] Hearst, M. A., "Texttiling: Segmenting text into multi-paragraph subtopic passages," *Computational Linguistics*, vol. 23, no. 1, pp. 33–64, 1997.

[49] Heilman, M., Collins-Thompson, K., Callan, J., and Eskenazi, M., "Combining lexical and grammatical features to improve readability measures for first and second language texts," in *Proceedings of the Human Language Technology Conference*, 2007.

[50] Heilman, M., Collins-Thompson, K., and Eskenazi, M., "An analysis of statistical models and features for reading difficulty prediction," in *The 3rd Workshop on Innovative Use of NLP for Building Educational Applications*, 2008.

[51] Heskes, T., "Empirical bayes for learning to learn," in *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 367–374, 2000.

[52] Hirotsu, C., "Ordered alternatives for interaction effects," *Biometrika*, vol. 65, no. 3, pp. 561–570, 1978.

[53] Horn, R. A. and Johnson, C. R., *Matrix Analysis.* Cambridge University Press, 1990.

[54] Jiang, J. J. and Conrath, D. W., "Semantic similarity based on corpus statistics and lexical taxonomy," in *International Conference Research on Computational Linguistics (ROCLING X)*, 1997.

[55] Jin, R. and Liu, Y., "A framework for incorporating class priors into discriminative classification," in *The Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 568–577, 2005.

[56] Joachims, T., "Optimizing search engines using clickthrough data," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 133–142, 2002.

[57] Jurafsky, D. and Martin, J. H., *Speech and Language Processing.* Prentice Hall, 2008.

[58] Lafferty, J. and Lebanon, G., "Diffusion kernels on statistical manifolds," *Journal of Machine Learning Research*, vol. 6, pp. 129–163, 2005.

[59] Lafferty, J., Pereira, F., and McCallum, A., "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proc. of the International Conference on Machine Learning*, 2001.

[60] Lanckriet, G. R., Ghaoui, L. E., Bhattacharyya, C., and Jordan, M. I., "A robust minimax approach to classification," *Journal of Machine Learning Research*, vol. 3, pp. 555–582, 2003.

[61] Lawrence, N. D. and Platt, J. C., "Learning to learn with the informative vector machine," in *Proceedings of the twenty-first international conference on Machine learning*, 2004.

[62] Lebanon, G., "Metric learning for text documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 497–508, 2006.

[63] Lee, S.-I., Chatalbashev, V., Vickrey, D., and Koller, D., "Learning a meta-level prior for feature relevance from multiple related tasks," in *Proceedings of the 24th international conference on Machine learning*, pp. 489–496, 2007.

[64] Lewis, D., Yang, Y., Rose, T., and Li, F., "RCV1: A new benchmark collection for text categorization research," *Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[65] Lugosi, G., "Learning with an unreliable teacher," *Pattern Recognition*, vol. 25, no. 1, pp. 79–87, 1992.

[66] Mann, G. S. and McCallum, A., "Simple, robust, scalable semi-supervised learning via expectation regularization," in *Proceedings of the 24th international conference on Machine learning*, pp. 593–600, 2007.

[67] Mao, Y., Balasubramanian, K., and Lebanon, G., "Dimensionality reduction for text using domain knowledge," in *Proc. of The 23rd International Conference on Computational Linguistics (COLING)*, 2010 (acceptance rate 42%).

[68] Mao, Y., Dillon, J., and Lebanon, G., "Sequential document visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1208–1215, 2007.

[69] Mao, Y. and Lebanon, G., "Isotonic conditional random fields and local sentiment flow," in *Advances in Neural Information Processing Systems 19*, pp. 961–968, 2007.

[70] Mao, Y. and Lebanon, G., "Domain knowledge uncertainty and probablistic parameter constraints," in *Proc. of the 25th Conference on Uncertainty in Artificial Intelligence*, AUAI Press, 2009.

[71] Mao, Y. and Lebanon, G., "Isotonic conditional random fields and local sentiment flow," *Machine Learning*, vol. 77, no. 2-3, pp. 225–248, 2009.

[72] Marx, Z., Rosenstein, M. T., Kaelbling, L. P., and Dietterich, T. G., "Transfer learning with an ensemble of background tasks," in *NIPS 2005 Workshop on Transfer Learning*, 2005.

[73] McCallum, A., Freitag, D., and Pereira, F., "Maximum entropy Markov models for information extraction and segmentation," in *Proc. 17th International Conference on Machine Learning*, pp. 591–598, 2000.

[74] McCallum, A., Mann, G., and Druck, G., "Generalized expectation criteria," Tech. Rep. 2007-60, University of Massachusetts Amherst, 2007.

[75] McDonald, R., Hannan, K., Neylon, T., Wells, M., and Reynar, J., "Structured models for fine-to-coarse sentiment analysis," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007.

[76] Murphy, K. P., *Dynamic Bayesian Networks: Representation, Inference and Learning.* PhD thesis, UC Berkeley, 2002.

[77] Niculescu, R. S., Mitchell, T. M., and Rao, R. B., "Bayesian network learning with parameter constraints," *Journal of Machine Learning Research*, vol. 7, pp. 1357–1383, 2006.

[78] O'Hagan, A., Buck, C. E., Daneshkhah, A., Eiser, J. R., Garthwaite, P. H., Jenkinson, D. J., Oakley, J. E., and Rakow, T., *Uncertain Judgements: Eliciting Experts' Probabilities.* Wiley, 2006.

[79] Paley, W. B., "TextArc: Showing word frequency and distribution in text," in *IEEE Symposium on Information Visualization Poster Compendium*, 2002.

[80] Pang, B. and Lee, L., "A sentimental eduction: sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. of the Association of Computational Linguistics*, 2004.

[81] Pang, B. and Lee, L., "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005.

[82] Pang, B., Lee, L., and Vaithyanathan, S., "Thumbs up?: sentiment classification using machine learning techniques," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2002.

[83] Poon, H. and Domingos, P., "Joint inference in information extraction," in *Proceedings of the 22nd national conference on Artificial intelligence*, pp. 913–918, 2007.

[84] Provost, S. B. and Cheong, Y.-H., "On the distribution of linear combinations of the components of a dirichlet random vector," *The Canadian Journal of Statistics*, vol. 28, 2000.

[85] Rabiner, L. R., "A tutorial on hidden markov models and selected applications in speech recognition," pp. 267–296, 1990.

[86] Raina, R., Ng, A., and Koller, D., "Constructing informative priors using transfer learning," in *Proceedings of the 23rd international conference on Machine learning*, 2006.

[87] Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., and Moy, L., "Learning from crowds," *Journal of Machine Learning Research*, vol. 11, pp. 1297–1322, 2010.

[88] Richardson, M. and Domingos, P., "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.

[89] Richardson, M., Prakash, A., and Brill, E., "Beyond pagerank: machine learning for static ranking," in *Proceedings of the 15th international conference on World Wide Web*, pp. 707–715, 2006.

[90] Roberts, H. V., "Probabilistic prediction," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 50–62, 1965.

[91] Roth, D. and Yih, W.-t., "Integer linear programming inference for conditional random fields," in *Proceedings of the 22nd international conference on Machine learning*, pp. 736–743, 2005.

[92] Roweis, S. and Saul, L., "Nonlinear dimensionality reduction by locally linear embedding.," *Science*, vol. 290, pp. 2323–2326, 2000.

[93] Schapire, R., Rochery, M., Rahim, M., and Gupta, N., "Incorporating prior knowledge into boosting," in *Proceedings of the 19th International Conference on Machine Learning*, 2002.

[94] Schutz, M. and Joachims, T., "Learning a distance metric from relative comparisons," in *Advances in Neural Information Processing Systems*, 2003.

[95] Schwarm, S. E. and Ostendorf, M., "Reading level assessment using support vector machines and statistical language models," in *Proc. of the Association of Computational Linguistics*, 2005.

[96] Shalev-Shwartz, S., Singer, Y., and Ng, A. Y., "Online and batch learning of pseudo-metrics," in *Proceedings of the twenty-first international conference on Machine learning*, 2004.

[97] Sheng, V. S., Provost, F., and Ipeirotis, P. G., "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 614–622, 2008.

[98] Shivaswamy, P. K., Bhattacharyya, C., Bhattacharyya, C., and Smola, A. J., "Second order cone programming approaches for handling missing and uncertain data," *Journal of Machine Learning Research*, vol. 7, pp. 1283–1314, 2006.

[99] Silvapulle, M. J. and Sen, P. K., *Constrained Statistical Inference: Order, Inequality, and Shape Constraints.* Wiley, 2004.

[100] Silverman, B. W., "Some asymptotic properties of the probabilistic teacher," *IEEE Transactions on Information Theory*, vol. 26, pp. 246–249, 1980.

[101] Smyth, P., *Learning with Probabilistic Supervision*, ch. 9. MIT Press, 1995.

[102] Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y., "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 254–263, 2008.

[103] Spoerri, A., "InfoCrystal: A visual tool for information retrieval," in *Proceedings of the 4th Conference on Visualization*, 1993.

[104] Stanley, R. P., *Enumerative Combinatorics*, vol. 1. Cambridge University Press, 2000.

[105] Stenner, A. J., *Measuring reading comprehension with the Lexile Framework*. Durham, NC: Metametrics, Inc., 1996.

[106] Taskar, B., Guestrin, C., and Koller, D., "Max-margin markov networks," in *Advances in neural information processing systems*, 2004.

[107] Teh, Y., Seeger, M., and Jordan, M., "Semiparametric latent factor models," in *Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

[108] Thomas, J. J. and Cook, K. A., eds., *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society, 2005.

[109] Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K., "Sparsity and smoothness via the fused lasso," *Journal Of The Royal Statistical Society Series B*, vol. 67, no. 1, pp. 91–108, 2005.

[110] Tong, S., *Active Learning: Theory and Applications*. PhD thesis, 2001.

[111] Torralba, A., Murphy, K. P., and Freeman, W. T., "Contextual models for object detection using boosted random fields," in *Advances in Neural Information Processing Systems 17*, pp. 1401–1408, 2005.

[112] van der Maaten, L. and Hinton, G., "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[113] Wu, X. and Srihari, R., "Incorporating prior knowledge with weighted margin support vector machines," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 326–333, 2004.

[114] Xing, E., Ng, A., Jordan, M., and Russel, S., "Distance metric learning with applications to clustering with side information," in *Advances in Neural Information Processing Systems*, 2003.

[115] Yu, K., Tresp, V., and Schwaighofer, A., "Learning gaussian processes from multiple tasks," in *Proceedings of the 22nd international conference on Machine learning*, pp. 1012–1019, 2005.

[116] Zhang, Y., "Using bayesian priors to combine classifiers for adaptive filtering," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004.