Building Maximum Entropy Text Classifier Using Semi-supervised Learning

Zhang Xinhua

HT031518L

Email: zhangxi2@comp.nus.edu.sg

Supervisor: A/P Lee Wee Sun

Submitted as PhD Qualifying Examination Term Paper

SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE
October 2004

Abstract

Over the recent years, text classification has become one of the key techniques for organizing information. Since hand-coding text classifiers is impractical and hand-labeling text is time and labor consuming, it is preferable to learn classifiers from a small amount of labeled examples and a large example of unlabeled data. In many cases, such as online information retrieval or database applications, such unlabeled data are easily and abundantly available.

Although a lot of this kind of learning algorithms have been designed, most of them rely on certain assumptions, which are dependent on specific datasets. Consequently, the lack of generality makes these algorithms unstable across different datasets. Therefore, we favor an algorithm with as little dependence on such assumptions or as weak assumption as possible.

The maximum entropy models (MaxEnt) offers a generic framework meeting this requirement. Built upon a set of features which is equivalent to undirected graphical models, it provides a natural leverage of feature selection. Most importantly, the only assumption made by MaxEnt is that the average feature values on labeled data give a reasonable (not too deviated) estimation of those values on both labeled *and unlabeled* data. This is a very weak statistical assumption, underpinning the generality of MaxEnt. Even if the assumption is not strictly satisfied in some situations, theoretical bounds are derived on the generalization error. Similar to soft-max regression, MaxEnt also employs a straightforward mechanism for multi-class classification. With its standard form equivalent to maximum likelihood estimation, there are numerous smoothing approaches proposed to overcome overfitting. Several algorithms for solving this convex optimization problem are also proposed, with different performance and constraints.

The main focus of the project is how to incorporate unlabeled data into MaxEnt. As the standard MaxEnt does not perform satisfactorily, side information is considered to be added to MaxEnt, in forms of minimal spanning tree, *k*-nearest neighbor, multi-representation of examples, etc. Our initial experimental result suggests that MaxEnt with side information is a promising tool. Other promising research areas under this framework are also pointed out in this report, after a comprehensive survey on both learning with unlabeled data and MaxEnt.

Contents

Chapter 1 Introduction

1.1 Motivation and background

Suppose we work for a web site that maintains a public listing of second-hand books from many different companies or individuals. A user of the web site might find books by browsing all books in a specific category. However, these books are spidered from the Web, and do not come with any category label. Instead of reading description of each book to manually determine the label, it would be helpful to have a system that automatically examines the text and makes the decision itself. This automatic process is called *text classification*. In general, text classification systems categorize documents into one (or several) of a set of pre-defined topics of interest. Text classification is of great practical importance today given the massive volume of online text available, such as electronic text from the World Wide Web, electronic mail, corporate databases, chat rooms, and digital libraries. By automatically populating and maintaining these taxonomies, we can aid people in their search for knowledge and information. Other applications of text classification involves: cataloging news articles (Lewis & Gale, 1994; Joachims, 1998b); classifying web pages into a symbolic ontology (Craven et al., 2000); finding a person's homepage (Shavlik & Eliassi-Rad, 1998); automatically learning the reading interests of users (Lang, 1995; Pazzani et al., 1996); automatically threading and filtering email by content (Lewis & Knowles, 1997; Sahami et al., 1998); and book recommendation (Mooney & Roy, 2000).

How are automatic text classifiers built? Early attempts were based on the manual construction of rule sets, but at significant cost. A more efficient approach is to use supervised learning to construct a classifier. Here, we provide an algorithm with an example set of documents for each class, and allow it to find a representation or decision rule for classifying future documents. This approach also gives high-accuracy classifiers, and is significantly less expensive than manual construction because the algorithm automatically constructs the decision rule itself. Supervised text classification algorithms have been successfully used in a wide variety of practical domains, almost in all above mentioned applications.

However, the supervised learning approach is not as effortless as we might hope. One key difficulty with these algorithms is that they require a large, often prohibitive, number of labeled training examples to learn accurately. Labeling must typically be done by a person. This is a painfully time-consuming process. Take, for example, the task of learning which newsgroup articles are of interest to a particular person reading UseNet news. Work by (Lang, 1995) found that after a person read and hand-labeled about 1000 articles, a learned classifier achieved a precision of about 50% when making predictions for only the top 10% of documents about which it was most confident. Most users of a practical system, however, would not have the patience to label a thousand articles, especially to obtain only this level of precision. One would obviously prefer algorithms that can provide accurate classifications after hand-labeling only a dozen articles, rather than thousands. This need for large quantities of expensive labeled examples raises an important question: what other sources of information can reduce the need for labeled data?

One goal of this research project is to demonstrate that a new type of high-accuracy classifiers can be created with a small number of labeled examples and a large number of *unlabeled* *examples*, which we call *semi-supervised learning*. In general, unlabeled examples are much less expensive and easier to come by than labeled examples. This is particularly true for text classification tasks involving online data sources, such as web pages, email, and news stories, where huge amounts of unlabeled text are readily available. Collecting this text can frequently be done automatically, so it is feasible to quickly gather a large set of unlabeled examples. If unlabeled data can be integrated into supervised learning, then building text classification systems will be significantly faster and less expensive than before.

1.2 Why do unlabeled data help?

At first glance, it might seem that nothing is to be gained from unlabeled data. After all, an unlabeled document does not contain the most important piece of information — its class. But thinking another way may reveal the value of unlabeled data. In the field of information retrieval, it is well known that words in natural language occur in strong co-occurrrence patterns (van Rijsbergen, 1977). Some words are likely to occur together in one document, others are not. For example, when asking search engine *Google* about all web pages containing the words *sugar* and *sauce*, it returns 1,390,000 results. When asking for the documents with the words *sugar* and *math*, we get only 191,000 results, though *math* is a more popular word on the web than *sauce*. Suppose we are interested in recognizing web pages about cuisine. We are given just a few known cuisine and non-cuisine web pages, along with a large number of web pages that are unlabeled. By looking at just the labeled data we determine that pages containing the word *sugar* tend to be about cuisine. If we use this fact to estimate the classification of many unlabeled web pages, we might find that the word *sauce* occurs frequently in the unlabeled examples that are now believed to belong to the positive class. This co-occurrence of the words *sugar* and *sauce* over the large set of unlabeled training data can provide useful information to construct a more accurate classifier that considers both *sugar* and *sauce* as indicators of positive examples.

In this project, we show how unlabeled data can be used to increase classification accuracy, especially when labeled data are scarce. Formally, unlabeled data provide us with knowledge only of the distribution of examples in feature space. In the most general case, distributional knowledge will not provide helpful information to supervised learning. Consider classifying uniformly distributed instances based on conjunctions of literals. Here there is no relationship between the uniform instance distribution and the space of possible classification tasks thus clearly, unlabeled data can not help.

We need to introduce appropriate assumptions/biases into our learner by assuming some dependence between the instance distribution and the classification task. Even standard supervised learning with labeled data must introduce bias in order to learn. In a well-known and often-proven result in (Watanabe, 1969), the theorem of the Ugly Duckling shows that without bias all pairs of training examples look equally similar, and generalization into classes is impossible. This result was foreshadowed long ago by both William of Ockham's philosophy of radical nominalism in the 1300's and David Hume in the 1700's. Somewhat more recently, (Zhang & Oles, 2000) formalized and proved that supervised learning with unlabeled examples must assume a dependence between the instance distribution and the classification task. In this project, we assume the dependence can be captured by a maximum entropy model for text documents.

1.3 Why do we choose maximum entropy models?

In principle, our maximum entropy model aims to maximize the entropy defined on conditional probability distribution, i.e., making one example's probability of belonging to all possible classes as evenly as possible. The probability is modeled to depend only on features, a kind of sufficient statistics defined on input attributes, instead of directly using the latter. This technique naturally allows for feature induction and feature selection. Of course, constraints must be incorporated to fit the training data. Just as almost all existing semi-supervised learning algorithms depend on certain assumptions, here we introduce an important assumption that the expected value of features can be estimated from labeled data to a good level. Then we push our model to produce the same (or nearly same) average value on these features as the empirical average. In the standard form, this approach proves just a dual problem of maximum likelihood. But various smoothing techniques exist to overcome overfitting.

We know this assumption is not very realistic, and that it will not capture the subtleties of the document writing process, or inversely the process by which readers discriminate the documents. Nevertheless, these assumptions encode a relationship between the document distribution and the classification task in a way that allows unlabeled data to be incorporated into learning. Besides, these statistical assumptions are rather generic and weak, weaker than those that are commonly made over distance metric or similarity measures. Therefore, we expect that it will be less prone to the common algorithms' instability across different datasets, due to the differing assumptions these datasets are based on.

Maximum entropy models also serve as a flexible framework for incorporating various forms of side information, by means of modifying the constraints and objective function. Useful side information may come from two sources:

- Instance similarity. This idea is based on the similarity between examples. It includes (not restricted to) neighboring relationship between different instances, assuming nearby instances having similar classes (e.g., *k* nearest neighbor or minimal spanning tree); redundant description, assuming invariance in labeling under different descriptions (e.g., image and voice for identifying a person); or tracking the same object (e.g., multiple occurrence of a word in the same article for word sense disambiguation).
- Class similarity. This idea makes use of information on classification tasks that are likely to be related to each other, assuming that there is a subset of features behaving similarly across a subset of classes that are known to be similar. Examples include combining different datasets (different distributions) which are for the same classification task; hierarchical classes; or structured class relationships (such as trees or other generic graphic models) for applications like word sense disambiguation exploiting synonyms, hyponyms and hypernyms.

Although some forms of side information are rather based on problem specific assumptions which are what we wish to avoid, there is no reason not to utilize them if they work well and the maximum entropy model does offer such an opening.

1.4 Organization of the report

This report is composed of five chapters. The next chapter surveys the existing text classification algorithm, especially in the area of semi-supervised learning. After that, we describe the various types of maximum entropy models in Chapter 3, including basic theoretical results related to standard model, smoothing techniques, and parameter estimation algorithms. Then, we move on to the focus of our project: how to incorporate unlabeled data in to maximum entropy classifiers for text classification. A number of challenges and promising research areas are pointed out and initial results are presented demonstrating the promise of the model. Finally, the report is concluded in Chapter 5.

Chapter 2 Models using unlabeled data

Text classification is a field with rich existing and ongoing research. As a start of the project, a survey of the theoretical and empirical approaches used in this area is necessary for building a sound foundation. This chapter discusses the history and state of the art in related works, including standard supervised learning, and semi-supervised learning using unlabeled data.

2.1 Supervised Learning

The machine learning technique for text classification can be traced at least back to Naive Bayes (NB) (Mitchell, 1997; Lewis, 1998), an early approach which is still enjoying popularity nowadays. Besides its competitive performance, its strength lies mainly in the straightforward probabilistic nature, amenable to a variety of extensions. As a generative classifier, NB is often used as a multinomial model, i.e., a mixture of multinomials that tracks the number of times a word appears in a document without considering grammar or semantics (Lewis & Gale, 1994; Mitchell, 1997; Joachims, 1997; Li & Yamanishi, 1997; Lewis, 1998; McCallum & Nigam, 1998a; McCallum & Nigam, 1998b). This corresponds to the *unigram* model in natural language processing (NLP). Based on NB, TAN trees (Sahami, 1996) incorporated limited word dependencies in the model. (Li & Yamanishi, 1997) relaxed the one-to-one class-to-component correspondence by treating the problem as statistical hypothesis testing over finite mixture models based on soft clustering. A class hierarchy can be made use of through statistical shrinkage (McCallum & Nigam, 1998a) or other more ad-hoc techniques (Koller & Sahami, 1997).

Other than NB, a variety of machine learning techniques have been applied to text classification. Support vector machines (Dumais et al., 1998; Joachims, 1998a), built upon statistical learning theory (Vapnik, 1998), is a popular and promising technique that significantly outperformed some other models in some tasks. Other approaches have used maximum entropy (Nigam et al., 1999a), neural nets (Wiener et al., 1995; Shavlik & Eliassi-Rad, 1998) and several rule learning algorithms (Apte et al., 1994; Cohen & Singer, 1996; Moulinier et al., 1996; Craven et al., 1998). Still others have used instance-based lazy learning methods like *k*-nearest neighbor (*k*NN) (Yang & Chute, 1994; Cohen & Hirsch, 1998), and a variety of committee machine boosting approaches (Apte et al., 1998; Sebastiani et al., 2000; Schapire & Singer, 2000). Thus far, no specific technique has been proved as clearly better than the others, though some comparison works suggest that *k*NN and SVMs perform at least as well as other algorithms when the number of labeled data for each class of interest in large (Yang, 1999).

For the practical problem of document representation, most studies use the simple bags-of-words method, tracking the number of times each word w_i occurs in a document x (denoted as $TF(w_i, x)$), or even just whether or not it occurred (binary). More sophisticated models incorporate more substantial linguistic or semantic information and they have made at most modest improvements to accuracy. (Salton & Buckley, 1998) showed that scaling the dimensions of the feature vector with their inverse document frequency ($IDF(w_i)$) leads to an improved performance. $IDF(w_i)$ is defined as $log(n/DF(w_i))$ where n is the total number of documents, and $DF(w_i)$ is the number of documents the word w_i occurs in. (Furnkranz et al., 1998) uses shallow syntactic phrase patterns and finds some improvements to NB and rule learning algorithms. (Mladenic, 1998) selects phrases of variable length from web pages. (Rodriguez et al., 1997; Scott & Matwin, 1998) incorporated semantic network of the English language, WordNet.

2.2 Generative semi-supervised learning

Generative learning models the probability of generating a data example for each class. Given a data example, the posterior probability of its belonging to each class is then calculated using Bayes theorem. The machine learning process that combines labeled and unlabeled data is called semi-supervised learning. This idea is not new in the statistics community and it has been joined by the machine learning community for about 10 years. At least as early as 1968, it was suggested that labeled and unlabeled could be combined for building classifiers with likelihood maximization by testing all possible class assignments (Hartley & Rao, 1968; Day, 1969). (Day, 1969) presented an iterative EM-like approach for parameters of a mixture of two normal distributions with known covariances from unlabeled data alone. Similar iterative algorithms for building maximum likelihood (ML) classifiers from labeled and unlabeled data are primarily for mixtures of normal distributions (McLachlan, 1975; Titterington, 1976).

The seminal paper by (Dempster et al., 1977) presented the theory of the Expectation-Maximization (EM) framework, bringing together and formalizing many of the commonalities of previously suggested iterative techniques for likelihood maximization with missing data (or latent variables). It was immediately recognized that EM is applicable to estimating ML or maximum a posteriori (MAP) parameters for mixture models from labeled and unlabeled data (Murray & Titterington, 1978) and then using this for classification (Little, 1977). Since then, this approach continues to be used and studied (McLachlan & Ganesalingam, 1982; Ganesalingam, 1989; Shahshahani & Landgrebe, 1994). EM and its application to mixture modeling enjoy a splendid history, summarized in (McLachlan & Basford, 1988; McLachlan & Krishnan, 1997; McLachlan & Peel, 2000).

(Miller & Uyar, 1996; Nigam et al., 1998; Baluja, 1999) started using ML of mixture models to combine labeled and unlabeled data for classification. (Ghahramani & Jordan, 1994) used EM to fill in missing feature values of examples when learning from incomplete data by assuming a mixture model. Mixture models have also been used as a generative model for unsupervised clustering, whose parameters have been estimated with EM (Cheeseman et al., 1988; Cheeseman & Stutz, 1996; Hofmann & Puzicha, 1998). Hierarchical mixture-of-experts are similar to mixture models, and their parameters are also typically set with EM (Jordan & Jacobs, 1994). The first comprehensive work of semi-supervised generative model for text classification is (Nigam et al., 1999b; Nigam, 2001).

2.3 Discriminative semi-supervised learning

For a lot of real world problems, we can not model the input distribution with sufficient accuracy or the number of parameters to estimate for generative model is too large. Then a practical approach is to build classifier by directly calculating its probability of belonging to each class. This is called discriminative model. A transductive support vector machine (TSVM) (Vapnik, 1998) finds parameters for a linear separator with labeled training data and test data. It is easily extended to training data with unlabeled examples. At a high level, they work by finding the linear separator between the labeled examples of each class that maximizes the margin over both the labeled and unlabeled examples. (Joachims, 1999) demonstrated the efficacy of this approach for several text classification tasks. (Bennett & Demiriz, 1999) proposed a computationally easier variant of TSVM and found small improvements on some datasets. By intuition, TSVMs assume that unlabeled examples help find low-density regions of instance space where decision boundaries between classes lie in. This is an important assumption called ‘cluster assumption’ because it is equivalent to stating that two points are likely to have the same class if there is a path connecting them passing through regions of high density only. For discriminative framework, most semi-supervised learning algorithms implicitly or explicitly make this assumption. However, (Zhang & Oles, 2000) argues that TSVMs are asymptotically unlikely to be helpful for classification in general, both experimentally and theoretically (via Fisher's information matrices) because such assumption may be violated.

Some works by Jaakkola through the years are related to this project. (Jaakkola et al., 2000) proposed a maximum entropy discrimination model based on maximum margin. It is formulated as an optimization problem, with objective function being the maximum entropy distribution over classifier parameters, and the constraints being that the distance (margin) between each labeled example and decision boundary should be no less than a fixed value. Of course, soft penalties can be adopted. An interesting feature is that unlabeled data are also covered by margin constraints in order to commit to the class of unlabeled examples during parameter estimation by entropy maximization. The contribution made by unlabeled data is to provide a more accurate distribution over the unknown class labels. Classification is then performed in a Bayesian manner, combining the expected value of the example's class over the learned parameter distribution. As for optimization method, an iterative relaxation algorithm is proposed that converges to a local minima, as the problem is not convex. The experimental results for predicting DNA splice points using unlabeled data is encouraging.

(Szummer & Jaakkola, 2001b) used kernel expansion for semi-supervised classification. It includes in the features of each labeled data the kernel densities between it and all other examples, including unlabeled data. With these features, a linear separator for the classification can be derived by maximum entropy discrimination or maximum likelihood. In essence, the relative importance of labeled data is further weighted by the unlabeled data according to the distribution density. But how to select the form and width of the kernel is still an important, but open question.

2.4 Theoretical value of unlabeled data

In this project, we generally discuss the value of unlabeled data in empirical sense, but looking at the existing research on their theoretical value helps in knowing what the limit is. (Ganesalingam & McLachlan, 1978) examined the simplest uni-variate normal distribution with variances known and equal. They calculated the asymptotic relative value of labeled and unlabeled data to first-order approximation. (O'Neill, 1978) calculated the same value further, but for multivariate normals with equivalent and known covariance matrices. (Ratsaby & Venkatesh, 1995) used PAC framework to perform a similar analysis. (Chen, 1995) worked on a class of mixture distributions (including normals) where the number of mixture components is bounded but not known. He bounded the rate of convergence of parameter estimates from unlabeled examples but nothing was said about classification error. All above mentioned results assume that the global ML parameterization is found instead of a local maxima, and the data were actually generated by the model used. For the more general and challenging cases beyond normals, there are little known results. (Cozman & Cohen, 2002) argued that unlabeled data can degrade the performance of a classifier when there are incorrect model assumptions (e.g., set of independence relations among variables or fixed number of labels). (Castelli & Cover, 1995) showed that labeled data reduce error exponentially fast with an infinite amount of unlabeled data, assuming the component distributions are known, but the class-to-component correspondence is not known. Further, (Castelli & Cover, 1996) obtained an important result that for class probability parameters estimation, labeled examples are exponentially more valuable than unlabeled examples, assuming the underlying component distributions are known and correct. (Zhang & Oles, 2000) examine the value of unlabeled data for discriminative classifiers such as TSVMs and for active learning. As mentioned above, they cast doubt on the generality of the helpfulness of TSVMs.

2.5 Active learning

Closely related to semi-supervised learning, there is another interesting problem to ask: if we are offered a chance of labeling a limited number of data, what examples in the pool of unlabeled data are more important for building a classifier? This is called selective sampling which is a form of active learning where one must select an existing example for labeling (Cohn et al., 1994). As error is composed of bias and variance (Geman et al., 1992), one approach is to try to maximally reduce the variance component (Cohn et al., 1996). Another approach is called query-by-committee (QBC) (Seung et al., 1992; Freund et al., 1997), whereby a committee of classifiers are built and the example with the highest classification variance is selected. (Freund et al., 1997) showed theoretically that if there is no error in labeling, QBC can exponentially reduce the number of labeled examples needed for learning. (Liere, 1999) used committees of multiple randomly initialized Perceptrons for QBC active learning for text classification. There, the document selected for labeling is the one whose label is most disagreed by two randomly selected committee members.

Applications of active learning and selective sampling text categorization are abundant. (Argamon-Engelson & Dagan, 1999) used a QBC to learn a part-of-speech tagger. (McCallum & Nigam, 1998a) employed EM in pool-based QBC active learning for text classification. For approaches based on a single classifier instead of a committee, (Lewis & Gale, 1994; Lewis, 1995) examined pool-based *uncertainty sampling* and *relevance sampling*. (Schohn & Cohn, 2000) used an approach for SVM, selecting the example that is closest to the linear decision boundary given by the classifier. (Tong & Koller, 2001) also used SVM, but did selection to maximally reduce the size of the version space of good hypotheses. (Muslea et al., 2003) designed *Aggressive Co-Testing*, exploiting both strong and weak views for detecting the most informative examples. On 33 wrapper induction tasks, this algorithm required significantly fewer labeled examples.

2.6 Other semi-supervised learning models

Besides generative and discriminative approaches, there are also some other effective algorithms to make use of unlabeled data. The early work of co-training describes every example by two disjoint views (Blum & Mitchell, 1998). A case in point is web page classification, where each example has words occurring on a web page, and also anchor texts attached to hyperlinks pointing *to* the web page. In essence, two learning algorithms are trained separately on each view and then each algorithm’s predictions on new unlabeled examples are used to enlarge the training set of the other. They also proved in a PAC-style framework that under certain theoretical assumptions, any weak hypothesis can be boosted from unlabeled data. (Nigam & Ghani, 2000) argue that algorithms explicitly leveraging a natural independent split of the features outperform algorithms that do not. When a natural split does not exist, co-training algorithms that manufacture a feature split may out-perform algorithms not using a split. These arguments help explain why co-training algorithms are both discriminative in nature and robust to the assumptions of their embedded classifiers. (Goldman & Zhou, 2000) went one step further, showing that co-training can even succeed on datasets without separate views, by carefully selecting underlying classifiers.

Also in the setting of co-training, (Collins & Singer, 1999) presented a boosting algorithm, coBoost. It builds a number of classifiers using different views of the data, and minimizes their difference of classification on the unlabeled data. Some other bootstrapping techniques can learn from nearly no labeled data and iteratively develop a concept of interest. (Riloff & Jones, 1999) requires only unannotated training texts and a handful of seed words for a category as input. They use a mutual bootstrapping technique to alternately select the best extraction pattern for the category and bootstrap its extractions into the semantic lexicon, which is the basis for selecting the next extraction pattern. (Yarowsky, 1995) bootstraps a word sense disambiguation algorithm.

A simple thought leads us to use unlabeled data for reducing overfitting. If we have two candidate classifiers or regressors, then overfitting is believed to occur (approximately) when the number of different classification on the unlabeled data is larger than the number of their errors on labeled data. (Schuurmans, 1997) used this observation for selecting the best complexity of a polynomial for regression and (Schuurmans & Southey, 2000) applied it for pruning decision trees. (Cataltepe & Magdon-Ismail, 1998) extended the minimization criteria of mean squared error with terms based on unlabeled (and testing) data to reduce overfitting in linear regression.

2.7 More notes on cluster assumption

In section 2.3, we mentioned the cluster assumption, an important concept that underpins many discriminative semi-supervised learning models. (Li & McCallum, 2004) argued that having a good distance metric is a key requirement for success in semi-supervised learning. This is essentially borrowing techniques from unsupervised learning, which deals only with unlabeled data. This principle is implemented by three commonly used approaches: *manifolds*, *kernels*, and *min-cut*.

(1) **Manifold**. The central idea of manifold is that classification functions are naturally defined only on the submanifold in question rather than the total ambient space. So transforming the representation of examples into a reasonable model of manifold performs feature selection and thus improves classification. As this process does not depend on examples' label, unlabeled data can be naturally utilized. For example, handwritten digit **0** can be fairly accurately represented as an ellipse, which is completely determined by the coordinates of its foci and the sum of distances from the foci to any point. Thus the space of ellipses is a five-dimensional manifold. Though an actual handwritten **0** may require more parameters (like 15 to 20), the dimensionality is absolutely less than ambient representation space, which is the number of pixels. In text data, documents are typically represented by long vectors whereas researchers are convinced by experiments that its space is a manifold, with complicated intrinsic structure occupying only a tiny portion of the original space. The main problem is how to choose a reasonable model of manifold, or more rigorously speaking, how to find a proper basis of the manifold space.

(Belkin & Niyogi, 2002; Belkin & Niyogi, 2004) used the Laplace-Beltrami operator $\Delta = -\sum_i \partial^2 / \partial x_i^2$, which is positive-semidefinite self-adjoint on twice differentiable functions. When $\mathcal{M}$ is a compact manifold, Δ has a discrete spectrum and its eigenfunctions provide an orthogonal basis for the Hilbert space $\mathcal{L}^2(\mathcal{M})$. Suppose we are given k points $x_1, \ldots, x_k \in \mathbb{R}^l$ and the first $s < k$ points have labels $c_i \in \{-1,1\}$. First construct an adjacency graph with n nearest and reverse nearest neighbors, with distance defined as standard Euclidean distance in $\mathbb{R}^l$, or other distance like angle/cosine. Then define adjacency matrix $W_{k\times k}$. $w_{ij} = 1$ if points x_i and x_j are close. Otherwise, $w_{ij} = 0$. Compute p eigenvectors corresponding to the smallest p eigenvalues for

$L = W - D$, where D is a diagonal matrix of the same size as W and $D_{ii} = \sum_j W_{ji}$. They comprise matrix $E_{p\times k}$. Denote the left $p \times s$ sub-matrix as E_{lab} and calculate $\vec{a} = (E_{lab}^T E_{lab})^{-1} E_{lab}^T \vec{c}$. Finally, for x_i ($i > s$) the rule for classification is: $c_i = 1$ if $\sum_{j=1}^{p} e_{ij} a_j \geq 0$ and $c_i = -1$ otherwise. In essence, this is a spectral clustering and there are many variants of such a PCA-like data representation (Weiss, 1999; Ng et al., 2001).

(2) **Kernel**. In this approach, kernels are designed so as to make the induced distance small for points in the same cluster and larger for points in different clusters. As an extension of *Fisher kernel* (Jaakkola & Haussler, 1998), a Marginalized kernel for mixture of Gaussians (μ_k, Σ_k) is proposed: $K(x,y) = \sum_{k=1}^{q} P(k \mid x) P(k \mid y) x^T \Sigma_k^{-1} y$ (Tsuda et al., 2002). But in addition to common problem of generative models, this method requires building a generative model: finding parameters μ_k and Σ_k using unsupervised learning. (Szummer & Jaakkola, 2001a) used RBF kernel matrix $K_{ij} = \exp(-\| x_i - x_j \| / \sigma)$ and used it as a transition matrix of Markov random walk on a graph with vertices x_i, $P(x_i \to x_j) = K_{ij} / \sum_p K_{ip}$. They designed a discriminative classifier based on t step transition matrix $P^t = (D^{-1}K)^t$, where D is a diagonal matrix with $D_{ii} = \sum_j K_{ij}$.

(Chapelle et al., 2002) proposed a framework of cluster kernels that unifies the Markov random walk, kernel PCA, and certain types of spectral clustering, by applying a transfer function to the eigenvalues of graph Laplacian for adjacency graph and then recover it. The different forms of transfer function, such as step, linear-step, polynomial, play the central role of unification.

(3) **Min-cut**. The intuition behind this method is to use pairwise relationships (more precisely, similarity measure) among the labeled and unlabeled examples to construct a graph, and then output a classification corresponding to partitioning the graph in a way that minimizes (roughly) the number of similar pairs of examples that are given different labels. The idea originated from computer vision (Greig et al., 1989; Boykov et al., 1998; Yu et al., 2002). (Blum & Chawla, 2001) applied it to semi-supervised learning. The formulation is illustrated in Figure 1.

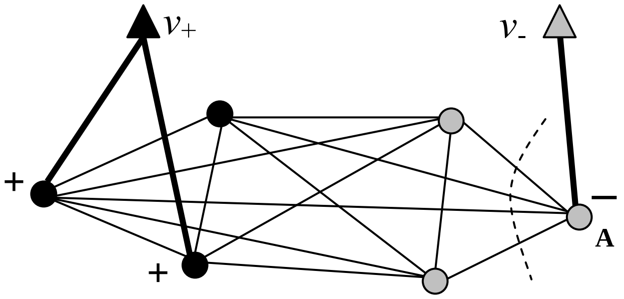

Figure 1: Min-cut algorithm. Three labeled examples are denoted with + and –. They are connected to respective classification nodes (denoted by triangles) with bold lines. Other three nodes are unlabeled and finally the left one is classified as + and right two nodes as –.

v_+ and v_- are artificial vertices called classification nodes. They are connected to positive and negative examples respectively, with infinite weight represented by bold lines. The edges between example vertices are assigned weights based on some relationship between them, such as similarity or distance. Now we determine a minimum (v_+, v_-) cut for the graph, i.e., find the minimum total weight set of edges whose removal disconnects v_+ and v_-. The problem can be solved using max-flow algorithm, in which v_+ is the source, v_- is the sink and the edge weights are treated as capacities (e.g., (Cormen et al., 2001)). Removing the edges in the cut partitions the graph into two sets of vertices which we call V_+ and V_-, with $v_+ \in V_+$, $v_- \in V_-$. We assign positive label to all unlabeled examples in V_+ and negative labels to all examples in V_-. (Blum & Chawla, 2001) also showed several theoretical proofs for the equivalence between min-cut and leave-one-out error minimization under various *k*NN settings.

But an obvious problem of min-cut is that it may lead to degenerative cuts. Suppose A is the only negative-labeled example in Figure 1. Then the partition given by min-cut may be the dashed line. Obviously this is not desired. The remedies in (Blum & Chawla, 2001), such as carefully adjusting edge weights, do not work across all problems they study. (Joachims, 2003) proposed Spectral Graph Partitioning (SGT) with normalization, by dividing the cut by the product of the number of positive and negative examples. He also approximated this NP-hard problem with the spectrum of Laplacian, and demonstrated its connection (equivalence) with TSVM, co-training and other *k*NN models. He also posed three postulates for building a good transductive learning:

1. It achieves low training error;
2. the corresponding inductive learner is highly self-consistent (e.g., low leave-one-out error);
3. Averages over examples (e.g., average margin, pos/neg ratio) should have the same expected value in the training and in the test set.

The last postulate is very similar to the maximum entropy model which we will discuss in the next chapter.

Chapter 3 Maximum Entropy models

Maximum Entropy (MaxEnt) modeling is a powerful and robust tool that has been successfully applied to a wide range of domains, including language modeling, species distribution modeling, as well as many other natural language tasks (Jaynes, 1957; Berger et al., 1996; Rosenfeld, 1996; Pietra et al., 1997; Ratnaparkhi, 1998; Phillips et al., 2004). In this project, we put more focus on conditional MaxEnt than to (unconditional) random field MaxEnt. The MaxEnt model can usually be expressed in the frame of generalized variants of Kullback-Leibler divergence, and has close relationship with boosting (Lebanon & Lafferty, 2001; Collins et al., 2002). For many problems, this type of modeling can be viewed as a variant of maximum likelihood (ML) training for exponential models, while at the same time making the model as similar as possible to the uniform distribution (minimizes KL divergence). Like other ML methods, it is prone to overfitting of training data. So an important direction of ME research is smoothing methods.

3.1 Standard MaxEnt formulation

One concept that contributed to the flexibility of MaxEnt is called *features.* An event is decomposed into many features, which indicate the strength of certain aspects in the event. For the same event, different features can be defined or induced for different purposes. For example, we can define $f_t(x,y)=1$ if and only if the current word, which is part of document x, is "back" and the class y is verb. Otherwise, $f_t(x, y) = 0$. Real valued features are also sometimes used.

The original MaxEnt model is formulated as follows. We use i, j as index for examples, k as index for classes, t as index for features. We use $\tilde{p}(x_i)$ to denote the empirical distribution (distribution for training examples) and $p(x_i)$ to denote the real distribution of examples.

$$\text{minimize} \qquad \sum_i p(x_i) \sum_k p(y_k \mid x_i) \log p(y_k \mid x_i) \tag{3.1}$$

$$s.t. \qquad E_{\tilde{p}}[f_t] - \sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k) = 0 \quad \text{for all } t \tag{3.2}$$

$$\sum_k p(y_k \mid x_i) = 1 \quad \text{for all } i \tag{3.3}$$

where $E_{\tilde{p}}[f_t] = \sum_i \tilde{p}(x_i) \sum_k \tilde{p}(y_k \mid x_i) f_t(x_i, y_k)$.

The dual problem is to minimize:

$$L(p_{\min}, \lambda) = -\sum_t \lambda_t E_{\tilde{p}}[f_t] + \sum_i p(x_i) \log Z_i \tag{3.4}$$

where $Z_i = \sum_k \exp\left(\sum_t \lambda_t f_t(x_i, y_k) \right)$.

If we let $p(x_i) = \tilde{p}(x_i)$, then it is equivalent to ML problem for logistic or soft-max regression, which maximizes:

$$
\begin{aligned}
L(\lambda) &= \sum_i \sum_k \tilde{p}(x_i, y_k) \log p(x_i, y_k) = \sum_i \sum_k \tilde{p}(x_i, y_k) \left(\log p(x_i) + \sum_t \lambda_t f_t(x_i, y_k) - \log Z_i \right) \\
&= \sum_i \tilde{p}(x_i) \log p(x_i) + \sum_i \sum_k \tilde{p}(x_i, y_k) \sum_t \lambda_t f_t(x_i, y_k) - \sum_i \tilde{p}(x_i) \log Z_i \\
&= \sum_i \tilde{p}(x_i) \log \tilde{p}(x_i) + \sum_t \lambda_t E_{\tilde{p}}[f_t] - \sum_i p(x_i) \log Z_i \qquad (3.5)
\end{aligned}
$$

This is equivalent to minimizing (3.4). Making $p(x_i) \neq \tilde{p}(x_i)$ will naturally extend the ML model to MaxEnt using unlabeled data x (formally, $p(x) \neq 0$ but $\tilde{p}(x) = 0$), because Z_i and the equation in (3.4) do not depend on the class of unlabeled data.

According to (Pietra et al., 1997), we have a generalized Pythagorean theorem for random field. Now we extend the conclusion to our conditional probability situation.

We define

$$
\mathcal{M} = \left\{ m \middle| \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+ \right\} \qquad \Delta = \{ m \in \mathcal{M} | \sum_{y \in \mathcal{Y}} m(x, y) = 1 \}
$$

$$
D(p,q) \triangleq \sum_i \tilde{p}(x_i) \sum_k \left(p(y_k | x_i) \log \frac{p(y_k | x_i)}{q(y_k | x_i)} - p(y_k | x_i) + q(y_k | x_i) \right) \text{ on } \mathcal{M} \times \mathcal{M} \qquad (3.6)
$$

$$
\mathcal{Q}_1(q_0, f) = \left\{ q \in \mathcal{M} \,|\, q(y_k | x_i) = q_0(y_k | x_i) \exp\left(\left\langle \lambda, f(x_i, y_k) - f(x_i, \tilde{y}(x_i)) \right\rangle \right) \right\} \qquad (3.7)
$$

$$
\mathcal{Q}_2(q_0, f) = \left\{ q \in \Delta \,|\, q(y_k | x_i) \propto q_0(y_k | x_i) \exp\left(\left\langle \lambda, f(x_i, y_k) \right\rangle \right), \lambda \in \mathbb{R}^m \right\} \qquad (3.8)
$$

$$
\mathcal{F}(\tilde{p}, f) = \left\{ p \in \mathcal{M} \middle| \sum_i \tilde{p}(x_i) \sum_k p(y_k | x_i) f_t(x_i, y_k) = E_{\tilde{p}}[f_t] \right\} \qquad (3.9)
$$

$$
q^{\star}_{boost} \in \mathcal{F}(\tilde{p}, f) \bigcap \mathcal{Q}_1(q_0, f) \qquad q^{\star}_{me} \in \mathcal{F}(\tilde{p}, f) \bigcap \mathcal{Q}_2(q_0, f) \qquad (3.10)
$$

Then we have:

If $D(\tilde{p}, q_0) < +\infty$, then $q^{\star}_{boost}$ and $q^{\star}_{me}$ both exist, are unique and satisfy:

$$
q^{\star}_{boost} = \arg\min_{p \in \mathcal{F}} D(p, q_0) = \arg\min_{q \in \overline{\mathcal{Q}}_1} D(\tilde{p}, q) = \arg\min_{q \in \overline{\mathcal{Q}}_1} \sum_i p(x_i) \sum_k q(y_k | x_i) \qquad (3.11)
$$

$$
q^{\star}_{me} = \arg\min_{p \in \mathcal{F} \bigcap \Delta} D(p, q_0) = \arg\min_{q \in \overline{\mathcal{Q}}_2} D(\tilde{p}, q) = \arg\min_{q \in \overline{\mathcal{Q}}_2} \sum_i \tilde{p}(x_i) \log {}^1\!\!\diagup\!{q(\tilde{y}_k | x_i)} \qquad (3.12)
$$

And $q^{\star}_{me}$ can be computed in terms of $q^{\star}_{boost}$ as $q^{\star}_{me} = \arg\min_{p \in \mathcal{F} \bigcap \Delta} D(p, q^{\star}_{boost})$.

The proof is very similar to (Pietra et al., 1997) and the crux is to prove a generalized "Pythagorean theorem":

$$
\text{for any } p \in \mathcal{F},\ q \in \mathcal{Q}_i, p_{M_i} \in \mathcal{F} \bigcap \mathcal{Q}_i,\ D(p \,\|\, q) = D(p \,\|\, p_{M_i}) + D(p_{M_i} \,\|\, q) \ \ i = 1{,}2.
$$

The following figure gives a geometric view of the conclusion. In (3.12), if we set q_0 to uniform distribution, then $\arg\min_{p \in \mathcal{F} \bigcap \Delta} D(p, q_0)$ is actually the formulation from (3.1) to (3.3). A more general and rigorous formulation is by using Bregman distance (Pietra et al., 2002; Collins et al., 2002).

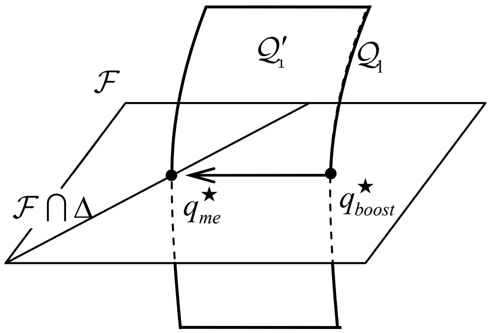

Figure 2: Geometric view of duality. $\mathcal{Q}_1$ intersects with $\mathcal{F}$ at $q^{\star}_{boost}$. If we impose additional constraint that each conditional distribution be normalized, we introduce a Lagrange multiplier giving a higher dimensional family $\mathcal{Q}_1'$. The projection of $q^{\star}_{boost}$ on $\mathcal{F} \bigcap \Delta$ should overlap with the intersection of $\mathcal{Q}_1'$ and $\mathcal{F} \bigcap \Delta$, which is $q^{\star}_{me}$.

3.2 Smoothing techniques for MaxEnt models

It should not be surprising that MaxEnt can severely overfit training data when the constraints on the output distribution are based on feature expectations, especially if there is a very large number of feature. Similar to ML models forcing probabilities to zero at unseen values, maximizing the entropy on real data often fails to produce finite parameters. Formally speaking, suppose there are some features f_t: $f_t(x,y) - f_t(x,\tilde{y}(x)) \geq 0$ for all y and x with $\tilde{p}(x) > 0$ or similarly $f_t(x,y) - f_t(x,\tilde{y}(x)) \leq 0$. In this case the corresponding Lagrangian multipliers must approach infinity to satisfy the constraints (3.2). If such features are the most important for classification, the effect is especially harmful. On the other hand, we do expect that empirical averages will be *close* to their expectations and we often have bounds or estimates on deviation of empirical feature averages from their true expectations, which can be used as constraints or soft penalty parameters. Therefore, we must consider smoothing techniques, which are divided into two categories: constraint relaxation and prior over exponential family distribution. Sometimes, one technique can be interpreted in both ways.

The simplest regularization is by using Gaussian Prior (Chen & Rosenfeld, 2000), which is equivalent to Maximum A Posterior (MAP) estimation problem for logistic or soft-max regression with Gaussian prior. Introducing parameters σ_i, the primal problem becomes:

$$\text{minimize:} \qquad \sum_i p(x_i) \sum_k p(y_k \mid x_i) \log p(y_k \mid x_i) + \sum_t \frac{\sigma_t^2}{2} \delta_t^2 \tag{3.13}$$

$$s.t. \qquad E_{\tilde{p}}[f_t] - \sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k) = \delta_t \quad \text{for all } t \tag{3.14}$$

$$\sum_k p(y_k \mid x_i) = 1 \quad \text{for all } i \tag{3.15}$$

Its dual problem is: minimize $L(\lambda) = -\sum_t \lambda_t E_{\tilde{p}}[f_t] + \sum_i \tilde{p}(x_i) \log(Z_i) + \sum_t \frac{\lambda_t^2}{2\sigma_t^2}$ (3.16) where $Z_i = \sum_k \exp\left(\sum_t \lambda_t f_t(x_i, y_k) \right)$ and optimization is over $\lambda_t \in \mathbb{R}$.

Another model is based on Laplace prior (Goodman, 2004). Laplace prior has been studied in the context of neural networks by (Williams, 1995). This formulation for MaxEnt is similar:

$$\text{minimize} \qquad \sum_i p(x_i) \sum_k p(y_k \mid x_i) \log p(y_k \mid x_i) \tag{3.17}$$

$$s.t. \qquad E_{\tilde{p}}[f_t] - \sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k) \le A_t \quad \text{for all } t \tag{3.18}$$

$$\sum_k p(y_k \mid x_i) = 1 \quad \text{for all } i \tag{3.19}$$

The dual problem is to minimize: $L(\lambda) = -\sum_t \lambda_t E_{\tilde{p}}[f_t] + \sum_i p(x_i) \log Z_i + \sum_t A_t \lambda_t$ (3.20)

where $Z_i = \sum_k \exp(\sum_t \lambda_t f_t(x_i, y_k))$, and A_t is the reciprocal of the standard deviation of $f_t(x, y)$ in training data. The optimization is over $\lambda_t \ge 0$. This problem is equivalent to maximizing:

$$\prod_i p(\tilde{y}_i \mid x_i) \times \prod_t A_t \exp(-A_t \lambda_t) \tag{3.21}$$

with respect to $\lambda_t \ge 0$. Here $A_t \exp(-A_t \lambda_t)$ ($\lambda_t \ge 0$) is obviously exponential prior. (3.18) indicates this model also belongs to constraint relaxation. Strictly speaking, Laplace prior means maximizing $\prod_i p(\tilde{y}_i \mid x_i) \times \prod_t \frac{1}{2A_t} \exp\left(-\frac{|\lambda_t|}{A_t} \right)$. But it is far easier to work with exponential prior, partly because it is difficult to find learning algorithms for Laplace prior due to its being nondifferentiable at $\lambda_t = 0$. Simply applying Kuhn-Tucker theorem, we expect that the resulting model will often favor Lagrangian multipliers that are exactly 0 (same conclusion applies to Laplace prior). This means the corresponding features can be removed from model without changing its prediction behavior, but improving robustness. Therefore, the model is useful as a kind of natural pruning, not found in Gaussian priors.

The model has another important theoretical characteristic: in (3.18) we discount the observed average by $A_t > 0$. This provides nice ground for discounting-based language modeling smoothing techniques, such as Kneser-Ney smoothing (Kneser & Ney, 1995; Chen & Goodman, 1999). But as λ_t must be positive for exponential prior, special care must be paid to feature selection. Suppose for a word sense disambiguation problem, we try to determine whether ‘cell’ means the biology or prison sense, with questions like whether the word ‘reagent’ occurs nearby. For Gaussian prior and Laplace prior, we only need to define $f_1(x, y) = 1$ iff ‘reagent’ occurs nearby and its corresponding Lagrangian multiplier will take value in $(-\infty, +\infty)$. But as we restrict λ_t to be positive for exponential prior, we must define two features: $f_1(x, y) = 1$ iff ‘reagent’ occurs nearby; $f_2(x, y) = 1$ iff ‘prisoner’ occurs nearby. We have two non-negative weights, one pushing towards one answer and the other pushing towards the other.

Extending exponential prior more, we arrive at *Inequality MaxEnt*, using box-type inequality constraints (Newman, 1977; Khudanpur, 1995; Kazama & Tsujii, 2003). The primal problem is:

$$\text{miminize} \qquad \sum_i p(x_i) \sum_k p(y_k \mid x_i) \log p(y_k \mid x_i) \tag{3.22}$$

$$s.t. \qquad -B_t \leq E_{\tilde{p}}[f_t] - \sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k) \leq A_t \quad \text{for all } t \tag{3.23}$$

$$\sum_k p(y_k \mid x_i) = 1 \quad \text{for all } i \tag{3.24}$$

Its dual problem is:

$$\text{minimize} \qquad L(\alpha, \beta) = -\sum_t (\alpha_t - \beta_t) E_{\tilde{p}}[f_t] + \sum_i p(x_i) \log Z_i + \sum_t A_t \alpha_t + \sum_t B_t \beta_t \tag{3.25}$$

where $Z_i = \sum_k \exp(\sum_t (\alpha_t - \beta_t) f_t(x_i, y_k))$. Optimization is over $\alpha_t \geq 0, \beta_t \geq 0$.

Inequality MaxEnt keeps the strength of easy feature selection. A new advantage is the existence of non-asymptotic bounds showing that with respect to the true underlying distribution, this relaxed version of MaxEnt produces conditional probability estimates that are almost as good as the best possible. These bounds are in terms of the deviation of the feature empirical averages relative to their true expectations, a number that can be bounded using standard uniform-convergence techniques. In particular, this leads to bounds that drop quickly with the number of samples, and that depend very moderately on the number of complexity of the features. Result will be presented in Chapter 4 under the settings of semi-supervised learning.

There are some other related forms of smoothing techniques. In brief,

Inequality with 2-norm Penalty (Kazama & Tsujii, 2003):

$$\text{minimize} \qquad \sum_i p(x_i) \sum_k p(y_k \mid x_i) \log p(y_k \mid x_i) + C_1 \sum_t \delta_t^2 + C_2 \sum_t \zeta_t^2 \tag{3.26}$$

$$s.t. \qquad E_{\tilde{p}}[f_t] - \sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k) \leq A_t + \delta_t \quad \text{for all } t \tag{3.27}$$

$$\sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k) - E_{\tilde{p}}[f_t] \leq B_t + \zeta_t \quad \text{for all } t \tag{3.28}$$

$$\sum_k p(y_k \mid x_i) = 1 \quad \text{for all } i \tag{3.29}$$

Inequality with 1-norm Penalty (Kazama & Tsujii, 2003):

$$\text{minimize} \qquad \sum_i p(x_i) \sum_k p(y_k \mid x_i) \log p(y_k \mid x_i) + C_1 \sum_t \delta_t + C_2 \sum_t \zeta_t \tag{3.30}$$

$$s.t. \qquad E_{\tilde{p}}[f_t] - \sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k) \leq A_t + \delta_t \quad \text{for all } t \tag{3.31}$$

$$\sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k) - E_{\tilde{p}}[f_t] \leq B_t + \zeta_t \quad \text{for all } t \tag{3.32}$$

$$\sum_k p(y_k \mid x_i) = 1 \quad \text{for all } i \tag{3.33}$$

$$\delta_t \geq 0, \zeta_t \geq 0 \quad \text{for all } t \tag{3.34}$$

3.3 MaxEnt parameter estimation

The flexibility of MaxEnt model is not without cost. While the parameter estimation for MaxEnt (solving optimization problem) is conceptually straightforward, in practice MaxEnt models may well contain many thousands of free parameters and the efficiency will become a crucial problem. The main algorithms for this task include: *Generalized Iterative Scaling*, *Improved Iterative Scaling*, general purpose optimization techniques such as *gradient descent*, *conjugate gradient descent*, and *limited memory variable metric* (*LMVM*) methods (Malouf, 2002). We discuss for both the random field and conditional situation.

From (3.4), we get $\frac{\partial L}{\partial \lambda_t} = -E_{\tilde{p}}[f_t] + E_p[f_t]$. As the problem is convex optimization, there is only one global minimum where the gradient is zero. But simply setting $\frac{\partial L}{\partial \lambda} = 0$ does not yield a closed form solution for λ. So we have to proceed iteratively, by adjusting estimate of $\lambda^{(s)}$ to a new estimate $\lambda^{(s+1)}$ based on the divergence between the estimated probability $p^{(s)}$ and $\tilde{p}$.

One popular method for iterative refining the model parameters is *Generalized Iterative Scaling* (*GIS*) (Darroch & Ratcliff, 1972), which is an extension of *Iterative Proportional Fitting* (*IPF*) (Deming & Stephan, 1940). The prerequisite of original GIS is that for all training examples x_i: $f_t(x_i) \geq 0$ and $\sum_t f_t(x_i) = 1$. The update rule is: $\lambda_t^{(s+1)} = \lambda_t^{(s)} + \log\left(\frac{E_{\tilde{p}}[f_t]}{E_{p^{(s)}}[f_t]}\right)$ and $p^{(s+1)}(x_i) = p^{(s)}(x_i)\prod_t \left(\frac{\sum_j \tilde{p}(x_j) f_t(x_j)}{\sum_j p^{(s)}(x_j) f_t(x_j)}\right)^{f_t(x_i)}$. The derivation is like EM, involving an auxiliary function after bounding the likelihood twice. Note the algorithm is parallel, in that $\lambda_t^{(s)}$ are updated synchronously and $p^{(s+1)}(x_i)$ contains a product over all features. It converges to the unique optimal value of λ. $p^{(s)}(x_i)$ may not be normalized, but it works fine and the limiting distribution *is* normalized.

The prerequisites of GIS can be relaxed. For positive constant C, GIS can be extended to $\sum_t f_t(x_i) = C$ by defining $f_t' = f_t / C$. In case not all training data have summed features equaling C, we can set C sufficiently large and incorporate a 'correction feature', though it effectively slows convergence to match the most difficult case. This technique can also help handle the negative features by lifting all features with a common constant, setting a large enough C and then compensating with the correction feature.

For conditional probability, GIS turns out to be very similar to the unconditional case: $\lambda_t^{(s+1)} = \lambda_t^{(s)} + \eta \log\left(\frac{E_{\tilde{p}}[f_t]}{\sum_i \tilde{p}(x_i) \sum_k p(y_k \mid x_i, \lambda^{(s)}) f_t(x_i, y_k)}\right) = \lambda_t^{(s)} + \eta \log\left(\frac{E_{\tilde{p}}[f_t]}{E_{p^{(s)}}[f_t]}\right)$. But it is not possible now to express update rules for $p(x_i)$ without normalization factors, though calculating them barely influences actual computation complexity.

To avoid this slowed convergence and the need for correction feature, *Improved Iterative Scaling* (*IIS*) algorithm is proposed (Pietra et al., 1997), which only requires non-negativity of feature values. Its update rule is the solution of $\Delta\lambda_t$ to $\Delta\lambda_t$

$$E_{\tilde{p}}[f_t] = \sum_i p^{(s)}(x_i) f_t(x_i) \exp\left(\Delta\lambda_t \sum_j f_j(x_i)\right) \quad \text{for all } t \qquad (3.35)$$

The main result is: the sequence $p(x_i)$ monotonically decreases the MaxEnt objective function and converges to the optimal value. The proof is still like EM and GIS, via expressing the incremental step in terms of an auxiliary function which bounds from below the objective function. Solving (3.35) requires some attention. A good thing is $\Delta\lambda_t$ are decoupled and are solved individually. If feature values are all integers, then equations are all polynomial in $\exp(\Delta\lambda_t)$ and can be found straightforwardly using, for example, Newton-Raphson method. Otherwise there are also efficient numerical algorithms. If the number of possible x_i is too large, Monte Carlo methods are to be used. Then the coefficients of all equations in (3.35) can be simultaneously estimated for all t and i, by generating a single set of samples from $p^{(s)}(x_i)$. The IIS for conditional probability distribution is similar, by modifying (3.35) as:

$$E_{\tilde{p}}[f_t] = \sum_i \tilde{p}(x_i) \sum_k p(y_k \mid x_i, \lambda^{(s)}) f_t(x_i, y_k) \exp\left(\Delta\lambda_t \sum_j f_j(x_i, y_k) \right) \text{ for all } t \tag{3.36}$$

The GIS and IIS discussed above are for standard MaxEnt models. The variant versions for MaxEnt with different smoothing methods can be found in their respective papers.

Iterative scaling algorithms have long been flying over the face of statistics and are still widely used for analysis of contingency tables. The main advantage is that each update depends only on the computation of expected values $E_{p^{(s)}}$, not requiring the gradient or higher derivatives whose computation can be prohibitively expensive for some practical distributions. However, in MaxEnt model, the expected values for all features are required in each iteration, and they are effectively gradients of objective function. Therefore, directly using gradient based optimization methods does make sense. The simplest form is gradient descent, updating $\lambda_t^{(s+1)} = \lambda_t^{(s)} + \eta \left.\frac{\partial L}{\partial \lambda_t}\right|_{\lambda = \lambda^{(s)}}$, where η is the step size applied to all variables. Its main minus is two folds. Firstly, it is difficult to find good η. Being too large will cause instability (divergence, oscillation), and being too small will lead to extremely poor rate of convergence. Also, a step locally optimal in a very narrow sense might make each new search direction nearly orthogonal to the immediate previous one, leading to zig-zag descent with convergence severely slowed down. One improvement considers each possible search direction only once, always taking a step of exactly right length in a direction orthogonal to all previous search directions. These are *conjugate gradient* methods, such as *Fletcher-Reeves* and *Polak-Ribiêre-Positive* algorithm differing in update rules and numerical properties, though they are theoretically equivalent.

More advanced algorithms make use of curvature, or second-order derivatives. Based on Taylor's series: $L(\lambda + \Delta\lambda) \approx L(\lambda) + \Delta\lambda^T \cdot \partial L / \partial \lambda + \frac{1}{2} \Delta\lambda^T H(\lambda) \Delta\lambda$, where H is the *Hessian* matrix. Setting to zero its derivative to Δλ and solving the equation, we get *Newton's method* $\Delta\lambda^{(s)} = H^{-1}(\lambda^{(s)}) \partial L / \partial \lambda^{(s)}$. Though convergence is clearly accelerated, the biggest problem of the algorithm is the huge cost of storing H (square to the number of variables) and computing its inverse in each iteration. So (*limited memory*) *variable metric*, *quasi-Newton* methods are designed that approximate Hessian using successive evaluations of gradient. They have excellent convergence properties and much more space/computation efficient.

An extensive comparison is given in (Malouf, 2002). Surprisingly, the standardly used iterative scaling algorithms perform quite poorly in comparison with the others and LMVM outperformed the other choices in almost all test problems.

All the algorithms above are in a parallel updating style, i.e., updating the parameters by considering all features. For a very large (or infinite) number of features, this kind of algorithms will be too resource consuming to be feasible. (Collins et al., 2002) proposed a sequential-update algorithm, which, in a style of coordinate-wise descent, modifies one parameter at a time. Theoretically, it converges to the same optimum as parallel update.

Chapter 4 Incorporating unlabeled data in MaxEnt models

The aim of the research project is to combine the advantage of both semi-supervised learning and MaxEnt models. As noted in section 2.7, most semi-supervised learning algorithms are based on assumptions, which must be adjusted throughout different datasets and may cause instability. The project aims to find a stable semi-supervised learning algorithm by applying the MaxEnt framework.

4.1 Why do we choose MaxEnt?

We will answer the question in the following three sections. The first advantage of MaxEnt is that it allows natural incorporation of unlabeled data. To make it clear, we copy the standard MaxEnt here and interpret it in the semi-supervised learning settings.

$$\text{minimize} \qquad \sum_i p(x_i) \sum_k p(y_k \mid x_i) \log p(y_k \mid x_i) \tag{4.1}$$

$$s.t. \qquad E_{\tilde{p}}[f_t] - \sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k) = 0 \quad \text{for all } t \tag{4.2}$$

$$\sum_k p(y_k \mid x_i) = 1 \quad \text{for all } i \tag{4.3}$$

where $E_{\tilde{p}}[f_t] = \sum_i \tilde{p}(x_i) \sum_k \tilde{p}(y_k \mid x_i) f_t(x_i, y_k)$.

The dual problem to minimize is:

$$L(p_{\min}, \lambda) = -\sum_t \lambda_t E_{\tilde{p}}[f_t] + \sum_i p(x_i) \log Z_i \tag{4.4}$$

In Chapter 3, we always assumed that $p(x) = \tilde{p}(x)$. But now, if we view the equations in a new angle, $p(x)$ can be easily extended to include unlabeled data. As Z_i sums up over all classes, it can be calculated even for unlabeled data. Here $E_{\tilde{p}}[f_t]$ is based on empirical data and (4.2) assumes that $E_{\tilde{p}}[f_t]$ is a good estimation of the average value of f_t on the whole dataset, including both the unlabeled and labeled data. This is similar to the postulation 3 in section 2.7, and this is the only assumption we make. No assumption over clustering, neighboring, or distance measurement is made. Besides, the MaxEnt has an in-built tendency to choose uniform distribution. This can be used as a normalization approach and there is often physical support for the bias.

This extension of $p(x)$ also applies to the variants of MaxEnt model. Besides providing a natural way to incorporate unlabeled data, MaxEnt possesses a second advantage of having provable estimation error bounds, which we describe in the next section.

4.2 Estimation error bounds

We use Inequality MaxEnt as an example. (Dudik et al., 2004) showed that for normal random fields, with respect to the true underlying distribution, Inequality MaxEnt produces probability estimates that are almost as good as the best possible. We hereby derive a new generalization error bound for semi-supervised learning with conditional probability distribution.

First the Inequality MaxEnt is restated here:

$$\text{miminize} \qquad \sum_i p(x_i) \sum_k p(y_k \mid x_i) \log p(y_k \mid x_i) \tag{4.5}$$

$$s.t. \qquad -B_t \le E_{\tilde{p}}[f_t] - \sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k) \le A_t \quad \text{for all } t \tag{4.6}$$

$$\sum_k p(y_k \mid x_i) = 1 \quad \text{for all } i \tag{4.7}$$

$$\text{where} \qquad E_{\tilde{p}}[f_t] = \sum_i \tilde{p}(x_i) \sum_k \tilde{p}(y_k \mid x_i) f_t(x_i, y_k) \tag{4.8}$$

We allow inconsistent data, i.e., different y's associated with the same x. But in our maxent formulation, we need the correct expectation of f_t to estimate the model. Let the universe of discourse of x be those present in the labeled and unlabeled data. Let the 'correct' conditional distribution of y on x be $p^C(y_k \mid x_i)$. Then we define the 'correct' expectation of f_t as:

$$E_p^C[f_t] = \sum_i p^C(x_i) \sum_k p^C(y_k \mid x_i) f_t(x_i, y_k) \tag{4.9}$$

$p(x_i) = p^C(x_i)$. We use $E_p^C[f_t]$ to emphasize that it is neither the empirical expectation, nor the expectation given by our estimated model. We will use $E_p[f_t]$ to denote the expectation given by our estimated model. Similarly, we use $p(x_i)$, $p(y_k|x_i)$ for our estimated model and $p^C(x_i)$, $p^C(y_k|x_i)$ for the theoretically 'correct' model.

Using a little different notation, the dual problem of Inequality MaxEnt is to minimize:

$$L_{\tilde{p}}^{A,B}(\lambda) = -\sum_t (\alpha_t - \beta_t) E_{\tilde{p}}[f_t] + \sum_i p(x_i) \log Z_i + \sum_t A_t \alpha_t + \sum_t B_t \beta_t \tag{4.10}$$

where $Z_i = \sum_k \exp(\sum_t (\alpha_t - \beta_t) f_t(x_i, y_k))$.

As there is at most one positive value in α_t and β_t, we define $\lambda_t = \alpha_t - \beta_t$. Obviously, $|\lambda_t| = \alpha_t + \beta_t$, so $\alpha_t = \left(\lambda_t + |\lambda_t|\right)/2$, $\beta_t = \left(|\lambda_t| - \lambda_t\right)/2$. Then (4.10) becomes:

$$L_{\tilde{p}}^{A,B}(\lambda) = -\sum_t \lambda_t E_{\tilde{p}}[f_t] + \sum_i p(x_i) \log Z_i + \sum_t A_t \frac{\left(\lambda_t + |\lambda_t|\right)}{2} + \sum_t B_t \frac{\left(|\lambda_t| - \lambda_t\right)}{2} \tag{4.11}$$

However, the 'correct' Lagrangian to be minimized is:

$$L_p^C(\lambda) = -\sum_t \lambda_t E_p^C[f_t] + \sum_i p(x_i) \log Z_i \tag{4.12}$$

So we wish to show that the optimal λ for $L_{\tilde{p}}^{A,B}(\lambda)$ will not make $L_{p^C}(\lambda)$ too deviated from its own optimal value. For convenience we also define:

$$\hat{\lambda} = \arg\min_{\lambda} L_{\tilde{p}}^{A,B}(\lambda) \tag{4.13}$$

$$\lambda^* = \arg\min_{\lambda} L_p^C(\lambda) \tag{4.14}$$

$$L_{\tilde{p}}(\lambda) = -\sum_t \lambda_t E_{\tilde{p}}[f_t] + \sum_i p(x_i)\log Z_i \tag{4.15}$$

Now,

$$\begin{aligned}
L_p^C(\hat{\lambda}) &= L_{\tilde{p}}(\hat{\lambda}) - \sum_k \hat{\lambda}_k \left(E_p^C[f_k] - E_{\tilde{p}}[f_k]\right) \\
&= L_{\tilde{p}}^{A,B}(\hat{\lambda}) - \sum_t A_t \left(\frac{\hat{\lambda}_t + \left|\hat{\lambda}_t\right|}{2}\right) - \sum_t B_t \left(\frac{\left|\hat{\lambda}_t\right| - \hat{\lambda}_t}{2}\right) - \sum_t \hat{\lambda}_t \left(E_p^C[f_t] - E_{\tilde{p}}[f_t]\right) \\
&\le L_{\tilde{p}}^{A,B}(\lambda^*) - \sum_k A_t \left(\frac{\hat{\lambda}_t + \left|\hat{\lambda}_t\right|}{2}\right) - \sum_t B_t \left(\frac{\left|\hat{\lambda}_t\right| - \hat{\lambda}_t}{2}\right) - \sum_t \hat{\lambda}_t \left(E_p^C[f_t] - E_{\tilde{p}}[f_t]\right) \\
&= L_{\tilde{p}}(\lambda^*) - \sum_t A_t \left(\frac{\hat{\lambda}_t + \left|\hat{\lambda}_t\right|}{2}\right) - \sum_t B_t \left(\frac{\left|\hat{\lambda}_t\right| - \hat{\lambda}_t}{2}\right) - \sum_t \hat{\lambda}_t \left(E_p^C[f_t] - E_{\tilde{p}}[f_t]\right) \\
&\qquad + \sum_t A_t \left(\frac{\lambda_t^* + \left|\lambda_t^*\right|}{2}\right) + \sum_t B_t \left(\frac{\left|\lambda_t^*\right| - \lambda_t^*}{2}\right) \\
&= L_p^C(\lambda^*) - \sum_t A_t \left(\frac{\hat{\lambda}_t + \left|\hat{\lambda}_t\right|}{2}\right) - \sum_t B_t \left(\frac{\left|\hat{\lambda}_t\right| - \hat{\lambda}_t}{2}\right) - \sum_t \hat{\lambda}_t \left(E_p^C[f_t] - E_{\tilde{p}}[f_t]\right) \\
&\qquad + \sum_t A_t \left(\frac{\lambda_t^* + \left|\lambda_t^*\right|}{2}\right) + \sum_t B_t \left(\frac{\left|\lambda_t^*\right| - \lambda_t^*}{2}\right) + \sum_t \lambda_t^* \left(E_p^C[f_t] - E_{\tilde{p}}[f_t]\right) \qquad (4.16)
\end{aligned}$$

The only inequality above is by the definition of $\hat{\lambda}$ (4.13).

As (4.6) $-B_t \le E_{\tilde{p}}[f_t] - E_p[f_t] \le A_t$ implies the assumption that $-B_t \le E_{\tilde{p}}[f_t] - E_p^C[f_t] \le A_t$, it is easy to prove that for all *k* and $\hat{\lambda}_t \in \mathbb{R}$

$$-\sum_t A_t \left(\frac{\hat{\lambda}_t + \left|\hat{\lambda}_t\right|}{2}\right) - \sum_t B_t \left(\frac{\left|\hat{\lambda}_t\right| - \hat{\lambda}_t}{2}\right) - \sum_t \hat{\lambda}_t \left(E_p^C[f_t] - E_{\tilde{p}}[f_t]\right) \le 0$$

So

$$\begin{aligned}
L_p^C(\hat{\lambda}) &\le L_p^C(\lambda^*) + \sum_t A_t \left(\frac{\lambda_t^* + \left|\lambda_t^*\right|}{2}\right) + \sum_t B_t \left(\frac{\left|\lambda_t^*\right| - \lambda_t^*}{2}\right) + \sum_t \lambda_t^* \left(E_p^C[f_t] - E_{\tilde{p}}[f_t]\right) \\
&\le L_p^C(\lambda^*) + \sum_t \left|\lambda_t^*\right| (A_t + B_t) \qquad (4.17)
\end{aligned}$$

This is the bound. It follows that there exists non-asymptotic bounds showing that with respect to the true underlying distribution, Inequality MaxEnt produces *conditional* probability estimates that are almost as good as the best possible. These bounds are in terms of the deviation of the feature empirical averages relative to their true expectations, a number that can be bounded using standard uniform-convergence techniques. In particular, this leads to bounds that drop quickly with the number of samples, and that depend very moderately on the number of complexity of the features.

There is an important note for Inequality MaxEnt. For supervised learning, there is always a distribution that satisfies (4.6). But now for semi-supervised learning, it is not guaranteed that the feasible area of the optimization problem is not empty, if A_t and B_t are not set sufficiently large. However, the dual problem:

$$\text{minimize} \quad L(\alpha,\beta) = -\sum_t (\alpha_t - \beta_t) E_{\tilde{p}}[f_t] + \sum_i p(x_i)\log Z_i + \sum_t A_t\alpha_t + \sum_t B_t\beta_t \tag{4.18}$$

where $Z_i = \sum_k \exp\left(\sum_t (\alpha_t - \beta_t) f_t(x_i, y_k)\right)$ with optimization over $\alpha_t \geq 0, \beta_t \geq 0$

always has a nonempty feasible area, and a unique globally optimal solution due to convexity. So we are always licensed to solve the dual problem though it may not make sense in the primal problem. Experimental results show that this method also produces reasonable performance. Obviously, relaxing Inequality MaxEnt with 1-norm or 2-norm penalty (Eq. (3.26) to Eq. (3.34)) will always make the problem feasible for semi-supervised learning.

4.3 MaxEnt learning with side information

Sometimes, the only assumptions over the accuracy of empirical sufficient statistics for estimation are not enough to produce a good result. Suppose on a rectangular co-ordinate plane, all points in the first and third quadrants are positive while all points in the second and fourth quadrants are negative. Then with randomly distributed labeled data, the average of x and y co-ordinates for both positive and negative data are 0. This estimation of the average feature value is correct! But now, any random assignment for unlabeled data and testing data will not violate the assumption over average feature value. In other words, these assumptions make no contribution to correct classification. Adding some margins along the axis will not be of help either. The essence of the problem is: if the average of features for all classes is the same, then MaxEnt, including all smoothing variants, will not help. This classification task only exemplifies the insufficiency of pure MaxEnt, and other problems are also likely to exist.

To patch up the problem, it is recognized that in many existing literatures, adding assumptions is used as a basic approach for semi-supervised learning. We also wish to make our MaxEnt model flexible enough to incorporate assumptions. Actually the MaxEnt model does provide the flexibility to learn with side information. We formulate a MaxEnt model based on minimal spanning tree. All work in this section is newly done, except explicitly cited.

We denote the *distance* between x_i, x_j as $w_{(i,j)}$, where distance may be defined under certain assumptions. We find a minimal spanning tree (MST) based on $w_{(i,j)}$. Denote the set of edges as E. We add the assumption that nearby nodes, especially those neighboring nodes on MST, have similar probability of belonging to any class. We penalize their difference by adding the weighted square of difference in objective function. So the MaxEnt model is formulated as:

$$\text{minimize} \quad \sum_i p(x_i)\sum_k p(y_k \mid x_i)\log p(y_k \mid x_i) + \sum_t \frac{\sigma_t^2}{2}\delta_t^2 + \sum_{k,(i,j)\in E} w_{k,(i,j)}\varepsilon_{i,j,k}^2 \tag{4.19}$$

$$s.t. \quad E_{\tilde{p}}[f_t] - \sum_i p(x_i)\sum_k p(y_k \mid x_i) f_t(x_i, y_k) = \delta_t \quad \text{for all } t \tag{4.20}$$

$$\sum_k p(y_k \mid x_i) = 1 \quad \text{for all } i \tag{4.21}$$

$$p(y_k \mid x_i) - p(y_k \mid x_j) = \varepsilon_{i,j,k} \quad \text{for all } k \text{ and } (i,j) \in E \tag{4.22}$$

where $w_{k,(i,j)}$ is defined as $C_s \big/ w_{(i,j)}$ and C_s, σ_t are just a constant positive parameters. The dual problem is:

$$L(p_{\min}, \lambda, \gamma_{\max}, \alpha) = -\sum_t \lambda_t E_{\tilde{p}}[f_t] + \sum_i p(x_i) \log Z_i + \sum_t \frac{\lambda_t^2}{2\sigma_t^2} + \sum_{k,(i,j)\in E} \frac{\alpha_{k,(i,j)}^2}{4 w_{k,(i,j)}} \tag{4.23}$$

where $Z_i = \sum_k \exp\left(\sum_t \lambda_t f_t(x_i, y_k) - \frac{1}{p(x_i)} \left(\sum_{j_0:(i,j_0)\in E} \alpha_{k,(i,j_0)} - \sum_{i_0:(i_0,i)\in E} \alpha_{k,(i_0,i)} \right) \right)$.

Although it might cause numerical problem (dividing zero if $w_{(i,j)} = 0$ when one x appears for multiple times) in the primal problem, the definition of $w_{k,(i,j)}$ is always appropriate for dual problem (4.23).

The side information can also come from a variety of sources based on instance similarity. For example, *k*NN can be used even without changing the equations (4.19) to (4.22). The only implicit change lies in *E*, where now it includes all the edges in *k*NN graph. The distance metric is also unchanged. We observe that *k*NN MaxEnt is strongly connected to SGT. Recall the formulation of SGT (Joachims, 2003)which maximizes the normalized cut of a graph:

$$\max_{\vec{y}} \frac{cut(G^+, G^-)}{\left|\{i \mid y_i = 1\}\right| \left|\{i \mid y_i = -1\}\right|} \tag{4.24}$$

$$y_i = +1 \quad \text{if } x_i \text{ is positively labeled} \tag{4.25}$$

$$y_i = -1 \quad \text{if } x_i \text{ is negatively labeled} \tag{4.26}$$

$$\vec{y} \in \{+1, -1\}^n \tag{4.27}$$

where *n* is the total number of labeled and unlabeled examples. The denominator in (4.24) corresponds to the MaxEnt principle because they both introduce a tendency to assign balanced probability to all classes. The numerator can be approximated by $\sum_{k,(i,j)\in E} w_{k,(i,j)} \varepsilon_{i,j,k}^2$, where $\varepsilon_{i,j,k} = p(y_k \mid x_i) - p(y_k \mid x_j)$. If the two points x_i and x_j belong to the same class, their $p(y_k|x_i)$ and $p(y_k|x_j)$ are supposed to be close and their contribution to $\sum_{k,(i,j)\in E} w_{k,(i,j)} \varepsilon_{i,j,k}^2$ is also smaller. If they belong to different classes, then $p(y_k \mid x_i) - p(y_k \mid x_j)$ will be larger and thus greater contribution is made. We also notice that multi-class classification is also naturally allowed in our *k*NN MaxEnt, while SGT requires additional efforts for conversion.

Moreover, just like the scenario in co-training, multiple redundant descriptions can be utilized, such as image and voice for identifying a person, different description of the same event in different newspapers for training word sense disambiguation under the assumption that the same word in different articles has the same sense. Then each description can be used to build a good classifier. Now suppose we use *s* to index descriptions, and description *s* has features f_t^s. The edge set *E* becomes links between all representations for the same object.

$$\text{minimize} \quad \sum_s \left(\sum_i p_s(x_i) \sum_k p_s(y_k \mid x_i) \log p_s(y_k \mid x_i) + \sum_t \frac{\sigma_{s,t}^2}{2} \delta_{s,t}^2 \right) + \sum_{k,(i,j)\in E} w_{k,(i,j)} \varepsilon_{i,j,k}^2 \tag{4.28}$$

$$s.t. \qquad E_{\tilde{p}}[f_t^s] - \sum_i p_s(x_i) \sum_k p_s(y_k \mid x_i) f_t^s(x_i, y_k) = \delta_t \quad \text{for all } s, t \tag{4.29}$$

$$\sum_k p_s(y_k \mid x_i) = 1 \quad \text{for all } i,\ s \tag{4.30}$$

$$p_{s_1}(y_k \mid x_i) - p_{s_2}(y_k \mid x_j) = \varepsilon_{i,j,k} \quad \text{for all } k \text{ and } (i, j) \in E \tag{4.31}$$

Another generic approach for applying MaxEnt is to add entropy term to the objective function of another model. For example, we can use MaxEnt to derive a normalized version of Markov random walk (Szummer & Jaakkola, 2001a). If we have a set of points $\{x_1, x_2, ..., x_N\}$ with first L instances labeled $\{ \tilde{y}_1, ..., \tilde{y}_L \}$ from C classes. A metric $d(x_i, x_j)$ is also defined. Then define $W_{ij} = \exp(-d(x_i, x_j)/\sigma)$ if x_i is among the k nearest neighbors of x_k or x_k is among the k nearest neighbors of x_i. Otherwise, $W_{ij} = 0$. Define $p_{ik} = W_{ik} \big/ \sum_j W_{ij}$ and construct matrix A whose (i, k)-th entry is p_{ik}. So $[A^t]_{ik}$ is used to define the t step transition probability, meaning the probability of randomly walking from x_i to x_k in t steps, denoted as $p_{t|0}(k \mid i)$. Suppose the prior probability of x_i belonging to class y is $P(y|i)$, then the posterior probability of x_k belonging to y is $P_{post}(y \mid k) = \sum_{i=1}^{N} P(y \mid i) P_{o|t}(i \mid k)$. Intuitively, if we define the margin of the classifier on labeled data k and class d to be $\gamma_{kd} = P_{post}(y = \tilde{y}_k \mid k) - P_{post}(y = d \mid k)$, then for correct classification the margin should be nonnegative for all classes d other than $\tilde{y}_k$ ($\gamma_{kd} \geq 0$) and zero for the correct class ($\gamma_{k\tilde{y}_k} = 0$). Now using maximum margin principle, the problem can be formulated as:

$$\text{maximize:} \qquad \sum_{k=1}^{L} \sum_{d=1}^{C} \frac{1}{N_{C(k)}} \gamma_{kd} \tag{4.32}$$

$$s.t. \qquad P_{post}(y = \tilde{y}_k \mid k) \geq P_{post}(y = d \mid k) + \gamma_{kd} \quad \forall k \in 1...L,\ \forall d \in 1...C \tag{4.33}$$

$$\sum_{c=1}^{C} P(y = c \mid i) = 1 \ \text{ and } 0 \leq P(y \mid i) \leq 1 \quad \forall i \in 1...N \tag{4.34}$$

The optimization is over $P(y|i)$ and $N_{C(k)}$ is the number of labeled examples in the same class as x_k. This is the original formulation in (Szummer & Jaakkola, 2001a).

To adapt the formulation for MaxEnt, we just need to add MaxEnt in objective function and express $P(y|i)$ in terms of a combination of features. The effect is similar to regularization.

$$\text{minimize:} \quad \sum_i p(x_i) \sum_k p(y_k \mid x_i) \log p(y_k \mid x_i) + \sum_t \frac{\sigma_t^2}{2} \delta_t^2 - C_1 \sum_{i=1}^{L} \sum_{k=1}^{C} \frac{1}{N_{C(i)}} \gamma_{ik} \tag{4.35}$$

$$s.t. \qquad E_{\tilde{p}}[f_t] - \sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k) = \delta_t \quad \text{for all } t \tag{4.36}$$

$$\sum_k p(y_k \mid x_i) = 1 \quad \text{for all } i \tag{4.37}$$

$$P_{post}(y = \tilde{y}_i \mid i) \geq P_{post}(y = y_k \mid i) + \gamma_{ik} \quad \forall i \in 1...L,\ \forall k \in 1...C \tag{4.38}$$

$$\sum_{c=1}^{C} P(y = y_k \mid i) = 1 \ \text{ and } 0 \leq P(y = y_k \mid i) \leq 1 \quad \forall i \in 1...N \tag{4.39}$$

In sum, MaxEnt provides nice flexibility to incorporate standard assumptions by modifying objective functions and constraints. It is thus highly worthwhile to investigate this generic picture in the project.

4.4 Miscellaneous promising research openings

To make the MaxEnt work, there are a lot of miscellaneous problems to be considered.

1) A fundamental problem for maximum entropy is what features we should use. How should we decide which features to include, in order to avoid overfitting and running out of memory but still at a reasonable computational cost of searching? (Pietra et al., 1997) used a greedy algorithm to incrementally add feature to the random field by selecting the feature which maximally reduces the objective function (KL divergence between current model and empirical distribution). (McCallum, 2003) used a similar principle, but in conditional random fields (Lafferty et al., 2001), to iteratively construct feature conjunctions that would significantly increase conditional log-likelihood if added to the model. Automated feature induction enables not only improved accuracy and dramatic reduction in parameter count, but also use larger cliques in a graphic model view, and more freedom to liberally hypothesize atomic input variables that may be relevant to the task. In this project, it is also worthwhile to explore utilization of standard search algorithms for feature selection, including stochastic approaches. Also the form of candidate features requires consideration. If a feature does not appear in labeled examples but does appear in unlabeled examples, how should we learn from this situation? We initially propose disjunctive features as candidates. For example, if *IBM* does not appear in labeled data while *Apple* appears, we may use '*IBM* **or** *Apple*' as a feature. Of course this will enlarge the feature space from $O(n)$ to $O(n^2)$ and may grow beyond normal computation ability. But with the sequential-update algorithm (Collins et al., 2002), a coordinate-wise descent style algorithm, such MaxEnt parameter estimation problems for very large (or infinite) number of features are feasible.

2) Another problem is how to estimate A_t and B_t for Inequality MaxEnt, called interval estimation. In other words, how accurate the $E_{\tilde{p}}[f_t]$ is as an estimation of $E_p[f_t] = \sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k)$. According to Hoeffding's inequality, if there are m labeled data, the probability that $\left| E_p[f_t] - E_{\tilde{p}}[f_t] \right| > \beta$ is at most $\exp(-2\beta^2 m)$. So the general interval should be of size $O(n^{-1/2})$. If for a feature t, its being 1 almost certainly indicates that the class is y_k, then the estimation can probably be tighten to $O(1/n)$. It is worth deriving a better interval than Chernoff or Hoeffding bounds, by using probably binomial or advanced concepts in statistical learning theory, e.g., VC-dimension of feature class.

3) Furthermore, if we pay attention to fact that labeled data are more important than unlabeled data, then like (Blum & Chawla, 2001) which tried assigning lower weights to edges between unlabeled examples, higher weights can be attached to labeled than to unlabeled data. We call this method re-weighting and we adopt the idea for MaxEnt requiring as little change as possible. To be specific, the constraint in (4.20): $E_{\tilde{p}}[f_t] - \sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k) = \delta_t$ for all t is not meaningful if the empirical estimation of f_t is very deviated from the correct one. However, if we attach higher weight to labeled data and lower weight to unlabeled data, then $E_{\tilde{p}}[f_t]$ will become a more accurate estimate for $E_p[f_t] = \sum_i p(x_i) \sum_k p(y_k \mid x_i) f_t(x_i, y_k)$.

In detail, suppose there are n_1 labeled examples and n_2 unlabeled examples. We can assume that all examples are different because even when there are duplicate examples, we can safely perturb them by a sufficiently small amount ε. We wish to give extra β times of weight to labeled data compared with unlabeled data. Originally, we have $\underbrace{x_1^l, x_2^l, \ldots, x_{n_1}^l}_{n_1 \text{ labeled data}}, \underbrace{x_1^u, x_2^u, \ldots, x_{n_2}^u}_{n_2 \text{ unlabeled data}}$.

Now we have $\underbrace{\overbrace{x_1^l, \ldots x_1^l}^{\substack{\beta \text{ copies} \\ \text{of } x_1^l}}, \ldots, \overbrace{x_{n_1}^l, \ldots x_{n_1}^l}^{\substack{\beta \text{ copies} \\ \text{of } x_{n_1}^l}}}_{\beta n_1 \text{ labeled data}}, \underbrace{x_1^u, x_2^u, \ldots, x_{n_2}^u}_{n_2 \text{ unlabeled data}}$. So the $p(x)$ for labeled data has changed from $\frac{1}{n_1 + n_2}$ to $\frac{\beta}{\beta n_1 + n_2}$ and the $p(x)$ for unlabeled data has changed from $\frac{1}{n_1 + n_2}$ to $\frac{1}{\beta n_1 + n_2}$. In such a formulation, the equations of MaxEnt (and all its variants) do not need to be modified at all.

Now a new problem arises. Each term for regularization introduces a regularization parameter, which represents the tradeoff between entropy maximization and smoothing. With an additional parameter for re-weighting (and possibly even more if we extend the MaxEnt in this fashion), the parameter estimation will become a problem, given the limited amount of labeled instances for cross validation. One possible solution may be similar to (Lee & Liu, 2003), which approximated the F score maximization by maximizing recall^2/P(model predicting positive). The latter term can be more reliably estimated from the training data.

4) Finally, to test the efficacy of models, it is necessary to experiment on different datasets. There are numerous online data repositories providing benchmark dataset for machine learning research. For general purposes, the UCI Repository (http://www.ics.uci.edu/~mlearn/MLRepository.html) and Delve (http://www.cs.toronto.edu/~delve/) are two commonly used sources of experimental data. Of course far more are available on Internet.

For text classification tasks, the most commonly used datasets are:

- 20 Newsgroup. The 20 newsgroups collection, originally collected by Ken Lang (Lang, 1995), has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering. It is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. Many of the categories fall into confusable clusters, e.g., 5 of them are comp.* discussion groups. A good web site with preprocessed data is:
  http://people.csail.mit.edu/u/j/jrennie/public_html/20Newsgroups/.
- Reuters-21578 dataset, Distribution 1.0. The dataset is a collection of labeled newswire articles in 1987, developed by David D. Lewis. It consists of 12902 articles and 90 topic categories. The dataset can be found at:
  http://www.daviddlewis.com/resources/testcollections/reuters21578/.
- WebKB (Craven et al., 1998). It contains 8145 web pages gathered from university computer science departments. The collection includes the entirety of four departments, and additionally, an assortment of pages from other universities. The pages are divided into seven categories: *student, faculty, staff, course, project, department* and *other*. The data is available at: http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/.
- OHSUMED (Hersh et al., 1994). This dataset is a clinically-oriented MEDLINE subset, consisting of 348,566 references (out of a total of over 7 million), covering all references

from 270 medical journals over a five-year period (1987-1991). The fields present include the title, abstract, MeSH indexing terms, author, source, and publication type. The class label is MeSH indexing terms. A source of the dataset is: http://www.mlnet.org/cgi-bin/mlnetois.pl/?File=dataset-details.html&Id=940411893OHSUMED.

4.5 Initial experimental results

Initial experiment is done for MST MaxEnt with Gaussian prior on optical digits dataset from UCI repository. This dataset deals with optical recognition of handwritten digits. There are 64 input attributes ranging in [0, 16] and 10 classes. The number of examples is nearly the same for all classes in both training and testing data. As data preprocessing, all examples are normalized to have length 1. Due to the nature of MST MaxEnt, all testing data are automatically included as unlabeled data. In this experiment, the number of testing data is fixed at 1687. This number is NOT included in the second column (*No. of Unlabeled Data*), i.e., there are still 1687 unlabeled data involved in the learning even when the column shows 0.

The resulting accuracy is given in Table 1. The *Gaussian regularization factor* stands for the σ_t in (4.19), which are made the same for all *t*. C_s is the parameter in the definition of $w_{k,(i,j)}$ in (4.19). *Re-weighted MST MaxEnt accuracy* is obtained by optimally adjusting the β in re-weighting formulation. The last three columns show the highest accuracy given by tuning model parameters, i.e., results for the parameter setting with the best performance on the test set. The *Gaussian MaxEnt accuracy* and *Inequality MaxEnt accuracy* are the best result by Gaussian MaxEnt ((3.13) to (3.15)) and Inequality MaxEnt ((3.22) to (3.24)) respectively. For TSVM, linear kernel and polynomial kernel are tried and one-against-all heuristic is adopted for multi-class classification.

No. of Labeled data	No. of Unlabeled Data	Gaussian regularization factor	C_s for MST side info	Re-weighted MST MaxEnt Accuracy	Gaussian MaxEnt Accuracy	Inequality MaxEnt Accuracy	TSVM Result
39	3894	10	10^{-5}	93.8352	59.57	58.80	73.444
39	0	10	10^{-3}	85.5957	77.53	77.59	
78	3855	10	10^{-5}	94.7244	79.37	70.59	84.766
78	0	10	10^{-5}	88.2039	87.72	86.01	
117	3816	10	10^{-5}	94.8429	84.77	77.53	84.766
117	0	10	10^{-5}	91.8791	89.27	88.20	
156	3777	0.1	10^{-5}	95.3764	87.31	81.21	90.279
156	0	10	10^{-5}	92.4718	90.10	89.27	
196	3737	10	10^{-5}	96.5027	89.27	85.12	89.627
196	0	10	10^{-5}	91.7012	92.95	90.75	

Table 1. Optimal accuracy for optical digit dataset

The resulting accuracy clearly shows that this model of MST MaxEnt with re-weighting is promising.

Chapter 5 Conclusion

In many practical applications of data classification and data mining, one finds a wealth of easily available unlabeled examples, while collecting labeled examples can be costly and time-consuming. This is especially true for text classification and it is of interest to develop algorithms that are able to utilize both labeled and unlabeled data for classification.

Although a number of such semi-supervised learning algorithms have been designed, most of them are dependent on certain assumptions, mainly generative model assumption, clustering assumptions over distance metrics or other similarity measures. They may be well compliant with the real dataset or may not hold at all. Therefore, an algorithm is desired with as little dependence on such assumptions as possible, or using weakest possible assumptions.

The maximum entropy model provides a generic framework that meets this requirement, with its weak statistical assumptions concerning the reliability of empirical feature expectation. It also provides a natural mechanism for multi-class classification. With its dual problem being maximum likelihood estimation, there are numerous regularization techniques proposed to overcome overfitting. The MaxEnt optimization problem enjoys another advantage of being convex, ensuring the existence of a unique global optimum. Several algorithms for MaxEnt convex optimization are also proposed, with different performance and constraints.

The project aims to implement semi-supervised learning in MaxEnt models. As the original MaxEnt model does not perform satisfactorily, side information is considered to be added into MaxEnt, in forms of minimal spanning tree, *k*-nearest neighbor, multi-representation of one example, etc. Our initial experimental result suggests that MaxEnt with side information is a promising tool.

Future work is proposed to be done in following areas:

- Feature selection/induction for or by MaxEnt model;
- Effective and efficient model parameter estimation;
- Different formulations for side information incorporation;
- Derivation of necessary sample size and generalization bound using statistical learning theory.

References

Apte, C., Damerau, F., & Weiss, S. M. (1998). Text mining with decision trees and decision rules. *Workshop on Learning from Text and the Web, Conference on Automated Learning and Discovery* .

Apte, C., Damerau, F., & Weiss, S. M. (1994). Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems* 12[3], 233-251.

Argamon-Engelson, S. & Dagan, I. (1999). Committee-based sample selection for probabilistic classifiers. *Journal of Artificial Intelligence Research* 11, 335-360.

Baluja, S. (1999). Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled examples. *Advances in Neural Information Processing Systems* 11, 854-860.

Belkin, M. & Niyogi, P. (2002). Semi-supervised learning on manifolds. *Technical Report TR-2002-12, Computer Science Department, The University of Chicago* .

Belkin, M. & Niyogi, P. (2004). Semi-Supervised Learning on Riemannian Manifolds. *Machine Learning* 56, 209-239.

Bennett, K. & Demiriz, A. (1999). Semi-supervised support vector machines. *Advances in Neural Information Processing Systems* 11, 368-374.

Berger, A., Pietra, S. D., & Pietra, V. D. (1996). A ME approach to natural language processing. *Computational Linguistics* 22, 39-71.

Blum, A. & Chawla, S. (2001). Learning from Labeled and Unlabeled Data using Graph Mincuts. *Proceedings of 18th International Conference on Machine Learning* , 19-26.

Blum, A. & Mitchell, T. (1998). Combining labeled and unlabeled data with cotraining. *Proceedings of the 11th Annual Conference on Computational Learning Theory* , 92-100.

Boykov, Y., Veksler, O., & Zabih, R. (1998). Markov Random Fields with Efficient Approximations. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* , 648-655.

Castelli, V. & Cover, T. M. (1995). On the exponential value of labeled samples. *Pattern Recognition Letters* 16[1], 105-111.

Castelli, V. & Cover, T. M. (1996). The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory* 42[6], 2101-2117.

Cataltepe, Z. & Magdon-Ismail, M. (1998). Incorporating test inputs into learning. *Advances in Neural Information Processing Systems* 10, 437-443.

Chapelle, O., Weston, J., & Schoelkopf, B. (2002). Cluster kernels for semi-supervised learning. *Advances in Neural Information Processing Systems* 15, 585-592.

Cheeseman, P. et al. (1988). AutoClass: A Bayesian classification system. *Machine Learning: Proceedings of the Fifth International Conference* , 54-64.

Cheeseman, P. & Stutz, J. (1996). Bayesian classification (AutoClass): Theory and results. *Advances in knowledge discovery and data mining* . MIT Press.

Chen, J. (1995). Optimal rate of convergence for finite mixture models. *The Annals of Statistics* 23[1], 221-233.

Chen, S. F. & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech and Language* 13, 359-394.

Chen, S. F. & Rosenfeld, R. (2000). A Survey of Smoothing Techniques for ME Models. *IEEE Transactions on Speech and Audio Processing* 8[1], 37-50.

Cohen, W. W. & Hirsch, H. (1998). Joins that generalize: text categorization using WHIRL. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining* , 169-173.

Cohen, W. W. & Singer, Y. (1996). Context-sensitive learning methods for text categorization. *SIGIR '96: Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* , 307-315.

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine Learning* 15[2], 201-221.

Cohn, D., Ghahramani, Z., & Jordan, M. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research* 4, 129-145.

Collins, M. & Singer, Y. (1999). Unsupervised models for named entity classification. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* .

Collins, M. C., Shapire, R. E., & Singer, Y. (2002). Logistic Regression, AdaBoost and Bregman Distances. *Machine Learning* 48, 253-285.

Cormen, T. H. et al. (2001). Introduction to Algorithms, Second Edition. MIT Press.

Cozman, F. G. & Cohen, I. (2002). Unlabeled Data Can Degrade Classification Performance of Generative Classifiers. *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference* , 327-331. AAAI Press.

Craven, M. et al. (2000). Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence* 118[1-2], 69-113.

Craven, M., Slattery, S., & Nigam, K. (1998). First-order learning for web mining. *Proceedings of the 10th European Conference on Machine Learning* , 250-255.

Darroch, J. N. & Ratcliff, D. (1972). Generalized Iterative Scaling for Log-Linear Models. *The Annals of Mathematical Statistics* 43[5], 1470-1480.

Day, N. E. (1969). Estimating the components of a mixture of normal distributions. *Biometrika* 56[3], 463-474.

Deming, W. E. & Stephan, F. F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginals are known. *Annals of Mathematical Statistics* 11, 427-444.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39[1], 1-38.

Dudik, M., Phillips, S. J., & Schapire, R. E. (2004). Performance Guarantees for Regularized Maximum Entropy Density Estimation. *Proceedings of the 17th Annual Conference on Computational Learning Theory* .

Dumais, S. T. et al. (1998). Inductive learning algorithms and representations for text categorization. *Proceedings of the Seventh International Conference on Information and Knowledge Management* , 148-155.

Freund, Y. et al. (1997). Selective sampling using the query by committee algorithm. *Machine Learning* 28[2/3], 133-168.

Furnkranz, J., Mitchell, T., & Riloff, E. (1998). A case study in using linguistic phrases for text categorization on the WWW. *Learning for Text Categorization: Papers from the AAAI Workshop, Tech.rep.WS-98-05* , 5-12. AAAI Press.

Ganesalingam, S. (1989). Classification and mixture approaches to clustering via maximum likelihood. *Applied Statistics* 38[3], 455-466.

Ganesalingam, S. & McLachlan, G. J. (1978). The efficiency of a linear discriminant function based on unclassified initial samples. *Biometrika* 65, 658-662.

Geman, S., Bienstock, E., & Doursat, R. (1992). Neural networks and bias/variance dilemma. *Neural Computation* , 1-58.

Ghahramani, Z. & Jordan, M. I. (1994). Supervised learning from incomplete data via an EM approach. *Advances in Neural Information Processing Systems* 6, 120-127.

Goldman, S. & Zhou, Y. (2000). Enhancing supervised learning with unlabeled data. *Proceedings of the Seventeenth International Conference on Machine Learning* .

Goodman, J. (2004). Exponential Priors for Maximum Entropy Models. *Microsoft Research, Available at:(http://research.microsoft.com/~joshuago/exponentialprior-final.pdf)* .

Greig, D., Porteous, B., & Seheult, A. (1989). Exact maximum a posteriori estimation for binary images. *Journal of Royal Statistical Society, Series B* 51, 271-279.

Hartley, H. O. & Rao, J. N. K. (1968). Classification and estimation in analysis of variance problems. *Review of International Statistical Institute* 36, 141-147.

Hersh, W. R. et al. (1994). OHSUMED: an interactive retrieval evaluation and new large test collection for research. *SIGIR 94: Proceedings of the 17th Annual ACM SIGIR Conference on Research and Development in Information Retrieval* , 192-201.

Hofmann, T. & Puzicha, J. (1998). *Statistical models for co-occurrence data. Tech.rep.AI Memo 1625.Articial Intelligence Laboratory, MIT.*

Jaakkola, T. & Haussler, D. (1998). Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing Systems* 11, 487-493.

Jaakkola, T., Meila, M., & Jebara, T. (2000). Maximum entropy discrimination. *MIT Technical Report AITR-1668* .

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physics Reviews* 100, 620-630.

Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. *Proceedings of the Fourteenth International Conference Machine Learning* , 143-151.

Joachims, T. (2003). Transductive Learning via Spectral Graph Partitioning. *Proceeding of The Twentieth International Conference on Machine Learning* .

Joachims, T. (1998a). Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98, Tenth European Conference on Machine Learning* , 137-142.

Joachims, T. (1999). Transductive Inference for Text Classification using Support Vector Machines. *Proceedings of the Sixteenth International Conference on Machine Learning* .

Joachims, T. (1998b). Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98, Tenth European Conference on Machine Learning* , 137-142.

Jordan, M. I. & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation* 6[2], 181-214.

Kazama, J. & Tsujii, J. (2003). Evaluation and Extension of Maximum Entropy Models with Inequality Constraints. *Proceedings of the 2003 Conference on Empirical Methods in NLP* .

Khudanpur, S. (1995). A method of ME estimation with relaxed constraints. *Johns Hopkins Univ.Language Modeling Workshop* , 1-17.

Kneser, R. & Ney, H. (1995). Improved backing-off for m-gram language modeling. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* , 181-184.

Koller, D. & Sahami, M. (1997). Hierarchically classifying documents using very few words. *Proceedings of the Fourteenth International Conference Machine Learning* , 170-178.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceeding of the 18th International Conference on Machine Learning* , 282-289.

Lang, K. (1995). NewsWeeder: Learning to filter netnews. *Machine Learning: Proceedings of the Twelfth International Conference* , 331-339.

Lebanon, G. & Lafferty, J. (2001). Boosting and maximum Likelihood for Exponential Models. *CMU Technical Report CMU-CS-01-144* .

Lee, W. S. & Liu, B. (2003). Learning with Positive and Unlabeled Examples Using Weighted Logistic Regression. *Proceedings of 20th International Conference on Machine Learning* , 448-455.

Lewis, D. D. (1995). A sequential algorithm for training text classifiers: Corrigendum and additional data. *SIGIR Forum* 29[2], 13-19.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. *ECML-98, Tenth European Conference on Machine Learning* , 4-15.

Lewis, D. D. & Gale, W. A. (1994). A sequential algorithm for training text classifiers. *SIGIR '94: Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* , 3-12.

Lewis, D. D. & Knowles, K. A. (1997). Threading electronic mail: A preliminary study. *Information Processing and Management* 33[2], 209-217.

Li, H. & Yamanishi, K. (1997). Document classification using a finite mixture model. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* , 39-47.

Li, W. & McCallum, A. (2004). A Note on Semi-supervised Learning using Markov Random Fields. *Technical Note, available at: http://www.cs.umass.edu/~mccallum/papers/li-ssmrf.pdf* .

Liere, R. (1999). *Active learning with committees: An approach to efficient learning in text categorization using linear threshold algorithms. Doctoral dissertation, Department of Computer Science, Oregon State University.*

Little, R. J. A. (1977). Discussion on the paper by Professor Dempster, Professor Laird and Dr. Rubin. *Journal of the Royal Statistical Society, Series B* 39[1], 25.

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. *Sixth Conference on Natural Language Learning* .

McCallum, A. (2003). Efficiently Inducing Features of Conditional Random Fields. *Nineteenth Conference on Uncertainty in Artificial Intelligence* .

McCallum, A. & Nigam, K. (1998a). Employing EM in pool-based active learning for text classification. *Proceedings of the Fifteenth International Conference Machine Learning* , 350-358.

McCallum, A. & Nigam, K. (1998b). A comparison of event models for naive Bayes text classification . *Learning for Text Categorization: Papers from the AAAI Workshop, Tech.rep.WS-98-05* , 41-48. AAAI Press.

McLachlan, G. & Basford, K. (1988). *Mixture models*. New York, Marcel Dekker.

McLachlan, G. & Peel, D. (2000). *Finite mixture models*. New York, John Wiley and Sons.

McLachlan, G. J. (1975). Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association* 70[350], 365-369.

McLachlan, G. J. & Ganesalingam, S. (1982). Updating a discriminant function on the basis of unclassified data. *Communications in Statistics: Simulation and Computation* 11[6], 753-767.

McLachlan, G. J. & Krishnan, T. (1997). *The EM algorithm and extensions*. New York, John Wiley and Sons.

Miller, D. J. & Uyar, H. (1996). A generalized Gaussian mixture classier with learning based on both labeled and unlabelled data. *Proceedings of the 1996 Conference on Information Science and Systems* .

Mitchell, T. M. (1997). *Machine learning*. New York, McGraw-Hill.

Mladenic, D. (1998). *Machine learning on non-homogeneous, distributed text data. Doctoral dissertation, Faculty of Computer and Information Science, University of Ljubljana, Slovenia.*

Mooney, R. J. & Roy, L. (2000). Content-based book recommending using learning for text categorization. *Proceedings of the Fifth ACM Conference on Digital Libraries* , 195-204.

Moulinier, I., Raskinis, G., & Ganascia, J.-G. (1996). Text categorization: a symbolic approach. *Fifth Annual Symposium on Document Analysis and Information Retrieval* , 87-99.

Murray, G. D. & Titterington, D. M. (1978). Estimation problems with data from a mixture. *Applied Statistics* 27[3], 325-334.

Muslea, I. et al. (2003). Active learning with strong and weak views: a case study on wrapper induction. *Proceedings of 2003 International Conference on Artificial Intelligence* , 415-420.

Newman, W. (1977). Extension to the ME method. *IEEE Transactions on Information Theory* IT-23, 89-93.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On Spectral Clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems* 14.

Nigam, K. & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. *Ninth International Conference on Information and Knowledge Management* , 86-93.

Nigam, K., Lafferty, J., & McCallum, A. (1999a). Using maximum entropy for text classification. *IJCAI-99 Workshop on Machine Learning for Information Filtering* , 61-67.

Nigam, K. et al. (1998). Learning to classify text from labeled and unlabeled documents. *Proceedings of the Fifteenth National Conference on Artificial Intelligence* , 792-799.

Nigam, K. et al. (1999b). Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning* 39[2/3], 103-134.

Nigam, K. P. (2001). Using unlabeled data to improve text classification. *PhD Thesis.Carnegie Mellon University* .

O'Neill, T. J. (1978). Normal discrimination with unclassified observations. *Journal of the American Statistical Association* 73[364], 821-826.

Pazzani, M. J., Muramatsu, J., & Billsus, D. (1996). Syskill & Webert: Identifying interesting Web sites. *Proceedings of the Thirteenth National Conference on Artificial Intelligence* , 54-59.

Phillips, S. J., Dudik, M., & Schapire, R. E. (2004). A maximum entropy approach to species distribution modeling. *Proceedings of Twenty-first international conference on Machine learning* .

Pietra, S. D., Pietra, V. D., & Lafferty, J. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 380-393.

Pietra, S. D., Pietra, V. D., & Lafferty, J. (2002). Duality and Auxiliary Functions for Bregman Distances. *CMU Technical Report CMU-CS-01-109R* .

Ratnaparkhi, A. (1998). ME models for natural language ambiguity resolution. *Ph.D.dissertation, Univ.Pennsylvania, Philadelphia, PA* .

Ratsaby, J. & Venkatesh, S. S. (1995). Learning from a mixture of labeled and unlabeled examples with parametric side information. *Proceedings of the Eighth Annual Conference on Computational Learning Theory* , 412-417.

Riloff, E. & Jones, R. (1999). Learning dictionaries for information extraction using multi-level boot-strapping. *Proceedings of the Sixteenth National Conference on Artificial Intelligence* , 474-479.

Rodriguez, M., Gomez-Hidalgo, J. M., & Agudo, B. D. (1997). Using WordNet to complement training information in text categorization. *Proceedings of the International Conference on Recent Advances in Natural Language Processing* , 150-157.

Rosenfeld, R. (1996). A ME approach to adaptive statistical language modeling. *Computer, Speech and Language* 10, 187-228.

Sahami, M. (1996). Learning limited dependence Bayesian classifiers. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* , 335-338.

Sahami, M. et al. (1998). A Bayesian approach to filtering junk e-mail. *Learning for Text Categorization: Papers from the AAAI Workshop, Tech.rep.WS-98-05* , 55-62.

Salton, G. & Buckley, C. (1998). Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management* 24, 513-523.

Schapire, R. E. & Singer, Y. (2000). BoosTexter: A boosting-based system for text categorization. *Machine Learning 39*[2/3], 135-168.

Schohn, G. & Cohn, D. (2000). Less is more: Active learning with support vector machines. *Proceedings of the Seventeenth International Conference on Machine Learning* .

Schuurmans, D. (1997). A new metric-based approach to model selection. *Proceedings of the Fourteenth National Conference on Artificial Intelligence* , 552-558.

Schuurmans, D. & Southey, F. (2000). An adaptive regularization criterion for supervised learning. *Proceedings of the Seventeenth International Conference on Machine Learning* .

Scott, S. & Matwin, S. (1998). Text classification using WordNet hypernyms. *Usage of WordNet in Natural Language Processing Systems: Proceedings of the Workshop* , 45-52.

Sebastiani, F., Sperduti, A., & Valdambrini, N. (2000). An improved boosting algorithm and its application to text categorization. *Proceedings of the Ninth International Conference on Information and Knowledge Management* , 78-85.

Seung, H., Opper, M., & Sompolinsky, H. (1992). Query by committee. *Machine Learning: Proceedings of the Fifth International Conference* , 287-294.

Shahshahani, B. & Landgrebe, D. (1994). The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing* 32[5], 1087-1095.

Shavlik, J. & Eliassi-Rad, T. (1998). Intelligent agents for web-based tasks: An advice-taking approach. *Learning for Text Categorization: Papers from the AAAI Workshop* , 63-70.

Szummer, M. & Jaakkola, T. (2001b). Kernel expansions with unlabeled data. *Advances in Neural Information Processing Systems* 13.

Szummer, M. & Jaakkola, T. (2001a). Partially labeled classification with Markov random walks. *Advances in Neural Information Processing Systems* 14. MIT Press.

Titterington, D. M. (1976). Updating a diagnostic system using unconfirmed cases. *Applied Statistics* 25[3], 238-247.

Tong, S. & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Proceedings of the Seventeenth International Conference on Machine Learning* .

Tsuda, K., Kin, T., & Asai, K. (2002). Marginalized kernels for biological sequences. *Bioinformatics* 18[90001].

Vapnik, V. (1998). *Statistical learning theory*. New York, John Wiley and Sons.

Watanabe, S. (1969). *Knowing and guessing: A quantitative study of inference and information*. New York, John Wiley and Sons.

Weiss, Y. (1999). Segmentation Using Eigenvectors: A Unifying View. *International Conference on Computer Vision* , 975-982.

Wiener, E., Pedersen, J. O., & Weigend, A. S. (1995). A neural network approach to topic spotting. *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval* , 317-332.

Williams, P. M. (1995). Bayesian regularization and pruning using a Laplace prior. *Neural Computation* 7, 117-143.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval* 1[1/2], 67-88.

Yang, Y. & Chute, C. G. (1994). An example-based mapping method for text classification and retrieval. *ACM Transactions on Information Systems* 12[3], 252-277.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics* , 189-196.

Yu, S. X., Gross, r., & Shi, J. (2002). Concurrent Object Recognition and Segmentation by Graph Partitioning. *Advances in Neural Information Processing Systems* 15, 1383-1390.

Zhang, T. & Oles, F. J. (2000). A probability analysis on the value of unlabeled data for classification problems. *Proceedings of the Seventeenth International Conference on Machine Learning* , 1191-1198.