Using EM to Classify Text from Labeled and Unlabeled Documents

Kamal Nigam   Andrew McCallum   Sebastian Thrun
Tom Mitchell

May 11, 1998

CMU-CS-98-120

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper shows that the accuracy of learned text classifiers can be improved by augmenting a small number of labeled training documents with a large pool of unlabeled documents. This is significant because in many important text classification problems obtaining classification labels is expensive, while large quantities of unlabeled documents are readily available. We present a theoretical argument showing that, under common assumptions, unlabeled data contain information about the target function. We then introduce an algorithm for learning from labeled and unlabeled text, based on the combination of Expectation-Maximization with a naive Bayes classifier. The algorithm first trains a classifier using the available labeled documents, and probabilistically labels the unlabeled documents. It then trains a new classifier using the labels for all the documents, and iterates. Experimental results, obtained using text from three different real-world tasks, show that the use of unlabeled data reduces classification error by up to 30%.

This research has been supported in part by the DARPA HPKB program under research contract F30602-97-1-0215.

1. Introduction

Consider the problem of training a computer to automatically classify text documents. Given the growing volume of online text available through the World Wide Web, Internet news feeds, electronic mail, and digital libraries, this problem is of great practical significance. There are statistical text learning algorithms that can be trained to approximately classify documents, given a sufficient set of labeled training examples. These text classification algorithms have been used to automatically catalog news articles [Lewis and Gale, 1994; Joachims, 1997b] and web pages [Craven *et al.*, 1998], automatically learn the reading interests of users [Pazzani *et al.*, 1996; Lang, 1995], and automatically sort electronic mail [Lewis and Knowles, 1997].

One key difficulty with these current algorithms, and the issue addressed by this paper, is that they require a large, often prohibitive, number of labeled training examples to learn accurately. Take, for example, the task of learning which newsgroup articles are of interest to a person reading UseNet news, as examined by Lang [Lang, 1995]. After reading and classifying about 1000 articles, precision of the learned classifier was about 50% for the top 10% of documents ranked by the classifier. As a practical user of such a filtering system, one would obviously prefer learning algorithms that can provide accurate classifications after hand-labeling only a few dozen articles, rather than thousands.

In this paper we describe an algorithm that learns to classify text documents more accurately by using *unlabeled* documents to augment the available *labeled* training examples. In our example, the labeled documents might be just 10 articles that have been read and judged by the user as interesting or not. Our learning algorithm can make use of the vast multitude of unlabeled articles available on UseNet to augment these 10 labeled examples. In many text domains, especially those involving online sources, collecting unlabeled examples is trivial; it is the labeling that is expensive.

We present experimental results showing that this unlabeled data can boost learning accuracy in three text classification domains: newsgroup articles, web pages, and newswire articles. For example, to identify the source newsgroup for a UseNet article with 70% classification accuracy, our algorithm takes advantage of 10,000 unlabeled examples and requires only 300 labeled examples; on the other hand, a traditional learner requires 1000 labeled examples to achieve the same accuracy. Thus, in this task, the technique reduces the need for labeled training examples by a factor of three.

Why do unlabeled examples boost learning accuracy? In brief, they provide information about the joint probability distribution over words within the documents. Suppose, for example, that using only the labeled data we determine that documents containing the word "learn" tend to belong to the positive class. If we use this fact to estimate the classification of the many unlabeled documents, we might find that the word "teach" occurs frequently in the unlabeled examples that are now believed to belong to the positive class. Thus the co-occurrence of the words "learn" and "teach" over the large set of unlabeled training data can provide useful information to construct a more accurate classifier that considers both "learn" and "teach" as indicators of positive examples.

The specific approach we describe here is based on a combination of two well-known learning algorithms: the naive Bayes classifier [Lewis and Ringuette, 1994] and the Expectation Maximization (EM) algorithm [Dempster *et al.*, 1977]. The naive Bayes algorithm is one of a class of statistical text classifiers that uses word frequencies as features. Other examples include TFIDF/Rocchio [Salton, 1991; Rocchio, 1971], regression models [Yang and Chute, 1993], k-nearest-neighbor [Yang and Pederson, 1997] and Support Vector Machines [Joachims, 1997b]. EM is a class of iterative algorithms for maximum likelihood estimation in problems with incomplete data. The result of combining these two is an algorithm that extends conventional text learning algorithms by using EM to dynamically derive pseudo-labels for unlabeled documents during learning, thereby providing a way to incorporate unlabeled data into supervised learning. Previous supervised algorithms for learning to classify from text do not incorporate unlabeled data.

A similar approach was used by Miller and Uyar [Miller and Uyar, 1997] for non-text data sources. We adapt this approach for the naive Bayes text classifier and conduct a thorough empirical analysis. We also show theoretically that unlabeled data carries information useful for improving parameter estimation under certain restrictive conditions, and survey results that show that this consequently improves classification.

In the following sections, we provide sufficient conditions under which the use of unlabeled data can be expected to improve classification accuracy. We present the probabilistic setting for naive Bayes, and its combination with the Expectation Maximization algorithm. We empirically demonstrate significantly improved performance on three text data sets. Finally, we discuss directions for future research. We argue that these results are of significant practical importance, since in many text learning domains (such as the Web), unlabeled data is available almost for free, whereas labeling the data can be truly expensive.

2. The Probabilistic Framework

To ground the theoretical aspects of our work, and to provide a setting for our algorithm, this section presents a probabilistic framework for characterizing the nature of documents and classifiers. We will then use this to introduce the classifier and show that unlabeled data can be used to improve classification. The framework follows commonly used assumptions [Lewis and Ringuette, 1994; Domingos and Pazzani, 1997] about the data—(1) that our text is produced by a mixture model, and (2) that there is a one-to-one correspondence between mixture components and classes.

In this setting, every document d_i is generated according to a probability distribution given by a mixture model, which is parameterized by θ. The mixture model consists of mixture components $c_j \in \mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$. Each component is parameterized by a disjoint subset of θ. Thus a document, d_i, is created by first selecting a component according to the priors, $\mathrm{P}(c_j|\theta)$, then, second, having the mixture component generate a document according to its own parameters, with distribution $\mathrm{P}(d_i|c_j;\theta)$. We can characterize the likelihood of a document with a sum of total probability over all mixture components:

$$\mathrm{P}(d_i|\theta) = \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta). \tag{1}$$

Each document has a class label. We assume that there is a one-to-one correspondence between classes and mixture model components, and thus use c_j to indicate both the jth mixture component and the jth class.[1] The class label of document d_i is written y_i, and if document d_i was generated by mixture component c_j we say $y_i = c_j$. This class label may or may not be known for a given document.

3. Proof of the Value of Unlabeled Data

In this section we show that, given this setting, documents with unknown class labels are useful for learning concept classes. 'Learning concept classes' in this setting is equivalent to estimating parameters of an unknown mixture model that produced the given training documents. First, we will provide a sufficient condition under which unlabeled documents can be used to estimate θ, and thus to improve the classification accuracy. We argue that this condition is fulfilled in the context of our high-dimensional mixture models. Without loss of generality, we will assume in this section the classification task is binary, that is $|\mathcal{C}| = 2$.

For unlabeled data to carry information about the parameters θ, it is sufficient that

1. the learning task is not *degenerate*, that is,

[1] This assumption will be relaxed in Section 6 by making this a one-to-many correspondence. Other work [Li and Yamanishi, 1997] relaxes this assumption in a many-to-one fashion.

$$\begin{aligned} \exists d_i, c_j, \theta, \theta'. \quad \mathrm{P}(D = d_i | C = c_j; \theta) &\neq \mathrm{P}(D = d_i | C = c_j; \theta') \\ \wedge\ \mathrm{P}(C = c_j | \theta) &\neq \mathrm{P}(C = c_j | \theta'). \end{aligned} \tag{2}$$

where D is a random variable over documents and C is a random variable over all classes, with events in $\mathcal{C}$.

2. $0 < |\mathcal{D}^l| < \infty$, where $|\mathcal{D}^l|$ is the number of labeled training documents.

The first assumption excludes tasks where learning is impossible, even given training data with class labels, for the reason that all parameterizations θ yield equivalent results. In this case knowledge of the parameters would not help aid the prediction of future data. The second assumption excludes cases where no labeled data is available, in which case unlabeled data cannot improve the classification accuracy. It also excludes cases where the labeled data is sufficient to estimate all parameters θ. It is easy to show that high-dimensional mixture models with unknown parameters meet the first condition, i.e., they are non-degenerate.

To show that knowledge about an unlabeled documents carries information about the parameters θ, we need to demonstrate the conditional dependence of θ on D. That is:

$$\mathrm{P}(\theta | D) \neq \mathrm{P}(\theta). \tag{3}$$

If this conjecture holds, a direct implication is that unlabeled data contain information about the parameters of θ. In this way, unlabeled data could help in supervised learning.

We provide a proof by contradiction. For this, we temporarily assume that θ and D are independent. By negating Equation 3, and applying Bayes' rule, we write:

$$\forall \theta; \quad \mathrm{P}(D | \theta) = \mathrm{P}(D). \tag{4}$$

One direct conclusion of this equation is that any two parameterizations, θ and θ', provide the same class probabilities for any sample. By substitution from Equation 1 this gives:

$$\sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(d_i | C = c_j; \theta) \mathrm{P}(C = c_j | \theta) = \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(d_i | C = c_j; \theta') \mathrm{P}(C = c_j | \theta'). \tag{5}$$

From here, it is a straightforward exercise to construct a document d_i and two parameterizations to generate a contradiction, making use of the non-degeneracy assumption above. Our assumption of non-degeneracy requires that for some document the individual terms in Equation 5 must differ for some θ and θ'; we can construct one for which the total probability of the document also differs for θ and θ'. Thus, our assumption of conditional independence is contradicted, and parameterizations must be conditionally dependent on the documents. This signifies that unlabeled data indeed contain information about parameters of the generative model. For this knowledge to aid classification, we have to exclude the two extreme cases: $|\mathcal{D}^l| = 0$ and $|\mathcal{D}^l| = \infty$. If there is no labeled data, unlabeled data cannot improve classification, as shown in [Castelli and Cover, 1995]. If there is infinite amounts of labeled data, all parameters θ can be recovered with probability 1 from the labeled data and the resulting classifier is Bayes-optimal [McLachlan and Basford, 1988]; thus, further unlabeled data cannot improve the classification accuracy.

Note that our argument does not immediately motivate an algorithm for extracting the information from the unlabeled data. Additionally, it not show that better parameter estimation will yield better classification. The following sections describe one way that is guaranteed to improve our parameter estimates. Section 7 contains a survey of related work that ties improvements in parameter estimates and increases in available training data to improvements in classification.

4. Naive Bayes for Text Classification

In this section, we present the naive Bayes classifier—a well-known, probabilistic algorithm for classifying text that is a special case of a mixture model. This algorithm forms the foundation upon which we will later incorporate unlabeled data. We continue the use of the two assumptions for data generation: that the documents are produced according to a mixture model, and that there is a one-to-one correspondence between classes and mixture components. Thus, the motivational result from the previous section still holds—that unlabeled documents can be beneficial. Naive Bayes makes the additional assumption that the probability of seeing a word in a document is independent of its context and its position [Lewis and Ringuette, 1994; Domingos and Pazzani, 1997].

The learning task is to use a set of training documents in order to form estimates for the parameters of the generative model. Naive Bayes forms Bayes optimal estimates of these parameters, then uses the estimated model to classify new documents.

The Generative Model

We now describe the full generative model for documents that will be used for learning text classifiers. It is a specialization of a mixture model, presented in Section 2. Document d_i is considered to be an ordered list of word events. We write $w_{d_{ik}}$ for the word in position k of document d_i, where the subscript of w indicates an index into the vocabulary $V = \langle w_1, w_2, \ldots, w_{|V|} \rangle$. When a document is generated and after a mixture component is selected, a document length is chosen independently of the component. (Note this assumes that document length is independent of class.[2]) Then, the selected mixture component generates a sequence words of the specified length. Thus, we can expand the second term from Equation 1, and express the probability of a document given a mixture component as the probability of the document's length and the product of the probabilities of the individual word events in the sequence. Note that, in this general setting, the probability of a word event must be conditioned on the words that precede it.

$$\mathrm{P}(d_i|c_j;\theta) = \mathrm{P}(\langle d_{i1}, d_{i2}, \ldots, d_{i|d_i|}\rangle|c_j;\theta) = \mathrm{P}(|d_i|)\prod_{k=1}^{|d_i|}\mathrm{P}(d_{ik}|c_j;\theta;d_{iq}, q<k). \tag{6}$$

Next we make the standard naive Bayes assumption: that the words of a document are generated independently of context, that is, independently of the other words in the same document given the class label. We further assume that the probability of a word is independent of its position within the document; thus, for example, the probability of seeing the word "dog" in the first position of a document is the same as seeing it in any other position. We can express these assumptions as:

$$\mathrm{P}(d_{ik}|c_j;\theta;d_{iq}, q<k) = \mathrm{P}(w_{d_{ik}}|c_j;\theta). \tag{7}$$

Combining these last two equations gives the complete naive Bayes expression for the probability of a document given its class:

$$\mathrm{P}(d_i|c_j;\theta) = \mathrm{P}(|d_i|)\prod_{k=1}^{|d_i|}\mathrm{P}(w_{d_{ik}}|c_j;\theta). \tag{8}$$

[2] Previous naive Bayes formalizations do not include this document length effect. In the most general case, document length should be modeled and parameterized.

The parameters of an individual mixture component are the collection of word probabilities, such that $\theta_{w_t|c_j} = \mathrm{P}(w_t|c_j;\theta)$, where $t = \{1, \ldots, |V|\}$ and $\sum_t \mathrm{P}(w_t|c_j;\theta) = 1$. Since we assume that for all classes, document length is uniformly distributed, it does not need to be parameterized. The only other parameters specified in the model are the class priors θ_{c_j}, which indicate the probabilities of selecting the different mixture components.

Training and Using a Classifier

Given these underlying assumptions of how the data is produced, the task of learning a text classifier consists of forming an estimate of θ by using a set of data and their associated class labels. The estimate of θ is written $\hat{\theta}$. With labeled training documents, $\mathcal{D} = \{d_1, \ldots, d_{|\mathcal{D}|}\}$, we can calculate Bayes optimal estimates for the parameters of the model that generated these documents. To calculate the probability of a word given a class, $\theta_{w_t|c_j}$, simply count the fraction of times that word occurs in the data for that class, and augment this fraction with Bayes optimal smoothing that primes the count for each word with a "pseudo-occurrence" of one [Vapnik, 1982]. This smoothing is sometimes referred to as the *Laplacean prior*, and is necessary to prevent probability zero probabilities for infrequently occurring words. These word probability estimates $\hat{\theta}_{w_t|c_j}$ are:

$$\hat{\theta}_{w_t|c_j} = \mathrm{P}(w_t|c_j;\hat{\theta}) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} N(w_t, d_i)\mathrm{P}(c_j|d_i)}{|V| + \sum_{s=1}^{|V|}\sum_{i=1}^{|\mathcal{D}|} N(w_s, d_i)\mathrm{P}(c_j|d_i)}, \tag{9}$$

where $N(w_t, d_i)$ is the count of the number of times word w_t occurs in document d_i and where $\mathrm{P}(c_j|d_i) = \{0, 1\}$ given by the class label. The class prior probabilities, $\hat{\theta}_{c_j}$, are estimated in the same fashion of counting, without smoothing, by

$$\hat{\theta}_{c_j} = \frac{\sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}{|\mathcal{D}|}. \tag{10}$$

Given estimates of these parameters calculated from the training documents, it is possible to turn the generative model up-side-down and calculate the probability that a particular mixture component generated a given document. We formulate this by an application of Bayes rule, and then substitutions using Equations 1 and 8.

$$\begin{aligned}
\mathrm{P}(c_j|d_i;\hat{\theta}) &= \frac{\mathrm{P}(c_j|\hat{\theta})\mathrm{P}(d_i|c_j;\hat{\theta})}{\mathrm{P}(d_i|\hat{\theta})} \\
&= \frac{\mathrm{P}(c_j|\hat{\theta})\prod_{k=1}^{|d_i|}\mathrm{P}(w_{d_{ik}}|c_j;\hat{\theta})}{\sum_{r=1}^{|\mathcal{C}|}\mathrm{P}(c_r|\hat{\theta})\prod_{k=1}^{|d_i|}\mathrm{P}(w_{d_{ik}}|c_r;\hat{\theta})}
\end{aligned} \tag{11}$$

Note that since document lengths are class independent, the document length terms from Equation 1 cancel and do not appear.

If the task is to classify a test document d_i into a single class, simply select the class with the highest posterior probability, $\arg\max_j \mathrm{P}(c_j|d_i;\hat{\theta})$.

Note that our four assumptions about the generation of text documents (mixture model, one-to-one correspondence between mixture components and classes, word independence, and document length distribution) are all violated in practice. Documents often fall into overlapping categories. Words within a document are not independent of each other—grammar and topicality ensure this. Despite these violations, empirically, the Naive Bayes classifier does a good job of classifying text documents [Lewis and Ringuette, 1994;

Craven *et al.*, 1998; Yang and Pederson, 1997; Joachims, 1997a]. This paradox is explained by the fact that classification estimation is only a function of the sign (in binary cases) of the function estimation; the function approximation can still be poor while classification accuracy remains high [Domingos and Pazzani, 1997; Friedman, 1997].

The above formulation of naive Bayes assumes a generative model that accounts for the number of times a word appears in a document. This is equivalent to a multinomial event model (without the factorial terms that account for event ordering) [McCallum and Nigam, 1998a]. This formulation has been used by numerous practitioners of naive Bayes text classification [Lewis and Gale, 1994; Kalt and Croft, 1996; Joachims, 1997a; Li and Yamanishi, 1997; Mitchell, 1997; McCallum *et al.*, 1998]. However, there is another formulation of naive Bayes text classification that instead assumes a generative model and document representation where each word in the vocabulary is a binary feature, and is modeled by a Bernoulli trial [Robertson and Sparck-Jones, 1976; Lewis, 1992; Kalt and Croft, 1996; Larkey and Croft, 1996; Koller and Sahami, 1997]. Empirical comparisons show that the multinomial formulation yields higher-accuracy classifiers [McCallum and Nigam, 1998a].

5. Using EM to Incorporate Unlabeled Data

When naive Bayes is given just a small set of labeled training data, classification accuracy will suffer because variance in the parameter estimates of the generative model will be high. However, by augmenting this small set with a large set of unlabeled data and combining the two sets with EM, we can improve our parameter estimates.

EM concurrently generates probabilistically-assigned labels for the unlabeled documents, and a more probable model with smaller parameter variance that predicts these same probabilistic labels.

This section describes how to use EM within the probabilistic framework of the previous section. This is a special case of the more general missing values formulation, as presented by [Ghahramani and Jordan, 1994]. While the theory of why EM works is not particularly simple, the resulting algorithm is very straightforward. Our algorithm is outlined in Table 1.

We are given a set of training documents $\mathcal{D}$ and the task is to build a classifier of the form in the previous section. However, unlike previously, in this section we assume that only some subset of the documents $d_i \in \mathcal{D}^l$ come with class labels $y_i \in \mathcal{C}$, and for the rest of the documents, in subset $\mathcal{D}^u$, the class labels are unknown. Thus we have a disjoint partitioning of $\mathcal{D}$, such that $\mathcal{D} = \mathcal{D}^l \cup \mathcal{D}^u$.

Consider the probability of all the training data, $\mathcal{D}$. The probability of all the data is simply the product over all the documents, because each document is independent of the others, given the model. From Equation 1, the probability of all the data is:

$$\begin{aligned} \mathrm{P}(\mathcal{D}|\theta) &= \prod_{i=1}^{|\mathcal{D}^u|} \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta) \\ &\quad \times \prod_{i=1}^{|\mathcal{D}^l|} \mathrm{P}(c_j = y_i|\theta)\mathrm{P}(d_i|c_j = y_i;\theta). \end{aligned} \tag{12}$$

For the unlabeled documents, we use a direct application of Equation 1. For the labeled documents, we are given the generative component by the label y_i and thus do not need to sum over all class components.

Again, learning a classifier corresponds to calculating a maximum likelihood estimate of θ—finding the parameterization that is most likely given our training data: $\arg\max \mathrm{P}(\theta|\mathcal{D})$. By Bayes' rule, $\mathrm{P}(\theta|\mathcal{D}) =$

- Build an initial classifier by estimating θ from the labeled documents only (Equations 9 and 10).
- Loop while classifier parameters change:
  - Use the current classifier to probabilistically label the unlabeled documents (Equation 11).
  - Recalculate the classifier parameters θ given the probabilistically assigned labels (Equations 9 and 10).

Table 1: The Algorithm.

$\mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)/\mathrm{P}(\mathcal{D})$. $\mathrm{P}(\mathcal{D})$ is a constant; maximum likelihood estimation assumes that $\mathrm{P}(\theta)$ is a constant, so taking the log, we define $\eta = \log(\mathrm{P}(\theta)/\mathrm{P}(\mathcal{D}))$. Maximizing the log likelihood is the same as maximizing the likelihood. Using Equation 13 and Bayes rule, we write the log likelihood, $l(\theta|\mathcal{D}) \equiv log(\mathrm{P}(\theta|\mathcal{D}))$, as:

$$
\begin{aligned}
l(\theta|\mathcal{D}) \quad = \quad & \eta + \sum_{i=1}^{|\mathcal{D}^u|} \log \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta) \\
& + \sum_{i=1}^{|\mathcal{D}^l|} \log\left(\mathrm{P}(c_j = y_i|\theta)\mathrm{P}(d_i|c_j = y_i;\theta)\right).
\end{aligned} \tag{13}
$$

Because the first line of this equation has a log of sums, it is not computable in closed-form. However, if we knew all the class labels, as in $\mathcal{D}^l$, then we could avoid this. If we had access to the class labels represented as the matrix of binary indicator variables $\mathbf{z}$, $\mathbf{z}_i = \langle z_{i1}, \ldots, z_{i|\mathcal{C}|}\rangle$, where $z_{ij} = 1$ iff $y_i = c_j$ else $z_{ij} = 0$, then we could express this complete log likelihood of the parameters, $l_c(\theta|\mathcal{D}, \mathbf{z})$, without a log of sums:

$$
l_c(\theta|\mathcal{D}, \mathbf{z}) = \eta + \sum_{i=1}^{|\mathcal{D}|}\sum_{j=1}^{|\mathcal{C}|} z_{ij} \log\left(\mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta_j)\right). \tag{14}
$$

This formulation of the log likelihood is readily computable in closed-form. Dempster [Dempster *et al.*, 1977] uses this insight in the Expectation Maximization algorithm, which finds a *local* maximum likelihood $\hat{\theta}$ by an iterative procedure that recomputes the expected value of $\mathbf{z}$ and the maximum likelihood parameterization given $\mathbf{z}$. Note that for the labeled documents $\mathbf{z}_i$ is already known. It must be estimated for the unlabeled documents. If we denote the expected value of $\mathbf{z}$ at iteration k, by $Q^{(k)}$, we can find a local maximum for $l(\theta|\mathcal{D})$ by iterating the following two steps:

- E-step: Set $Q^{(k)} = E[\mathbf{z}|\mathcal{D};\hat{\theta}^{(k)}]$.
- M-step: Set $\hat{\theta}^{(k+1)} = \arg\max_\theta \mathrm{P}(\theta|\mathcal{D};Q^{(k)})$.

In practice, the E-step corresponds to calculating probabilistic labels $\mathrm{P}(c_j|d_i;\theta)$ for every document by using the current estimate of θ and Equation 11. The M-step corresponds to calculating a new maximum likelihood estimate for θ given the current estimates for $\mathrm{P}(c_j|d_i;\theta)$ using Equations 9 and 10. See Table 1 for an outline of our algorithm.

EM finds the $\hat{\theta}$ that locally maximizes the probability of all the data, both the labeled and the unlabeled.

6. Experimental Results

In this section, we give empirical evidence that using the algorithm in Table 1 with labeled and unlabeled documents outperforms naive Bayes, which does not on its own use unlabeled documents. We present experimental results with three different text corpora from the domains of UseNet news articles (20 Newsgroups), web pages (WebKB), and newswire articles (Reuters).[3]

Datasets and Protocol

The 20 Newsgroups data set [Joachims, 1997a], collected by Ken Lang, consists of 20,017 articles divided almost evenly among 20 different UseNet discussion groups. When words from a stoplist of common short words are removed, there are 62,258 unique words that occur more than once. Many of the categories fall into confusable clusters; for example, five of them are `comp.*` discussion groups, and three of them discuss religion. When tokenizing this data, we skip the UseNet headers (thereby discarding the subject line); tokens are formed from contiguous alphabetic characters, which are left unstemmed. Best performance was obtained with no feature selection, and by normalizing word counts by document length. Accuracy results are reported as averages of ten test/train splits, with 20% of the documents randomly selected for placement in the test set.

The WebKB data set [Craven *et al.*, 1998] contains 8145 web pages gathered from university computer science departments. For four departments, all web pages were included; additionally, there are many pages from an assortment of other universities. The pages are divided into seven categories: student, faculty, staff, course, project, department and other. In this paper, we use the four most populous non-other categories: student, faculty, course and project, all together containing 4199 pages. We did not use stemming or a stoplist; we found that using a stoplist actually hurt performance because, for example, "my" is the fourth-ranked word by information gain, and is an excellent indicator of a student homepage. As done previously [Craven *et al.*, 1998], we use only the 2000 most informative words, as measured by average mutual information with the class variable. This feature selection method is commonly used for text [Yang and Pederson, 1997; Koller and Sahami, 1997; Joachims, 1997a]. Accuracy results presented below are an average of twenty test/train splits, again randomly holding out 20% of the documents for testing.

The 'ModApte' train/test split of the Reuters 21578 Distribution 1.0 data set consists of 12902 articles and 90 topic categories from the Reuters newswire. Following several other studies [Joachims, 1997b; Liere and Tadepalli, 1997] we use the 10 most populous classes and build binary classifiers for each class. We use all the words inside the <TEXT> tags, including the title and the dateline, except that we remove the REUTER and &# tags that occur at the top and bottom of every document. We use a stoplist, but do not stem. Vocabulary selection, when used, is again performed with average mutual information with the class variable. In the standard ModApte split, there are 3299 documents in the test set and 9603 in the training set. Results are reported as average results of ten randomly selected training sets. The complete ModApte test set is used to calculate precision-recall breakeven points, a standard information retrieval measure for binary classification.

In experiments with EM, the initial θ was estimated using only the labeled data, and EM iterations progressed from there, as in Table 1. All experiments were performed with eight EM iterations; significant changes occur in the first few iterations. We never found classification accuracy to improve beyond the eighth iteration.

Results

Figure 1 shows the effect of using EM with unlabeled data in the 20 Newsgroups data set. The vertical axis

[3] All three of these data sets are available on the Internet. See http://www.cs.cmu.edu/~textlearning and http://www.research.att.com/~lewis.

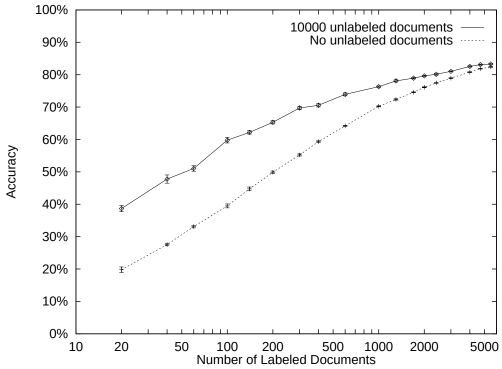

Figure 1: Classification accuracy on the 20 Newsgroups data set, both with and without 10000 unlabeled documents. The narrow error bars on each data point are twice the standard error. With small amounts of training data, using EM yields more accurate classifiers. In the limit, the two methods converge.

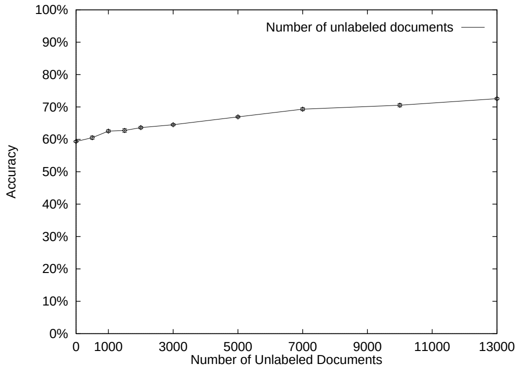

Figure 2: The effect of varying the number of unlabeled data. Classification accuracy is shown on the 20 Newsgroups data set with 400 labeled documents, and varying amounts of unlabeled data. More unlabeled data helps.

indicates classifier accuracy on test sets, and the horizontal axis indicates the amount of labeled data used in training. We vary the amount of labeled training data, and compare the classification accuracy of traditional naive Bayes (no unlabeled data) with an EM learner that has access to 10000 unlabeled documents. EM performs significantly better. For example, with 400 labeled documents (20 documents per class), naive Bayes reaches 59% accuracy, while EM achieves 70%. Note here that EM performs well even with a very small number of labeled documents; with only 20 documents (a single labeled document per class), naive Bayes gets 19%, EM 39%. As expected, when there is a lot of labeled data, and the naive Bayes learning curve has reached a plateau, the learner is already saturated, and having unlabeled data does not help. In Figure 2 we hold the number of labeled documents constant at 400, and vary the number of unlabeled documents in the horizontal axis. Naturally, more unlabeled data helps.

These results demonstrate that EM finds a model with more probable parameter estimates, and that these improved estimates reduce classification accuracy and the need for labeled training examples. For example, to get 75% classification accuracy, EM requires 1000 labeled examples, while naive Bayes requires 2000 labeled examples to achieve the same accuracy.

To gain some intuition about why EM works, we present a detailed trace of one example. Table 2 provides a window into the evolution of the classifier over the course of EM iterations for this example. Based on the WebKB data set, each column shows the ordered list of words that the model believes are most "predictive"

Iteration 0	Iteration 1	Iteration 2
intelligence	*DD*	*D*
DD	*D*	*DD*
artificial	lecture	lecture
understanding	cc	cc
*DD*w	*D*⋆	*DD*:*DD*
dist	*DD*:*DD*	due
identical	handout	*D*⋆
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	*DD*am	hw
cognitive	yurttas	exam
logic	homework	problem
proving	kfoury	*DD*am
prolog	sec	postscript
knowledge	postscript	solution
human	exam	quiz
representation	solution	chapter
field	assaf	ascii

Table 2: Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific example. By the second iteration of EM, many common course-related word have high weights. The symbol D indicates an arbitrary digit.

of the course class. Word are judged to be "predictive" using a weighted log likelihood ratio.[4] At Iteration 0, the parameters were estimated from a randomly-chosen single labeled document per class. Notice that the course document seems to be about a specific Artificial Intelligence course at Dartmouth. After two EM iterations with 2500 unlabeled documents, we see that EM has used the unlabeled data to find words that are more generally indicative of courses. The classifier corresponding to the first column gets 50% accuracy; by the eighth iteration, the classifier achieves 71% accuracy.

The graph in Figure 3 shows the benefits of 2500 unlabeled documents on the WebKB data set. Again, EM improves accuracy significantly, especially when the amount of labeled data is small. When there are 12 labeled documents (three per class), traditional naive Bayes attains 50% accuracy, while EM reaches 64%. When there is a lot of labeled data however, EM hurts performance slightly. With 280 labeled documents, naive Bayes obtains 82% accuracy, and EM gets 78%.

Varying the Weight of the Unlabeled Data

We hypothesize that the reason EM hurts performance here is that the data does not fit the assumptions of our model as well as 20 Newsgroups—that is, the mixture components that best explain the unlabeled data do not correspond as well to the class labels. In other words, EM places strong assumptions about the generative process for the documents and optimizes the parameters subject to those assumptions and

[4] The weighted log likelihood ratio used to rank the words in Figure 2 is:

$$\mathrm{P}(w_t|c_j)\log\left(\frac{\mathrm{P}(w_t|c_j)}{\mathrm{P}(w_t|\neg c_j)}\right), \tag{15}$$

which can be understood in information-theoretic terms as word w_t's contribution to the average inefficiency of encoding words from class c_j using a code that is optimal for the distribution of words in $\neg c_j$; the sum of this quantity over all words is the Kullback-Leibler divergence between the distribution of words in c_j and the distribution of words in $\neg c_j$ [Cover and Thomas, 1991].

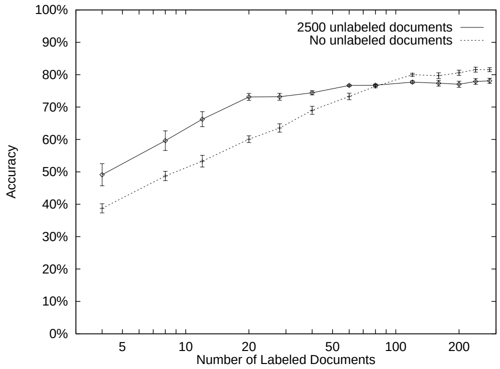

Figure 3: Classification accuracy on the WebKB data set, both with and without 2500 unlabeled documents, averaged over 20 trials per data point. With small amounts of labeled documents, EM helps, but in the limit, it degrades performance slightly, indicating a misfit between the data and the assumed generative model.

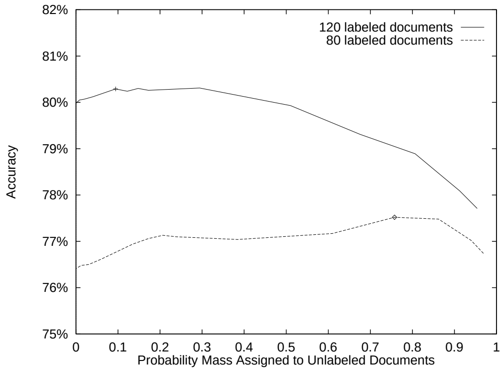

Figure 4: The effects of varying the relative importance of the labeled and unlabeled documents on the WebKB data for two different amounts of labeled data. By using cross-validation, automatic selection picks the near-optimal values of α indicated with the cross and the diamond. Note the magnified vertical scale.

all the data. If the assumptions do not hold for the data, the optimization may no longer be beneficial for classification. When EM has very little labeled training data, parameter estimation is so desperate for guidance that EM still helps in spite of the somewhat violated assumptions; however, when there is enough labeled training data that the labeled data alone is already sufficient for good parameter estimation, the estimates can be modestly thrown off by EM's inclusion of the unlabeled data. It is not surprising that the unlabeled data can throw off parameter estimation when one considers that the number of unlabeled documents is always much greater than the number of labeled documents (*e.g.* 2500 versus 280), and thus, even at the points in Figure 3 with the largest amounts of labeled data, the great majority of the probability mass used in the M-step to estimate the classifier parameters actually comes from the unlabeled data.

This insight suggests a simple fix. We can add a learning parameter that varies the relative contributions of the labeled and unlabeled data to parameter estimation in the M-step. In our implementation this parameter is embodied by a factor, α, that reduces the weight of unlabeled documents in the estimation of $\theta_{w_t|c_j}$ in Equation 9. In essence we can make each unlabeled document count as only a fraction, α, of a document, thus correctly balancing the "mass" of the labeled and unlabeled documents to optimize performance. We can build models for varying values of α and choose the best one using leave-one-out cross-validation on the training data to tune this parameter after EM has iterated to convergence. Empirically, cross-validation picks the optimal value most of the time, and a near-optimal value otherwise.

Figure 4 plots classification accuracy while varying α in the horizontal axis, and does so for two different amounts of labeled training data. The bottom curve is obtained using 80 labeled documents—a vertical

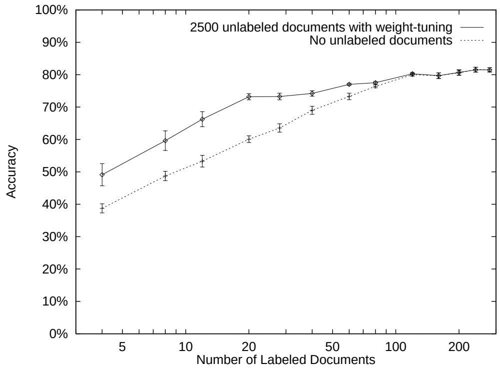

Figure 5: Classification accuracy on the WebKB data set, with α optimization selected by cross-validation compared to using no unlabeled data. Note that accuracy using unlabeled data now does not degrade with large amounts of labeled data, as in Figure 3, and still maintains the large benefits for small training sets.

slice in Figure 3 at the point where the naive Bayes and EM curves cross. The top curve is obtained using 120 labeled documents—a slice after the crossover. First, note the magnified vertical scale to facilitate interpretation of this data. Second, (remembering that the right-most point corresponds to EM with the weighting used to generate Figure 3, and the left-most to regular naive Bayes), note that the best-performing values of α are somewhere between the extremes. A paired t-test indicates that both of these maxima are statistically significantly higher than either end point ($p < 0.01$). Third, note that we can select optimal or near-optimal values automatically using cross-validation. These plots show performance on the held-out test set; the values of α are selected by leave-one-out cross-validation are indicated by the diamond and cross on the bottom and top curves respectively. The trend across all amounts of labeled data is that with more labeled data, the unlabeled data gets less weight.

Figure 5 compares the performance of naive Bayes against EM with α tuned by cross-validation. Unlike Figure 3 with a fixed α, here EM strictly dominates naive Bayes. This indicates that we can automatically avoid degradation in accuracy when using EM and still preserve benefits seen with a small training set.

Multiple Mixture Components per Class

Faced with data that does not fit the assumptions of our model, the α-tuning approach described above addresses this problem by allowing the model to incrementally ignore the unlabeled data. Another, more direct approach is to change the model so that it more naturally fits the data. Above, we hypothesized that the data violates our assumption that there is a one-to-one correspondence between mixture components and classes, and that the mixture model components found by EM do not correspond well to the class labels. Flexibility can be added to the mapping between mixture components and class labels by allowing multiple mixture components per class, and we expect this to improve performance when data for each class is actually multi-modal.

With an eye towards testing this hypothesis, we applied EM to the Reuters corpus. Since the documents in this data set can have multiple class labels, each category is traditionally evaluated with a binary classifier. Thus, the negative class covers 89 distinct categories, and we expected this task to strongly violate the assumption that all the data for the negative class is generated by a single mixture component. For these experiments, we randomly selected ten positively labeled documents, 40 negatively labeled documents, and 7000 unlabeled documents. This uneven labeling is justified because in all of the binary Reuters classification tasks the negative class is much more frequent than the positive class.

The left column of Table 3 shows average precision-recall breakeven points for 10 trials of each experiment, for naive Bayes. These numbers are presented at the best vocabulary size for each task, indicated in parentheses. Classifiers for different categories performed best with widely varying vocabulary sizes. This variance of optimal vocabulary size is unsurprising. As previously noted [Joachims, 1997a], categories like

Category	NB 1	EM 1	EM 5	EM 20	EM 40	Diff
acq	75.9 (19371)	39.5 (19371)	87.2 (19371)	88.4 (5000)	**88.9** (5000)	+13.0
corn	40.5 (100)	21.1 (100)	36.6 (100)	**39.8** (100)	39.1 (200)	-0.7
crude	60.7 (19371)	27.8 (100)	55.3 (100)	63.9 (100)	**66.6** (500)	+5.9
earn	92.6 (19371)	90.2 (2000)	95.0 (19371)	**95.3** (19371)	95.2 (19371)	+2.7
grain	51.7 (2000)	21.0 (100)	54.0 (100)	54.6 (100)	**55.8** (1000)	+4.1
interest	52.0 (2000)	25.9 (100)	42.0 (100)	48.6 (100)	**50.3** (500)	-1.7
money-fx	57.7 (3000)	28.8 (100)	46.2 (100)	54.7 (500)	**59.7** (500)	+2.0
ship	58.1 (19371)	9.3 (100)	40.1 (300)	46.5 (500)	**55.0** (500)	-3.1
trade	56.8 (19371)	34.7 (100)	49.0 (100)	54.3 (100)	**57.0** (1000)	+0.2
wheat	48.9 (2000)	13.0 (100)	38.5 (100)	42.1 (100)	**44.2** (1000)	-4.7

Table 3: Precision-Recall breakeven points showing performance of binary classifiers on Reuters with traditional naive Bayes, EM with one mixture component per class, and EM with varying multi-component models for the negative class. The best multi-component model is noted in bold, and the difference in performance between it and naive Bayes is noted in the rightmost column. Results are shown on the optimal vocabulary size, indicated in parentheses. Note that performance is poor with a single component per class for EM because the data-model fit is poor. When a more natural multi-component model is used for the negative class, EM improves upon naive Bayes.

"wheat" and "corn" are known for a strong correspondence between words and categories, while categories like "acq" are known for a more subtle class definition. The categories with narrow definitions require small vocabularies for best classification, while those with a broader definition require a large vocabulary to capture the category.

The second column of Table 3 shows the results of performing EM on the data with a single negative centroid, as in previous experiments. As expected, the fit between the assumed model and the Reuters data is poor, and the results using EM are dramatically worse than simple naive Bayes. Because the negative class is truly multi-modal, fitting a single naive Bayes class with EM to the data does not accurately capture its distribution. However, by choosing an appropriate multi-component model with which to run EM, we can get results that do improve upon naive Bayes.

The remainder of Table 3 shows the effects of using different multi-component models in conjunction with EM. The negative class is modeled with five, 20 or 40 negative centroids. When initializing these centroids for running EM, they are initialized with randomly assigned negative documents. The best performer is noted in bold, and the difference between it and naive Bayes is noted in the difference column. A paired t-test on each trial over all categories shows that the improvement in average breakeven point from 59.5% to 61.3% is statistically significant ($p < 0.0001$). Note that in most cases, EM does best with 40 components, confirming our hypothesis that a more complex multi-component model more accurately represents the Reuters data.

These results indicate that correct model selection is crucial for EM with data sets that are not naturally modeled with a small number of generative components. When the data is accurately modeled, gains from using EM are readily seen. One obvious question is how to select the best model representation. Cheeseman and Stutz [Cheeseman and Stutz, 1996] investigate this for clustering tasks with no labeled data, and explicitly compare the probability of the data for different models, and select the best match, with a prior that prefers smaller models. For classification tasks, it may be more beneficial to select this with a more appropriate classification-oriented criteria. Consider that if the number of components equals the number of examples, the data can be modeled perfectly. However, this will have poor generalization. One possibility then is to use leave-one-out cross-validation in the same manner as with tuning α.

7. Survey of Related Theoretical Work

The previous section provides empirical evidence that the use of unlabeled data in conjunction with labeled data can help improve classification accuracy. This is built upon theoretical work showing that use of unlabeled data can improve parameter estimates. Here, we provide a link between the two, and survey the literature on convergence and error bounds describing the degree to which labeled and unlabeled data improve classification.

This section assumes a high-dimensional mixture model, of which naive Bayes is one special case. Recall that θ is used to denote the parameters of the mixture model. We will use m to denote the number of different words (each of which induces a parameter in each mixture component). In certain asymptotic cases, the relative value of unlabeled data in learning classification is well-understood.

- **No unlabeled data.** First consider the efficiency of estimating θ from a pool of labeled data only, $\mathcal{D}^l$. According to [Devroye *et al.*, 1996], estimating θ using the maximum likelihood estimator is subject to the following error bound:

$$\mathrm{P}(|\theta - \hat{\theta}| > \varepsilon) \quad \leq \quad 2^m e^{-|\mathcal{D}^l|\varepsilon^2/2} \tag{16}$$

  Here $\hat{\theta}_{D|C}$ and $\hat{\theta}_C$ are the maximum likelihood estimates for θ. From this it follows that the parameter estimation error $|\theta - \hat{\theta}|$ converges to zero at the rate $O(1/\sqrt{|\mathcal{D}^l|})$.

- **Infinite unlabeled data.** If *infinite* amounts of unlabeled data are available, however, the parameters of the mixture components θ can be recovered from the unlabeled data [McLachlan and Basford, 1988], but not the assignment of mixture components to classes. Thus, the estimation problem reduces to the problem of learning a permutation matrix, which assigns labels to the different mixture components. Without any labeled data, this permutation cannot be found, and thus, although the parameters are known, classification error is not reduced from random guessing. As shown in [Castelli and Cover, 1995], with infinite unlabeled data, the classification error approaches the Bayes optimal solution at an exponential rate in the number of labeled examples given. Thus, if infinite amounts of unlabeled data are available, the convergence rate of learning from labeled data is changed by an exponential factor.

- **Trade-off.** As shown in [Castelli and Cover, 1996], labeled data can be exponentially more valuable than unlabeled data in reducing the probability of classification error by non-degenerate Bayesian classifiers. Their analysis investigates a restricted estimation problem, in which the individual mixture components are known, but two things aren't: the a priori likelihood of each mixture component, and the assignment of mixture components to class labels. In such a situation, the classification error is essentially dominated by the number of unlabeled documents; unless the number of unlabeled data grows faster than an exponential function in the number of labeled documents, in which case the classification error is essentially determined by the number of labeled samples. This result, however, assumes that the parameters of the individual mixture components are known; little is known for the more general case, where unlabeled data can be used to estimate those.

Shahshahani and Landgrebe [Shahshahani and Landgrebe, 1994] investigates the utility of unlabeled data in supervised learning, with quite different results. They analyze the convergence rate under the assumption that unbiased estimators are available for θ, for both the labeled and the unlabeled data. Their bounds, which are based on Fisher information gain, show a linear (instead of exponential) value of labeled vs. unlabeled data. Unfortunately, their analysis assumes that unlabeled data alone is sufficient to estimate both parameter vectors; thus, they assume that the target concept can be recovered without any target labels. This assumption is unrealistic. As shown in [Castelli and Cover, 1995], unlabeled data does not improve the classification results in the absence of labeled data. Shahshahani and Landgrebe's analysis also does not investigate the classification error.

Unfortunately, all these results rest on two restrictive assumptions, both of which are usually violated in text classification domains. First, they are asymptotic, *i.e.*, they characterize the importance of labeled and unlabeled documents in the limit. Little is known for the non-asymptotic case. Second, they assume that the underlying mixture model is correct, that is, there exists a parameter θ so that $P(D|\theta)$ is identical to the distribution that generated the data has been generated. Unfortunately, if this assumption is violated, estimators such as the maximum likelihood estimator may generate poor results. As shown in [Devroye *et al.*, 1996], under such conditions the maximum likelihood estimator can easily fail to minimize the classification error on the training set.

8. Related Work

Two other studies have used EM to combine labeled and unlabeled data for classification [Miller and Uyar, 1997; Shahshahani and Landgrebe, 1994]. Instead of naive Bayes, Shahshahani and Landgrebe use a mixture of Gaussians; Miller and Uyar use Mixtures of Experts. They demonstrate experimental results on non-text data sets with up to 40 features. In contrast, our textual data sets have three orders of magnitude more features.

Our work is an example of applying EM to fill in missing values—the missing values are the class labels of the unlabeled training examples. Ghahramani and Jordan have used EM with mixture models to fill in missing values [Ghahramani and Jordan, 1994]. The emphasis of their work was on missing feature values, where we focus on augmenting a very small but complete set of labeled data.

The AutoClass project [Cheeseman and Stutz, 1996; Hanson *et al.*, 1991] investigated the combination of the EM algorithm with an underlying model of a naive Bayes classifier. The emphasis of their research was the discovery of novel clusterings for unsupervised learning over unlabeled data. AutoClass has not been applied to text or classification.

Many approaches to reducing the need for labeled training examples have used active learning, in which an algorithm iteratively selects an unlabeled example, asks a human labeler for its classification, and rebuilds its classifier. Approaches differ in their methods for selecting the unlabeled example to request a label. Three such examples are relevance sampling and uncertainty sampling [Lewis and Gale, 1994; Lewis, 1995], and a "Query By Committee" approach [Liere and Tadepalli, 1997].

Several other statistical text classifiers have been used by others in a variety of domains [Yang and Pederson, 1997; Joachims, 1997b; Cohen and Singer, 1997] However, naive Bayes has a strong probabilistic foundation for EM, and is more efficient for large data sets. The thrust of this paper is to straightforwardly demonstrate the value of unlabeled data; a similar approach could apply unlabeled data to more complex classifiers.

9. Summary and Conclusions

This paper has explored the question of when and how unlabeled data may be used to supplement scarce labeled data in machine learning problems, especially when learning to classify text documents. This is an important question in text learning, because of the high cost of hand-labeling data and because of the availability of huge volumes of unlabeled data. In this paper we have presented a theoretical model, an algorithm, and experimental results that show significant improvements from using unlabeled documents for training classifiers in three real-world text classification tasks.

Our theoretical model characterizes a setting in which unlabeled data can be used to boost the accuracy of learned classifiers: when the probability distribution that generates documents can be described as a mixture distribution, and where the mixture components correspond to the class labels. These conditions fit exactly the model used by the naive Bayes classifier.

However, the complexity of natural language text will not soon be completely captured by statistical models. It is interesting then, to consider the sensitivity of a classifier's model to data that is inconsistent with that model. When the data is inconsistent with the assumptions of the model, our method for adjusting the weight of the contribution of unlabeled data, (as presented in our results on WebKB), prevents the unlabeled data from hurting classification accuracy. With our results on Reuters, we study ways to improve the model so that it better matches the assumptions about mixture models and the correspondence between components and classes. The results show improved classification accuracy, and suggest exploring the use of even more complex mixture models that better correspond to textual data distributions.

We believe that our algorithm and others using unlabeled data require a closer match between the data and the model than those with only labeled data; if the intended target concept and model differ too much with the actual distribution of the data, then the use of unlabeled data will hurt instead of help. We intend to make a closer theoretical and empirical study on the tradeoffs between the use of unlabeled data and the inherent model inadequacies for several text learning algorithms.

We also see several other interesting directions for future work with unlabeled data. Two other learning task formulations could also benefit from using EM: (1) an active learning approach that uses an explicit model of unlabeled data could incorporate EM iterations at every stage to improve its classification, and to better select for which data to request class labels from a labeler [McCallum and Nigam, 1998b]; (2) an incremental learning algorithm that re-trains throughout the testing phase could use the unlabeled test data received early in the testing phase in order to improve performance on the later test data.

Other problem domains share some similarities with text domains, and also have abundant unlabeled data with limited, expensive labeled data. Robotics, vision, and information extraction are three such domains. Applying the techniques in this paper could improve classification in these areas as well.

Acknowledgements

We thank Larry Wasserman for his extensive help on some of the theoretical aspects of this work. We thank Doug Baker for help formatting the Reuters data set. This research was supported in part by the Darpa HPKB program under contract F30602-97-1-0215.

References

[Castelli and Cover, 1995] V. Castelli and T. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 16:105–111, January 1995.

[Castelli and Cover, 1996] V. Castelli and T. Cover. 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, November 1996.

[Cheeseman and Stutz, 1996] Peter Cheeseman and John Stutz. Bayesian classification (AutoClass): Theory and results. In U. Fayyad, editor, *Advances in Knowledge Discovery and Data Mining*, 1996.

[Cohen and Singer, 1997] W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. In *Proceedings of ACM SIGIR Conference*, 1997.

[Cover and Thomas, 1991] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley, 1991.

[Craven *et al.*, 1998] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of AAAI-98*, 1998.

[Dempster *et al.*, 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM. algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

[Devroye *et al.*, 1996] L. Devroye, L. Gyöfri, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer Verlag, Berlin, 1996.

[Domingos and Pazzani, 1997] P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classifier. *Machine Learning*, 29:103–130, 1997.

[Friedman, 1997] Jerome H. Friedman. On bias, variance, 0/1 - loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1:55–77, 1997.

[Ghahramani and Jordan, 1994] Zoubin Ghahramani and Michael Jordan. Supervised learning from incomplete data via an EM approach. In *Advances in Neural Information Processing Systems (NIPS 6)*. Morgan Kauffman Publishers, 1994.

[Hanson *et al.*, 1991] Hanson, Cheeseman, and Stutz. Bayesian classification theory. Technical Report Technical Report FIA-90-12-7-01, NASA AMES Research Center, 1991.

[Joachims, 1997a] Thorsten Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *International Conference on Machine Learning (ICML)*, 1997.

[Joachims, 1997b] Thorsten Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. Technical Report LS8-Report, University of Dortmund, November 1997.

[Kalt and Croft, 1996] T. Kalt and W. B. Croft. A new probabilistic model of text classification and retrieval. Technical Report IR-78, University of Massachusetts Center for Intelligent Information Retrieval, 1996. http://ciir.cs.umass.edu/publications/index.shtml.

[Koller and Sahami, 1997] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.

[Lang, 1995] Ken Lang. Newsweeder: Learning to filter netnews. In *International Conference on Machine Learning (ICML)*, pages 331–339, 1995.

[Larkey and Croft, 1996] Leah S. Larkey and W. Bruce Croft. Combining classifiers in text categorization. In *SIGIR-96*, 1996.

[Lewis and Gale, 1994] D. Lewis and Gale. A sequential algorithm for training text classifiers. In *Proceedings of ACM SIGIR Conference*, 1994.

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

[Lewis and Ringuette, 1994] David Lewis and Marc Ringuette. A comparison of two learning algorithms for text categorization. In *Third Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, 1994.

[Lewis, 1992] David D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *SIGIR-92*, 1992.

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

[Li and Yamanishi, 1997] Hang Li and Kenji Yamanishi. Document classification using a finite mixture model. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 1997.

[Liere and Tadepalli, 1997] Liere and Tadepalli. Active learning with committees for text categorization. In *AAAI-97*, 1997.

[McCallum and Nigam, 1998a] Andrew McCallum and Kamal Nigam. A comparison of event models for naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998. http://www.cs.cmu.edu/~mccallum.

[McCallum and Nigam, 1998b] Andrew McCallum and Kamal Nigam. Employing EM and pool-based active learning for text classification. In *ICML-98*, 1998.

[McCallum *et al.*, 1998] Andrew McCallum, Ronald Rosenfeld, Tom Mitchell, and Andrew Ng. Improving text clasification by shrinkage in a hierarchy of classes. In *ICML-98*, 1998.

[McLachlan and Basford, 1988] G.J. McLachlan and K.E. Basford. *Mixture Models.* Marcel Dekker, New York, 1988.

[Miller and Uyar, 1997] David J. Miller and Hasan S. Uyar. A mixture of experts classifier with learning based on both labelled and unlabelled data. In *Advances in Neural Information Processing Systems (NIPS 9)*, 1997.

[Mitchell, 1997] Tom M. Mitchell. *Machine Learning.* WCB/McGraw-Hill, 1997.

[Pazzani *et al.*, 1996] M. J. Pazzani, J. Muramatsu, and D. Billsus. Syskill & Webert: Identifying interesting Web sites. In *AAAI-96*, 1996.

[Robertson and Sparck-Jones, 1976] S. E. Robertson and K. Sparck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[Rocchio, 1971] J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System:Experiments in Automatic Document Processing*, chapter 14, pages 313–323. Prentice Hall, 1971.

[Salton, 1991] G. Salton. Developments in automatic text retrieval. *Science*, 253:974–979, 1991.

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

[Vapnik, 1982] V. Vapnik. *Estimations of dependences based on statistical data.* Springer Publisher, 1982.

[Yang and Chute, 1993] Yiming Yang and Christopher G. Chute. An application of least squares fit mapping to text information retrieval. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference*, 1993.

[Yang and Pederson, 1997] Yiming Yang and Jan Pederson. Feature selection in statistical learning of text categorization. In *ICML-97*, pages 412–420, 1997.