Learning to Classify Text from Labeled and Unlabeled Documents

Kamal Nigam[†] knigam@cs.cmu.edu
Andrew McCallum[‡†] mccallum@cs.cmu.edu
Sebastian Thrun[†] thrun@cs.cmu.edu
Tom Mitchell[†] mitchell+@cs.cmu.edu

[†]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[‡]Just Research
4616 Henry Street
Pittsburgh, PA 15213

Abstract

In many important text classification problems, acquiring class labels for training documents is costly, while gathering large quantities of unlabeled data is cheap. This paper shows that the accuracy of text classifiers trained with a small number of labeled documents can be improved by augmenting this small training set with a large pool of unlabeled documents. 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 to convergence. Experimental results, obtained using text from three different real-world tasks, show that the use of unlabeled data reduces classification error by up to 33%.

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 & Ringuette 1994; Joachims 1998) and web pages (Craven *et al.* 1998), automatically learn the reading interests of users (Pazzani, Muramatsu, & Billsus 1996; Lang 1995), and automatically sort electronic mail (Lewis & 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 postings are of interest to a person reading UseNet news, as examined by Lang (1995). After reading and classifying about 1000 postings, 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 dozen postings, rather than a thousand.

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 postings 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 postings 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 difficult.

We present experimental results showing that this unlabeled data can boost learning accuracy in three text classification domains: newsgroup postings, web pages, and newswire articles. For example, to identify the source newsgroup for a UseNet posting 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. So, in this case, 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 "lecture" tend to belong to the class of academic course web pages. If we use this fact to estimate the classification of the many unlabeled documents, we might find that the word "homework" occurs frequently in the unlabeled examples that are now believed to belong to the course class. Thus the co-occurrence of the words "lecture" and "homework" over the large set of unlabeled training data can provide useful information to construct a more accurate classifier that considers both "lecture" and "homework" as indicators of course web pages.

The specific approach we describe here is based on a combination of two well-known learning algorithms: the naive Bayes classifier (Lewis & Ringuette

1994; McCallum & Nigam 1998) and the Expectation-Maximization (EM) algorithm (Dempster, Laird, & Rubin 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), Support Vector Machines (Joachims 1998), k-nearest-neighbor (Yang & Pederson 1997), exponentiated-gradient and regression models (Lewis *et al.* 1996). 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 (1997) for non-text data sources. We adapt this approach for the naive Bayes text classifier and conduct a thorough empirical analysis. Furthermore, we show theoretically that, under certain conditions, unlabeled data carry information useful for improving parameter estimation and classification accuracy. We also introduce a method for balancing the contributions of the labeled and unlabeled data that avoids degradation in classification accuracy when these conditions are violated. A more detailed version of this paper is available (Nigam *et al.* 1998).

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. The framework follows commonly used assumptions (Lewis & Ringuette 1994; Domingos & Pazzani 1997) about the data: (1) that our data is produced by a mixture model, and (2) that there is a one-to-one correspondence between generative 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 generative components $c_j \in \mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$, each component being parameterized by a disjoint subset of θ. Thus a document is created by (1) selecting a component according to the prior probabilities, $\mathrm{P}(c_j|\theta)$, then (2) having that 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 generative 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. The class label of document d_i is written y_i, and if document d_i was generated by component c_j we say $c_j = c_{y_i}$. This class label may or may not be known for a given document.

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. For unlabeled data to carry information about the parameters θ, it is sufficient that the learning task is not *degenerate*, that is,

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

This assumption excludes tasks where learning is impossible, for the reason that all parameterizations θ' yield equivalent results. High-dimensional mixture models meet this condition.

To show that knowledge about unlabeled documents carries information about the parameters θ, we need to demonstrate the conditional dependence of θ on D, a random variable over the unlabeled document distribution. That is, show $\mathrm{P}(\theta|D) \neq \mathrm{P}(\theta)$. If this conjecture holds, a direct implication is that unlabeled data contain information about the parameters of θ. We provide a proof by contradiction. For this, we temporarily assume that θ and D are independent, that is, $\forall\theta'.\ \mathrm{P}(\theta'|D) = \mathrm{P}(\theta')$. One direct conclusion of this assumption is that any two parameterizations, θ' and θ'', provide the same class probabilities for any sample. By applying Bayes' rule to our assumption, and substituting using Equation 1 this gives:

$$\sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(d_i|c_j;\theta')\mathrm{P}(c_j|\theta') = \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(d_i|c_j;\theta'')\mathrm{P}(c_j|\theta''). \tag{3}$$

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 3 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.

However, this does not show that unlabeled data aids the reduction of classification error. For example, unlabeled data does not help if there is already an infinite amount of labeled data; all parameters can be recovered from just the labeled data and the resulting classifier is Bayes-optimal (McLachlan & Basford 1988). With an infinite amount of unlabeled data and no labeled data, the parameters can be estimated except classes cannot be matched with components, so classification error remains unimproved. But, with infinite unlabeled data and finite labeled data, there is classification improvement. With infinite unlabeled data, the classification error approaches the Bayes optimal solution at an exponential rate in the number of labeled examples given (Castelli & Cover 1995). Thus, infinite amounts of unlabeled data are shown to help classification when there is some, but not infinite, amounts of labeled data. Unfortunately, little is known for the case in which there are finite amounts of each.

Naive Bayes for Text Classification

In this section, we present the naive Bayes classifier—a well-known, probabilistic algorithm for classifying text that is one instantiation of a mixture model. This algorithm forms the foundation upon which we will later incorporate unlabeled data. The learning task for the naive Bayes classifier is to use a set of training documents to estimate the mixture model parameters, then use the estimated model to classify new documents.

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 $V = \langle w_1, w_2, \ldots, w_{|V|}\rangle$. In order to generate a document, after a mixture component is selected, a document length is chosen independently of the component and the words in the document. Then, the selected mixture component generates a sequence words of the specified length. Thus, we can expand the second term from Equation 1, and correctly express the probability of a document given its class using the general multiplication rule over the sequence of individual word events:

$$\mathrm{P}(d_i|c_j;\theta) = \mathrm{P}(|d_i|)\prod_{k=1}^{|d_i|}\mathrm{P}(w_{d_{ik}}|c_j;\theta;w_{d_{iq}}, \forall q < k). \tag{4}$$

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. We further assume that the probability of a word is independent of its position within the document. Thus, we can rewrite Equation 4 more simply as:

$$\mathrm{P}(d_i|c_j;\theta) = \mathrm{P}(|d_i|)\prod_{k=1}^{|d_i|}\mathrm{P}(w_{d_{ik}}|c_j;\theta). \tag{5}$$

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 $\sum_t \mathrm{P}(w_t|c_j;\theta) = 1$. The other parameters of the model are the class prior probabilities $\theta_{c_j} = \mathrm{P}(c_j|\theta)$, which indicate the probabilities of selecting the different mixture components. Document length is not parameterized because we assume it is independent of classification.

Given these underlying assumptions of how the data is produced, the task of learning a text classifier consists of forming an estimate of θ, written $\hat{\theta}$, based on a set of training data. 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 (Vapnik 1982). To calculate the probability of a word given a class, $\theta_{w_t|c_j}$, simply count the fraction of times the word occurs in the data for that class, augmented with a Laplacean prior. This smoothing prevents zero probabilities for infrequently occurring words. These word probability estimates $\hat{\theta}_{w_t|c_j}$ are:

$$\hat{\theta}_{w_t|c_j} = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} N(w_t, d_i)\mathrm{P}(c_j|d_i)}{|V| + \sum_{s=1}^{|V|}\sum_{i=1}^{|\mathcal{D}|} N(w_s, d_i)\mathrm{P}(c_j|d_i)}, \tag{6}$$

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, but without smoothing:

$$\hat{\theta}_{c_j} = \frac{\sum_{i=1}^{|\mathcal{D}|}\mathrm{P}(c_j|d_i)}{|\mathcal{D}|}. \tag{7}$$

Given estimates of these parameters calculated from the training documents, it is possible to turn the generative model around and calculate the probability that a particular component generated a given document. We formulate this by an application of Bayes' rule, and then substitutions using Equations 1 and 5:

$$\begin{aligned} \mathrm{P}(c_j|d_i;\hat{\theta}) &= \frac{\mathrm{P}(c_j|\hat{\theta})\mathrm{P}(d_i|c_j;\hat{\theta})}{\mathrm{P}(d_i|\hat{\theta})} \\ &= \frac{\mathrm{P}(c_j|\hat{\theta})\prod_{k=1}^{|d_i|}\mathrm{P}(w_{d_{ik}}|c_j;\hat{\theta})}{\sum_{r=1}^{|\mathcal{C}|}\mathrm{P}(c_r|\hat{\theta})\prod_{k=1}^{|d_i|}\mathrm{P}(w_{d_{ik}}|c_r;\hat{\theta})}. \end{aligned} \tag{8}$$

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 assumptions about the generation of text documents are all violated in practice, and yet empirically, naive Bayes does a good job of classifying text documents (Lewis & Ringuette 1994; Craven *et al.* 1998; Yang & Pederson 1997; Joachims 1997). This paradox is explained by the fact that classification estimation is only a function of the sign (in binary cases) of the function estimation (Domingos & Pazzani 1997; Friedman 1997). Also note that our formulation of naive Bayes assumes a multinomial event model for documents; this generally produces better text classification accuracy than another formulation that assumes a multi-variate Bernoulli (McCallum & Nigam 1998).

Incorporating Unlabeled Data with EM

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. As described in Table 1, EM alternately generates probabilistically-weighted labels for the unlabeled documents, and a more probable model with smaller parameter variance. EM finds a local maximum likelihood parameterization using more data—both the labeled and the unlabeled. 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).

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 of the documents $d_i \in \mathcal{D}^l$ come with class labels $y_i \in \{1, \ldots, |\mathcal{C}|\}$, and for the rest of the documents, in subset $\mathcal{D}^u$, the class labels are unknown. Thus there is 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. Using Equation 1, this is:

$$\begin{aligned} \mathrm{P}(\mathcal{D}|\theta) &= \prod_{d_i \in \mathcal{D}^u} \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta) \\ &\quad \times \prod_{d_i \in \mathcal{D}^l} \mathrm{P}(c_{y_i}|\theta)\mathrm{P}(d_i|c_{y_i};\theta). \end{aligned} \tag{9}$$

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 in our context 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, $\mathrm{P}(\theta|\mathcal{D}) = \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)/\mathrm{P}(\mathcal{D})$. $\mathrm{P}(\mathcal{D})$ is a constant; maximum likelihood estimation assumes that $\mathrm{P}(\theta)$ is a constant, so taking the log, we define the constant $\eta = \log(\mathrm{P}(\theta)/\mathrm{P}(\mathcal{D}))$. Maximizing the log likelihood is the same as maximizing the likelihood. Using Equation 9 and Bayes rule, we write the log likelihood, $l(\theta|\mathcal{D}) \equiv \log(\mathrm{P}(\theta|\mathcal{D}))$, as:

```
- Build an initial classifier by calculating $\hat{\theta}$ from the labeled documents only (Equations 6 and 7).
- Loop while classifier parameters change:
    - Use the current classifier to calculate probabilistically-weighted labels for the unlabeled documents (Equation 8).
    - Recalculate the classifier parameters $\hat{\theta}$ given the probabilistically assigned labels (Equations 6 and 7).
```

Table 1: The Algorithm.

$$\begin{aligned} l(\theta|\mathcal{D}) &= \eta + \sum_{d_i \in \mathcal{D}^u} \log \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta) \\ &\quad + \sum_{d_i \in \mathcal{D}^l} \log\left(\mathrm{P}(c_{y_i}|\theta)\mathrm{P}(d_i|c_{y_i};\theta)\right). \end{aligned} \tag{10}$$

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 log of sums. Representing this potential knowledge about the class labels 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 $y_i = j$ else $z_{ij} = 0$, we can express the complete log likelihood of the parameters, $l_c(\theta|\mathcal{D}, \mathbf{z})$:

$$l_c(\theta|\mathcal{D}, \mathbf{z}) = \eta + \sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{C}|} z_{ij} \log\left(\mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta_j)\right). \tag{11}$$

This formulation of the log likelihood would be readily computable in closed-form. Dempster, Laird and Rubin (1977) use this insight in the formulation of 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 $\mathrm{P}(c_j|d_i;\hat{\theta})$ for every document by using the current estimate $\hat{\theta}$ and Equation 8. The M-step corresponds to calculating a new maximum likelihood estimate for θ given the current estimates for document labels, $\mathrm{P}(c_j|d_i;\hat{\theta})$ using Equations 6 and 7. See Table 1 for an outline of our algorithm. In summary, EM finds the $\hat{\theta}$ that locally maximizes the probability of all the data, both the labeled and the unlabeled.

Experimental Results

In this section, we give empirical evidence that using the algorithm in Table 1 outperforms traditional naive Bayes. We present experimental results with three different text corpora.[1]

Datasets and Protocol

The 20 Newsgroups data set (Joachims 1997), collected by Ken Lang, consists of 20,017 articles divided almost evenly among 20 different UseNet discussion groups. We remove words from a stoplist of common short words and words that occur only once. 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 web pages gathered from university computer science departments. In this paper, we use the four most populous entity-representing 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 (Yang & Pederson 1997; Joachims 1997). Accuracy results presented below are an average of twenty test/train splits, again randomly holding out 20% of the documents for testing.

The 'ModApte' test/train split of the Reuters 21578 Distribution 1.0 data set consists of 12902 articles and 135 topic categories from the Reuters newswire. Following other studies (Joachims 1998) we present results on the 10 most populous classes, building binary classifiers for each class that include all 134 other classes in the negative category. We use a stoplist, but do not stem. Vocabulary selection, when used, is again performed with average mutual information with the class variable. Results are reported as averages of ten randomly selected subsets of ModApte's training set. The complete ModApte test set is used to calculate precision-recall breakeven points, a standard information retrieval measure for binary classification.

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.

[1]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.

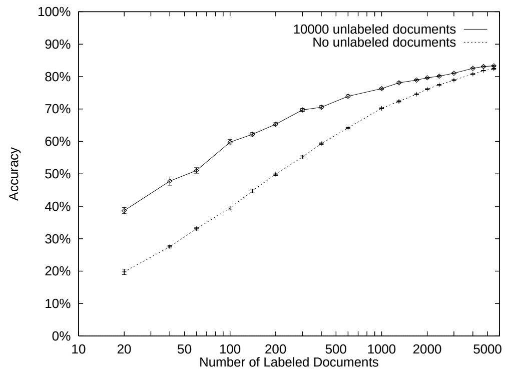

Figure 1: Classification accuracy on the 20 Newsgroups data set. The narrow error bars on each data point are twice the standard error.

Results

Figure 1 shows the effect of using EM with unlabeled data in the 20 Newsgroups data set. 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 300 labeled documents (15 documents per class), naive Bayes reaches 55% accuracy, while EM achieves 70%—providing a 33% reduction in error. 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 flattened, having unlabeled data does not help.

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 70% classification accuracy, EM requires 300 labeled examples, while naive Bayes requires 1000 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" of the course class. Words are judged to be "predictive" using a weighted log likelihood ratio. 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 (final) iteration, the classifier achieves 71% accuracy.

Iteration 0	Iteration 1	Iteration 2
intelligence	*DD*	*D*
DD	*D*	*DD*
artificial	lecture	lecture
understanding	cc	cc
*DD*w	$D^\star$	*DD:DD*
dist	*DD:DD*	due
identical	handout	$D^\star$
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. The symbol *D* indicates an arbitrary digit.

The graph in Figure 2 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.

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 does—that is, the generative components that best explain the unlabeled data are not in good correspondence with the class labels. As seen in Figure 2, EM can still help in spite of somewhat violated assumptions when EM has very little labeled training data, because parameter estimation is so desperate for guidance. 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 incorporation 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). Thus, the great majority of the probability mass used in the M-step 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 in the M-step. In our implementation this parameter is embodied by a

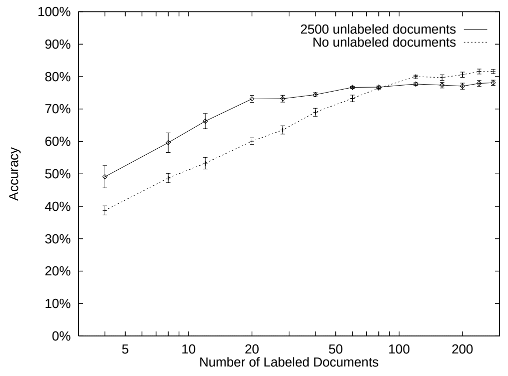

Figure 2: Classification accuracy on the WebKB data set, both with and without 2500 unlabeled documents, averaged over 20 trials per data point.

factor, $0 \leq \alpha \leq 1$, that reduces the weight of unlabeled documents in the estimation of $\theta_{w_t|c_j}$ in Equation 6. 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 value using leave-one-out cross-validation on the training data after EM has iterated to convergence.

Re-running the WebKB experiment in Figure 2 with α selected by cross-validation provides two results of note: (1) Cross-validation picks the optimal value most of the time, and a near-optimal value otherwise. These optimal values for α do not fall at the 0/1 extremes, and are smaller when there is a lot of labeled data, and larger when there is little. (2) The accuracy of this classifier is equal to or slightly higher than the maximum of naive Bayes and EM without α-tuning. This result indicates that, even when our assumptions about the correspondence between generative components and classes are violated, we can automatically avoid any degradation in accuracy when using EM, and can still preserve the performance improvements seen when the labeled data is scarce.

Multiple Generative 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. Flexibility can be added to the mapping between generative components and class labels by allowing *multiple* components per class—relaxing our assumption about a one-to-one correspondence between generative components and classes into a model with a many-to-one correspondence. We expect this to improve performance when data for each class is actually multi-modal.

With an eye towards testing this hypothesis, we apply EM to the Reuters corpus. Since the documents in this

Category	NB 1	EM 1	EM 20	EM 40	Diff
acq	75.9	39.5	88.4	**88.9**	+13.0
corn	40.5	21.1	**39.8**	39.1	-0.7
crude	60.7	27.8	63.9	**66.6**	+5.9
earn	92.6	90.2	**95.3**	95.2	+2.7
grain	51.7	21.0	54.6	**55.8**	+4.1
interest	52.0	25.9	48.6	**50.3**	-1.7
money-fx	57.7	28.8	54.7	**59.7**	+2.0
ship	58.1	9.3	46.5	**55.0**	-3.1
trade	56.8	34.7	54.3	**57.0**	+0.2
wheat	48.9	13.0	42.1	**44.2**	-4.7

Table 3: Precision-Recall breakeven points showing performance of binary classifiers on Reuters with traditional naive Bayes, EM with one generative 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.

data set can have multiple class labels, each category is traditionally evaluated with a binary classifier. Thus, the negative class covers 134 distinct categories, and we expect that this task strongly violates the assumption that all the data for the negative class is generated by a single component. For these experiments, we randomly selected 10 positively labeled documents, 40 negatively labeled documents, and 7000 unlabeled documents.

The left column of Table 3 shows average precision-recall breakeven points for naive Bayes. These numbers are presented at the best vocabulary size for each task. The second column of Table 3 shows the results of performing EM on the data with a single negative generative component, as in previous experiments, without α-tuning. As expected, EM's results are dramatically worse than traditional naive Bayes because fitting a single naive Bayes component with EM to multi-modal negative data does not accurately capture its distribution. However, by running EM on an appropriate model with multiple components per class, we obtain results that improve upon naive Bayes. The remainder of Table 3 shows the effects of modeling the negative class with 20 or 40 generative components. These components are each initialized for EM with randomly assigned negative documents. 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 ($p < 0.0001$).

These results indicate that correct model selection is crucial for EM when there is not a simple one-to-one correspondence between generative components and classes. When the data is accurately modeled, EM provides significant gains in performance. One obvious question is how to select the best model. AutoClass (Cheeseman & Stutz 1996) does this for unsupervised clustering tasks by selecting the most probable model given the data and a prior that prefers smaller models. For classification tasks, it may be more beneficial to use classification accuracy with leave-one-out cross-validation, as was successful for α-tuning.

Related Work

Two other studies use EM to combine labeled and unlabeled data for classification (Miller & Uyar 1997; Shahshahani & 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. Shahshahani and Landgrebe present a proof that unlabeled data reduce variance in parameter estimation. Their proof does not apply in our case, however, because our target concept can not be learned without some labeled data. (There is no efficient estimator for *all* parameters.)

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 (1994) use EM with mixture models to fill in missing values. The emphasis of their work is on missing feature values, where we focus on augmenting a very small but complete set of labeled data.

The AutoClass project (Cheeseman & Stutz 1996) investigates the combination of the EM algorithm with an underlying model of a naive Bayes classifier. The emphasis of their research is the discovery of novel clusterings for unsupervised learning over unlabeled data. AutoClass has not been applied to text or classification.

Several other text classifiers have been used by others in a variety of domains (Yang & Pederson 1997; Joachims 1998; Cohen & Singer 1997). 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.

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.

Since the complexity of natural language text will not soon be completely captured by statistical models, it is interesting to consider the sensitivity of a classifier's model to data that is inconsistent with that model. We believe that our algorithm and others using unlabeled data require a closer match between the data and the model than those using only labeled data. 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 the correspondence between generative 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 see several interesting directions for future work using EM and unlabeled data. Work in progress suggests that active learning can benefit from explicitly modeling the unlabeled data by incorporating EM iterations at every stage; this allows better selection of examples for which to request class labels from a labeler. Also, 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 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.

Acknowledgments

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

Cheeseman, P., and Stutz, J. 1996. Bayesian classification (AutoClass): Theory and results. In Fayyad, U.; Piatetski-Shapiro, G.; Smyth, P.; and Uthurusamy, R., eds., *Advances in Knowledge Discovery and Data Mining*. Cambridge: MIT Press.

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

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

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

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

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

Ghahramani, Z., and Jordan, M. 1994. Supervised learning from incomplete data via an EM approach. In *Advances in Neural Information Processing Systems (NIPS 6)*.

Joachims, T. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Intl. Conference on Machine Learning (ICML-97)*.

Joachims, T. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. In *European Conference on Machine Learning (ECML-98)*.

Lang, K. 1995. Newsweeder: Learning to filter netnews. In *Intl. Conference on Machine Learning (ICML-95)*.

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

Lewis, D., and Ringuette, M. 1994. A comparison of two learning algorithms for text categorization. In *Third Annual Symposium on Document Analysis and IR*.

Lewis, D.; Schapire, R.; Callan, J.; and Papka, R. 1996. Training algorithms for linear text classifiers. In *Proceedings of ACM SIGIR Conference*.

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

McLachlan, G., and Basford, K. 1988. *Mixture Models*. New York: Marcel Dekker.

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

Nigam, K.; McCallum, A.; Thrun, S.; and Mitchell, T. 1998. Learning to classify text from labeled and unlabeled documents. Technical Report CMU-CS-98-120, Carnegie Mellon University.

Pazzani, M. J.; Muramatsu, J.; and Billsus, D. 1996. Syskill & Webert: Identifying interesting Web sites. In *Proceedings of AAAI-96*.

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

Shahshahani, B., and Landgrebe, D. 1994. 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.

Vapnik, V. 1982. *Estimation of dependences based on empirical data*. Springer.

Yang, Y., and Pederson, J. 1997. Feature selection in statistical learning of text categorization. In *Intl Conference on Machine Learning (ICML-97)*, 412–420.