Machine Learning, , 1–34 ()

Text Classification from Labeled and Unlabeled Documents using EM

KAMAL NIGAM[†] knigam@cs.cmu.edu
ANDREW KACHITES MCCALLUM[‡†] mccallum@justresearch.com
SEBASTIAN THRUN[†] thrun@cs.cmu.edu
TOM MITCHELL[†] tom.mitchell@cmu.edu
[†]*School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213*
[‡]*Just Research, 4616 Henry Street, Pittsburgh, PA 15213*

Received March 15, 1998; Revised February 20, 1999

Editor: William W. Cohen

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 important because in many text classification problems obtaining training labels is expensive, while large quantities of unlabeled documents are readily available.

We introduce an algorithm for learning from labeled and unlabeled documents based on the combination of Expectation-Maximization (EM) and 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. This basic EM procedure works well when the data conform to the generative assumptions of the model. However these assumptions are often violated in practice, and poor performance can result. We present two extensions to the algorithm that improve classification accuracy under these conditions: (1) a weighting factor to modulate the contribution of the unlabeled data, and (2) the use of multiple mixture components per class. 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%.

Keywords: text classification, Expectation-Maximization, integrating supervised and unsupervised learning, combining labeled and unlabeled data, Bayesian learning

1. Introduction

Consider the problem of automatically classifying text documents. This problem is of great practical importance given the massive volume of online text available through the World Wide Web, Internet news feeds, electronic mail, corporate databases, medical patient records and digital libraries. Existing statistical text learning algorithms 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 & Gale, 1994; Joachims, 1998) and web pages (Craven, DiPasquo, Freitag, McCallum, Mitchell, Nigam, & Slattery, 1998; Shavlik & Eliassi-Rad, 1998), automatically learn the reading interests of users (Pazzani, Muramatsu, & Billsus, 1996; Lang, 1995), and automatically sort electronic mail (Lewis & Knowles, 1997; Sahami, Dumais, Heckerman, & Horvitz, 1998).

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. Labeling must often be done by a person; this is a painfully time-consuming process.

Take, for example, the task of learning which UseNet newsgroup articles are of interest to a particular person reading UseNet news. Systems that filter or pre-sort articles and present only the ones the user finds interesting are highly desirable, and are of great commercial interest today. Work by Lang (1995) found that after a person read and labeled about 1000 articles, a learned classifier achieved a precision of about 50% when making predictions for only the top 10% of documents about which it was most confident. Most users of a practical system, however, would not have the patience to label a thousand articles—especially to obtain only this level of precision. One would obviously prefer algorithms that can provide accurate classifications after hand-labeling only a few dozen articles, rather than thousands.

The need for large quantities of data to obtain high accuracy, and the difficulty of obtaining labeled data, raises an important question: what other sources of information can reduce the need for labeled data?

This paper addresses the problem of learning accurate text classifiers from limited numbers of labeled examples by using *unlabeled* documents to augment the available *labeled* documents. In many text domains, especially those involving online sources, collecting unlabeled documents is easy and inexpensive. The filtering task above, where there are thousands of unlabeled articles freely available on UseNet, is one such example. It is the labeling, not the collecting of documents, that is expensive.

How is it that unlabeled data can increase classification accuracy? At first consideration, one might be inclined to think that nothing is to be gained by access to unlabeled data. However, they do provide information about the joint probability distribution over words. Suppose, for example, that using only the labeled data we determine that documents containing the word "homework" 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 "lecture" occurs frequently in the unlabeled examples that are now believed to belong to the positive class. This co-occurrence of the words "homework" and "lecture" over the large set of unlabeled training data can provide useful information to construct a more accurate classifier that considers both "homework" and "lecture" as indicators of positive examples. In this paper, we explain that such correlations are a helpful source of information for increasing classification rates, specifically when labeled data are scarce.

This paper uses Expectation-Maximization (EM) to learn classifiers that take advantage of both labeled and unlabeled data. EM is a class of iterative algorithms for maximum likelihood or maximum a posteriori estimation in problems with incomplete data (Dempster, Laird, & Rubin, 1977). In our case, the unlabeled data are considered incomplete because they come without class labels. The algorithm first trains a classifier with only the available *labeled* documents, and uses the classifier to assign probabilistically-weighted class labels to each unlabeled document by calculating the expectation of the missing class labels. It then trains a new classifier using all the documents—both the originally labeled and the formerly unlabeled—and iterates. In its maximum likelihood formulation, EM performs hill-climbing in data likelihood space, finding the classifier parameters that locally maximize the likelihood of all the data—both the labeled and the unlabeled. We combine EM with naive Bayes, a classifier based on a mixture of multinomials, that is commonly used in text classification.

We also propose two augmentations to the basic EM scheme. In order for basic EM to improve classifier accuracy, several assumptions about how the data are generated must be satisfied. The assumptions are that the data are generated by a mixture model, and that there is a correspondence between mixture components and classes. When these assumptions are not satisfied, EM may actually degrade rather than improve classifier accuracy. Since these assumptions rarely hold in real-world data, we propose extensions to the basic EM/naive-Bayes combination that allow unlabeled data to still improve classification accuracy, in spite of violated assumptions. The first extension introduces a weighting factor that dynamically adjusts the strength of the unlabeled data's contribution to parameter estimation in EM. The second reduces the bias of naive Bayes by modeling each class with multiple mixture components, instead of a single component.

Over the course of several experimental comparisons, we show that (1) unlabeled data can significantly increase performance, (2) the basic EM algorithm can suffer from a misfit between the modeling assumptions and the unlabeled data, and (3) each extension mentioned above often reduces the effect of this problem and improves classification.

The reduction in the number of labeled examples needed can be dramatic. For example, to identify the source newsgroup for a UseNet article with 70% classification accuracy, a traditional learner requires 2000 labeled examples; alternatively our algorithm takes advantage of 10000 unlabeled examples and requires only 600 labeled examples to achieve the same accuracy. Thus, in this task, the technique reduces the need for labeled training examples by more than a factor of three. With only 40 labeled documents (two per class), accuracy is improved from 27% to 43% by adding unlabeled data. These findings illustrate the power of unlabeled data in text classification problems, and also demonstrate the strength of the algorithms proposed here.

The remainder of the paper is organized as follows. Section 2 describes, from a theoretical point of view, the problem of learning from labeled and unlabeled data. Sections 3 and 4 present the formal framework for naive Bayes. In Section 5, we present the combination of EM and naive Bayes, and our extensions to this algorithm. Section 6 describes a systematic experimental comparison using three classification domains: newsgroup articles, web pages, and newswire articles. The first two domains are multi-class classification problems where each class is relatively frequent. The third domain is treated as binary classification, with the "positive" class having a frequency between 1% and 30%, depending on the task. Related work is discussed in Section 7. Finally, advantages, limitations, and future research directions are discussed in Section 8.

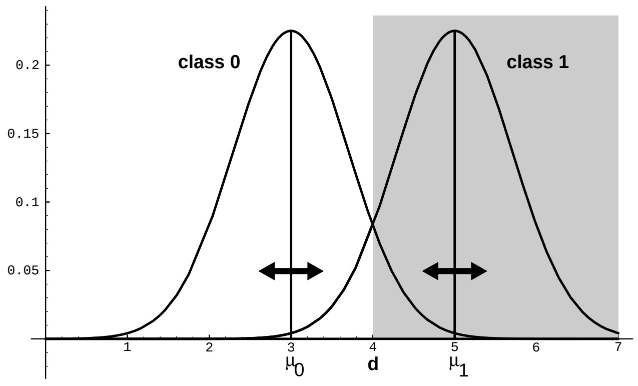

Figure 1. Classification by a mixture of Gaussians. If unlimited amounts of unlabeled data are available, the mixture components can be fully recovered, and labeled data are used to assign labels to the individual components, converging exponentially quickly to the Bayes-optimal classifier.

2. Argument for the Value of Unlabeled Data

How are unlabeled data useful when learning classification? Unlabeled data *alone* are generally insufficient to yield better-than-random classification because there is no information about the class label (Castelli & Cover, 1995). However, unlabeled data do contain information about the joint distribution over features other than the class label. Because of this they can sometimes be used—together with a sample of labeled data—to significantly increase classification accuracy in certain problem settings.

To see this, consider a simple classification problem—one in which instances are generated using a Gaussian mixture model. Here, data are generated according to two Gaussian distributions, one per class, whose parameters are unknown. Figure 1 illustrates the Bayes-optimal decision boundary ($x > d$), which classifies instances into the two classes shown by the shaded and unshaded areas. Note that it is possible to calculate d from Bayes rule if we know the Gaussian mixture distribution parameters (*i.e.*, the mean and variance of each Gaussian, and the mixing parameter between them).

Consider when an infinite amount of unlabeled data is available, along with a finite number of labeled samples. It is well known that unlabeled data alone, when generated from a mixture of two Gaussians, are sufficient to recover the original mixture components (McLachlan & Krishnan, 1997, section 2.7). However, it is impossible to assign class labels to each of the Gaussians without any labeled data. Thus, the remaining learning problem is the problem of assigning class labels to the two Gaussians. For instance, in Figure 1, the means, variances, and mixture parameter can be learned with unlabeled data alone. Labeled data must be used to determine which Gaussian belongs to which class. This problem is known to converge exponentially quickly in the number of labeled samples (Castelli & Cover, 1995). Informally, as long as there are enough labeled examples to determine the class of each component, the parameter estimation can be done with unlabeled data alone.

It is important to notice that this result depends on the critical assumption that the data indeed have been generated using the same parametric model as used in classification, something that almost certainly is untrue in real-world domains such as text classification. This raises the important empirical question as to what extent unlabeled data can be useful in practice in spite of the violated assumptions. In the following sections we address this by describing in detail a parametric generative model for text classification and by presenting empirical results using this model on real-world data.

3. The Probabilistic Framework

This section presents a probabilistic framework for characterizing the nature of documents and classifiers. The framework defines a probabilistic generative model for the data, and embodies two assumptions about the generative process: (1) the data are produced by a mixture model, and (2) there is a one-to-one correspondence between mixture components and classes.[1] The naive Bayes text classifier we will discuss later falls into this framework, as does the example in Section 2.

In this setting, every document is generated according to a probability distribution defined by a set of parameters, denoted θ. The probability distribution consists of a mixture of components $c_j \in \mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$. Each component is parameterized by a disjoint subset of θ. A document, d_i, is created by first selecting a mixture component according to the mixture weights (or class prior probabilities), $\mathrm{P}(c_j|\theta)$, then having this selected mixture component generate a document according to its own parameters, with distribution $\mathrm{P}(d_i|c_j;\theta)$.[2] Thus, we can characterize the likelihood of document d_i 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 mixture model components and classes, and thus (for the time being) use c_j to indicate the jth mixture component as well as, the jth class. The class label for a particular document d_i is written y_i. If document d_i was generated by mixture component c_j we say $y_i = c_j$. The class label may or may not be known for a given document.

4. Text Classification with Naive Bayes

This section presents naive Bayes—a well-known probabilistic classifier—and describes its application to text. Naive Bayes is the foundation upon which we will later build in order to incorporate unlabeled data.

The learning task in this section is to estimate the parameters of a generative model using labeled training data only. The algorithm uses the estimated parameters to classify new documents by calculating which class was most likely to have generated the given document.

4.1. The Generative Model

Naive Bayes assumes a particular probabilistic generative model for text. The model is a specialization of the mixture model presented in the previous section, and thus also makes the two assumptions discussed there. Additionally, naive Bayes makes word independence assumptions that allow the generative model to be characterized with a greatly reduced number of parameters. The rest of this subsection describes the generative model more formally, giving a precise specification of the model parameters, and deriving the probability that a particular document is generated given its class label (Equation 4).

First let us introduce some notation to describe text. A document, d_i, is considered to be an ordered list of word events, $\langle w_{d_{i,1}}, w_{d_{i,2}}, \ldots \rangle$. We write $w_{d_{i,k}}$ for the word w_t in position k of document d_i, where w_t is a word in the vocabulary $V = \langle w_1, w_2, \ldots, w_{|V|} \rangle$.

When a document is to be generated by a particular mixture component, c_j, a document length, $|d_i|$, is chosen independently of the component. (Note that this assumes that document length is independent of class.[3]) Then, the selected mixture component generates a word sequence of the specified length. We furthermore assume it generates each word independently of the length.

Thus, we can expand the second term from Equation 1, and express the probability of a document given a mixture component in terms of its constituent features: the document length and the words in the document. Note that, in this general setting, the probability of a word event must be conditioned on all the words that precede it.

$$\mathrm{P}(d_i|c_j;\theta) = \mathrm{P}(\langle w_{d_{i,1}}, \ldots, w_{d_{i,|d_i|}} \rangle | c_j;\theta) = \mathrm{P}(|d_i|) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_j;\theta; w_{d_{i,q}}, q < k) \quad (2)$$

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 "homework" in the first position of a document is the same as seeing it in any other position. We can express these assumptions as:

$$\mathrm{P}(w_{d_{i,k}}|c_j;\theta; w_{d_{i,q}}, q < k) = \mathrm{P}(w_{d_{i,k}}|c_j;\theta). \quad (3)$$

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

$$\mathrm{P}(d_i|c_j;\theta) = \mathrm{P}(|d_i|) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_j;\theta). \quad (4)$$

Thus the parameters of an individual mixture component are a multinomial distribution over words, *i.e.* the collection of word probabilities, each written $\theta_{w_t|c_j}$, 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 identically distributed, it does not need to be parameterized for classification. The only other parameters of the model are the mixture weights (class prior probabilities), written θ_{c_j}, which indicate the probabilities of selecting the different mixture components. Thus the complete collection of model parameters, θ, is a set of multinomials and prior probabilities over those multinomials: $\theta = \{\theta_{w_t|c_j} : w_t \in V, c_j \in \mathcal{C}\ ;\ \theta_{c_j} : c_j \in \mathcal{C}\}$.

4.2. Training a Classifier

Learning a naive Bayes text classifier consists of estimating the parameters of the generative model by using a set of labeled training data, $\mathcal{D} = \{d_1, \ldots, d_{|\mathcal{D}|}\}$. This subsection derives a method for calculating these estimates from the training data.

The estimate of θ is written $\hat{\theta}$. Naive Bayes uses the maximum a posteriori estimate, thus finding $\arg\max_\theta \mathrm{P}(\theta|\mathcal{D})$. This is the value of θ that is most probable given the evidence of the training data and a prior.

The parameter estimation formulae that result from this maximization are the familiar ratios of empirical counts. The estimated probability of a word given a class, $\hat{\theta}_{w_t|c_j}$, is simply the number of times word w_t occurs in the training data for class c_j, divided by the total number of word occurrences in the training data for that class—where counts in both the numerator and denominator are augmented with "pseudo-counts" (one for each word) that come from the prior distribution over θ. The use of this type of prior is sometimes referred to as *Laplace smoothing*. Smoothing is necessary to prevent zero probabilities for infrequently occurring words.

The word probability estimates $\hat{\theta}_{w_t|c_j}$ are:

$$\hat{\theta}_{w_t|c_j} \equiv \mathrm{P}(w_t|c_j;\hat{\theta}) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} N(w_t, d_i)\mathrm{P}(y_i = c_j|d_i)}{|V| + \sum_{s=1}^{|V|}\sum_{i=1}^{|\mathcal{D}|} N(w_s, d_i)\mathrm{P}(y_i = c_j|d_i)}, \tag{5}$$

where $N(w_t, d_i)$ is the count of the number of times word w_t occurs in document d_i and where $\mathrm{P}(y_i = c_j|d_i) \in \{0, 1\}$ as given by the class label.

The class prior probabilities, $\hat{\theta}_{c_j}$, are estimated in the same manner, and also involve a ratio of counts with smoothing:

$$\hat{\theta}_{c_j} \equiv \mathrm{P}(c_j|\hat{\theta}) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(y_i = c_j|d_i)}{|\mathcal{C}| + |\mathcal{D}|}. \tag{6}$$

The derivation of these "ratios of counts" formulae comes directly from maximum a posteriori parameter estimation, and will be appealed to again later when deriving parameter estimation formulae for EM and augmented EM. Finding the θ that maximizes $\mathrm{P}(\theta|\mathcal{D})$ is accomplished by first breaking this expression into two terms by Bayes' rule: $\mathrm{P}(\theta|\mathcal{D}) \propto \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$. The first term is calculated by the product of all the document likelihoods (from Equation 1). The second term, the prior distribution over parameters, we represent by a Dirichlet distribution: $\mathrm{P}(\theta) \propto \prod_{c_j \in \mathcal{C}} \left((\theta_{c_j})^{\alpha-1} \prod_{w_t \in V} (\theta_{w_t|c_j})^{\alpha-1}\right)$, where α is a parameter that effects the strength of the prior, and is some constant greater than zero.[4] In this paper, we set $\alpha = 2$, which (with maximum a posteriori estimation) is equivalent to Laplace smoothing. The whole expression is maximized by solving the system of partial derivatives of $\log(\mathrm{P}(\theta|\mathcal{D}))$, using Lagrange multipliers to enforce the constraint that the word probabilities in a class must sum to one. This maximization yields the ratio of counts seen above.

4.3. Using a Classifier

Given estimates of these parameters calculated from the training documents according to Equations 5 and 6, it is possible to turn the generative model backwards and calculate the probability that a particular mixture component generated a given document. We derive this by an application of Bayes' rule, and then by substitutions using Equations 1 and 4:

$$\begin{aligned} \mathrm{P}(y_i = c_j|d_i;\hat{\theta}) &= \frac{\mathrm{P}(c_j|\hat{\theta})\mathrm{P}(d_i|c_j;\hat{\theta})}{\mathrm{P}(d_i|\hat{\theta})} \\ &= \frac{\mathrm{P}(c_j|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_j;\hat{\theta})}{\sum_{r=1}^{|\mathcal{C}|} \mathrm{P}(c_r|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_r;\hat{\theta})}. \end{aligned} \tag{7}$$

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

4.4. Discussion

Note that all 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 violated in real-world text data. Documents are often mixtures of multiple topics. Words within a document are not independent of each other—grammar and topicality make this so.

Despite these violations, empirically the Naive Bayes classifier does a good job of classifying text documents (Lewis & Ringuette, 1994; Craven et al., 1998; Yang & Pederson, 1997; Joachims, 1997; McCallum, Rosenfeld, Mitchell, & Ng, 1998). This observation is explained in part by the fact that classification estimation is only a function of the sign (in binary classification) of the function estimation (Domingos & Pazzani, 1997; Friedman, 1997). The word independence assumption causes naive Bayes to give extreme (almost 0 or 1) class probability estimates. However, these estimates can still be poor while classification accuracy remains high.

The above formulation of naive Bayes uses a generative model that accounts for the number of times a word appears in a document. It is a multinomial (or in language modeling terms, "unigram") model, where the classifier is a mixture of multinomials (McCallum & Nigam, 1998). This formulation has been used by numerous practitioners of naive Bayes text classification (Lewis & Gale, 1994; Joachims, 1997; Li & Yamanishi, 1997; Mitchell, 1997; McCallum et al., 1998; Lewis, 1998). However, there is another formulation of naive Bayes text classification that instead uses a generative model and document representation in which each word in the vocabulary is a binary feature, and is modeled by a mixture of multi-variate Bernoullis (Robertson & Sparck-Jones, 1976; Lewis, 1992; Larkey & Croft, 1996; Koller & Sahami, 1997). Empirical comparisons show that the multinomial formulation yields classifiers with consistently higher accuracy (McCallum & Nigam, 1998).

5. Incorporating Unlabeled Data with EM

We now proceed to the main topic of this paper: how unlabeled data can be used to improve a text classifier. 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 the parameter estimates.

EM is a class of iterative algorithms for maximum likelihood or maximum a posteriori estimation in problems with incomplete data (Dempster et al., 1977). In our case, the unlabeled data are considered incomplete because they come without class labels.

Applying EM to naive Bayes is quite straightforward. First, the naive Bayes parameters, $\hat{\theta}$, are estimated from just the labeled documents. Then, the classifier is used to assign probabilistically-weighted class labels to each unlabeled document by calculating expectations of the missing class labels, $\mathrm{P}(c_j|d_i;\hat{\theta})$. Next, new classifier parameters, $\hat{\theta}$, are estimated using all the documents—both the originally and newly labeled. These last two steps are iterated until $\hat{\theta}$ does not change. As shown by Dempster et al. (1977), at each iteration, this process is guaranteed to find model parameters that have equal or higher likelihood than at the previous iteration.

This section describes EM and our extensions within the probabilistic framework of naive Bayes text classification.

5.1. Basic EM

We are given a set of training documents $\mathcal{D}$ and the task is to build a classifier in the form of 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 $\mathcal{D} = \mathcal{D}^l \cup \mathcal{D}^u$.

As in Section 4.2, learning a classifier is approached as calculating a maximum a posteriori estimate of θ, *i.e.* $\arg\max_\theta \mathrm{P}(\theta)\mathrm{P}(\mathcal{D}|\theta)$. Consider the second term of the maximization, 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

- **Inputs:** Collections $\mathcal{D}^l$ of labeled documents and $\mathcal{D}^u$ of unlabeled documents.
- Build an initial naive Bayes classifier, $\hat{\theta}$, from the labeled documents, $\mathcal{D}^l$, only. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$ (see Equations 5 and 6).
- Loop while classifier parameters improve, as measured by the change in $l_c(\theta|\mathcal{D};\mathbf{z})$ (the complete log probability of the labeled and unlabeled data, and the prior) (see Equation 10):
  - **(E-step)** Use the current classifier, $\hat{\theta}$, to estimate component membership of each unlabeled document, *i.e.*, the probability that each mixture component (and class) generated each document, $\mathrm{P}(c_j|d_i;\hat{\theta})$ (see Equation 7).
  - **(M-step)** Re-estimate the classifier, $\hat{\theta}$, given the estimated component membership of each document. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$ (see Equations 5 and 6).
- **Output:** A classifier, $\hat{\theta}$, that takes an unlabeled document and predicts a class label.

Table 1. The basic EM algorithm described in Section 5.1.

independent of the others, given the model. For the unlabeled data, the probability of an individual document is a sum of total probability over all the classes, as in Equation 1. For the labeled data, the generating component is already given by labels y_i, and we do not need to refer to all mixture components—just the one corresponding to the class. Thus, the probability of all the data is:

$$\begin{aligned}\mathrm{P}(\mathcal{D}|\theta) \;=\; & \prod_{d_i \in \mathcal{D}^u} \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta) \\ & \times \prod_{d_i \in \mathcal{D}^l} \mathrm{P}(y_i = c_j|\theta)\mathrm{P}(d_i|y_i = c_j;\theta). \end{aligned} \tag{8}$$

Instead of trying to maximize $\mathrm{P}(\theta|\mathcal{D})$ directly we work with $\log(\mathrm{P}(\theta|\mathcal{D}))$ instead, as a step towards making maximization (by solving the system of partial derivatives) tractable. Let $l(\theta|\mathcal{D}) \equiv \log(\mathrm{P}(\theta)\mathrm{P}(\mathcal{D}|\theta))$. Then, using Equation 8, we write

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

Notice that this equation contains a log of sums for the unlabeled data, which makes a maximization by partial derivatives computationally intractable. Consider, though, that if we had access to the class labels of all the documents—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 the *complete* log likelihood of the parameters, $l_c(\theta|\mathcal{D},\mathbf{z})$, without a log of sums, because only one term inside the sum would be non-zero.

$$l_c(\theta|\mathcal{D};\mathbf{z}) = \log(\mathrm{P}(\theta)) + \sum_{d_i \in \mathcal{D}} \sum_{j=1}^{|\mathcal{C}|} z_{ij} \log\left(\mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta)\right) \tag{10}$$

If we replace z_{ij} by its expected value according to the current model, then Equation 10 bounds from below the incomplete log likelihood from Equation 9. This can be shown by an application of Jensen's inequality (*e.g.* $E[\log(X)] \geq \log(E[X])$). As a result one can find a locally maximum $\hat{\theta}$ by a hill climbing procedure. This was formalized as the *Expectation-Maximization* (EM) algorithm by Dempster et al. (1977).

The iterative hill climbing procedure alternately recomputes the expected value of $\mathbf{z}$ and the maximum a posteriori parameters given the expected value of $\mathbf{z}$, $E[\mathbf{z}]$. Note that for the labeled documents $\mathbf{z}_i$ is already known. It must, however, be estimated for the unlabeled documents. Let $\hat{\mathbf{z}}^{(k)}$ and $\hat{\theta}^{(k)}$ denote the estimates for $\mathbf{z}$ and θ at iteration k. Then, the algorithm finds a local maximum of $l(\theta|\mathcal{D})$ by iterating the following two steps:

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

In practice, the E-step corresponds to calculating probabilistic labels $\mathrm{P}(c_j|d_i;\hat{\theta})$ for the unlabeled documents by using the current estimate of the parameters, $\hat{\theta}$, and Equation 7. The M-step, maximizing the complete likelihood equation, corresponds to calculating a new maximum a posteriori estimate for the parameters, $\hat{\theta}$, using the current estimates for $\mathrm{P}(c_j|d_i;\hat{\theta})$, and Equations 5 and 6.

Our iteration process is initialized with a "priming" M-step, in which only the labeled documents are used to estimate the classifier parameters, $\hat{\theta}$, as in Equations 5 and 6. Then the cycle begins with an E-step that uses this classifier to probabilistically label the unlabeled documents for the first time.

The algorithm iterates over the E- and M-steps until it converges to a point where $\hat{\theta}$ does not change from one iteration to the next. Algorithmically, we determine that convergence has occurred by observing a below-threshold change in the log-probability of the parameters (Equation 10), which is the height of the surface on which EM is hill-climbing.

Table 1 gives an outline of the basic EM algorithm from this section.

5.2. Discussion

In summary, EM finds a $\hat{\theta}$ that locally maximizes the likelihood of its parameters given *all* the data—both the labeled and the unlabeled. It provides a method whereby unlabeled data can augment limited labeled data and contribute to parameter estimation. An interesting empirical question is whether these higher likelihood parameter estimates will improve classification accuracy. Section 4.4 discusses the fact that naive Bayes usually performs classification well despite violations of its assumptions. Will EM also have this property?

Note that the justifications for this approach depend on the assumptions stated in Section 3, namely, that the data is produced by a mixture model, and that there is a one-to-one correspondence between mixture components and classes. When these assumptions do not hold—as certainly is the case in real-world textual data—the benefits of unlabeled data are less clear.

Our experimental results in Section 6 show that this method can indeed dramatically improve the accuracy of a document classifier, especially when there are only a few labeled documents. But on some data sets, when there are a lot of labeled and a lot of unlabeled documents, this is not the case. In several experiments, the incorporation of unlabeled data decreases, rather than increases, classification accuracy.

Next we describe changes to the basic EM algorithm described above that aim to address performance degradation due to violated assumptions.

5.3. Augmented EM

This section describes two extensions to the basic EM algorithm described above. The extensions help improve classification accuracy even in the face of somewhat violated assumptions of the generative model. In the first we add a new parameter to modulate the degree to which EM weights the unlabeled data; in the second we augment the model to relax one of the assumptions about the generative model.

5.3.1. Weighting the unlabeled data. As described in the introduction, a common scenario is that few labeled documents are on hand, but many orders of magnitude more unlabeled documents are readily available. In this case, the great majority of the data determining EM's parameter estimates comes from the unlabeled set. In these circumstances, we can think of EM as almost entirely performing unsupervised clustering, since the model is mostly positioning the mixture components to maximize the likelihood of the unlabeled documents. The number of labeled data is so small in comparison to the unlabeled, that the only significant effect of the labeled data is to initialize the classifier parameters (*i.e.* determining EM's starting point for hill climbing), and to identify each component with a class label.

When the two mixture model assumptions are true, and the natural clusters of the data are in correspondence with the class labels, then unsupervised clustering with many unlabeled documents will result in mixture components that are useful for classification (*c.f.* Section 2, where infinite amounts of unlabeled data are sufficient to learn the parameters of the mixture components). However, when the mixture model assumptions are not true, the natural clustering of the unlabeled data may produce mixture components that are not in correspondence with the class labels, and are therefore detrimental to classification accuracy. This effect is particularly apparent when the number of labeled documents is already large enough to obtain reasonably good parameter estimates for the classifier, yet the orders of magnitude more unlabeled documents still overwhelm parameter estimation and thus badly skew the estimates.

This subsection describes a method whereby the influence of the unlabeled data is modulated in order to control the extent to which EM performs unsupervised clustering. We introduce a new parameter λ, $0 \leq \lambda \leq 1$, into the likelihood equation which decreases the contribution of the unlabeled documents to parameter estimation. We term the resulting method EM-λ. Instead of using EM to maximize Equation 10, we instead maximize:

$$\begin{aligned} l_c(\theta|\mathcal{D};\mathbf{z}) &= \log(\mathrm{P}(\theta)) + \sum_{d_i \in \mathcal{D}^l} \sum_{j=1}^{|\mathcal{C}|} z_{ij} \log\left(\mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta)\right) \\ &\quad + \lambda \left(\sum_{d_i \in \mathcal{D}^u} \sum_{j=1}^{|\mathcal{C}|} z_{ij} \log\left(\mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta)\right) \right). \end{aligned} \quad (11)$$

Notice that when λ is close to zero, the unlabeled documents will have little influence on the shape of EM's hill-climbing surface. When $\lambda = 1$, each unlabeled document will be weighted the same as a labeled document, and the algorithm is the same as the original EM previously described.

When iterating to maximize Equation 11, the E-step is performed exactly as before. The M-step is different, however, and entails the following substitutes for Equations 5 and 6. First define $\Lambda(i)$ to be the weighting factor λ whenever d_i in the unlabeled set, and to be 1 whenever d_i is in the labeled set:

$$\Lambda(i) = \begin{cases} \lambda & \text{if } d_i \in \mathcal{D}^u \\ 1 & \text{if } d_i \in \mathcal{D}^l. \end{cases} \quad (12)$$

Then the new estimate $\hat{\theta}_{w_t|c_j}$ is again a ratio of word counts, but where the counts of the unlabeled documents are decreased by a factor of λ:

$$\hat{\theta}_{w_t|c_j} \equiv \mathrm{P}(w_t|c_j;\hat{\theta}) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} \Lambda(i) N(w_t, d_i) \mathrm{P}(y_i = c_j|d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|\mathcal{D}|} \Lambda(i) N(w_s, d_i) \mathrm{P}(y_i = c_j|d_i)}. \quad (13)$$

Class prior probabilities, $\hat{\theta}_{c_j}$, are modified similarly:

$$\hat{\theta}_{c_j} \equiv \mathrm{P}(c_j|\hat{\theta}) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} \Lambda(i) \mathrm{P}(y_i = c_j|d_i)}{|\mathcal{C}| + |\mathcal{D}^l| + \lambda|\mathcal{D}^u|}. \quad (14)$$

These equations can be derived by again solving the system of partial derivatives using Lagrange multipliers to enforce the constraint that probabilities sum to one.

In this paper we select the value of λ that maximizes the leave-one-out cross-validation classification accuracy of the labeled training data. Experimental results with this technique are described in Section 6.3. As shown there, setting λ to some value between 0 and 1 can result in classification accuracy higher than either $\lambda = 0$ or $\lambda = 1$, indicating that there can be value in the unlabeled data even when its natural clustering would result in poor classification.

5.3.2. Multiple mixture components per class. The EM-λ technique described above addresses violated mixture model assumptions by reducing the effect of those violated assumptions on parameter estimation. An alternative approach is to attack the problem head-on by removing or weakening a restrictive assumption. This subsection takes exactly this approach by relaxing the assumption of a one-to-one correspondence between mixture components and classes. We replace it with a less restrictive assumption: a *many*-to-one correspondence between mixture components and classes.

For textual data, this corresponds to saying that a class may be comprised of several different sub-topics, each best captured with a different word distribution. Furthermore, using multiple mixture components per class can capture some dependencies between words. For example, consider a sports class consisting of documents about both hockey and baseball. In these documents, the words "ice" and "puck" are likely to co-occur, and the words "bat" and "base" are likely to co-occur. However, these dependencies cannot be captured by a single multinomial distribution over words in the sports class. On the other hand, with multiple mixture components per class, one multinomial can cover the hockey sub-topic, and another the baseball sub-topic—thus more accurately capturing the co-occurrence patterns of the above four words.

For some or all of the classes we now allow multiple multinomial mixture components. Note that as a result, there are now "missing values" for the labeled as well as the unlabeled documents—it is unknown which mixture component, among those covering the given label, is responsible for generating a particular labeled document. Parameter estimation will still be performed with EM except that, for each labeled document, we must now estimate which mixture component the document came from.

Let us introduce the following notation for separating mixture components from classes. Instead of using c_j to denote both a class and its corresponding mixture component, we will now write t_a for the ath class ("topic"), and c_j will continue to denote the jth mixture component. We write $\mathrm{P}(t_a|c_j;\hat{\theta}) \in \{0,1\}$ for the predetermined, deterministic, many-to-one mapping between mixture components and classes.

Parameter estimation is again done with EM. The M-step is the same as basic EM, building maximum a posteriori parameter estimates for the multinomial of each component. In the E-step, unlabeled documents are treated as before, calculating probabilistically-weighted mixture component membership, $\mathrm{P}(c_j|d_i;\hat{\theta})$. For labeled documents, the previous $\mathrm{P}(c_j|d_i;\hat{\theta}) \in \{0,1\}$ that was considered to be fixed by the class label is now allowed to vary between 0 and 1 for mixture components assigned to that document's class. Thus, the algorithm also calculates probabilistically-weighted mixture component membership for the labeled documents. Note, however, that all $\mathrm{P}(c_j|d_i;\hat{\theta})$, for which $\mathrm{P}(y_i = t_a|c_j;\hat{\theta})$ is zero, are clamped at zero, and the rest are normalized to sum to one.

Multiple mixture components for the same class are initialized by randomly spreading the labeled training data across the mixture components matching the appropriate class label. That is, components are initialized by performing a ran-

- **Inputs:** Collections $\mathcal{D}^l$ of labeled documents and $\mathcal{D}^u$ of unlabeled documents.
- **[Weighted only]:** Set the discount factor of the unlabeled data, λ, by cross-validation (see Sections 6.1 and 6.3).
- **[Multiple only]:** Set the number of mixture components per class by cross-validation (see Sections 6.1 and 6.4).
- **[Multiple only]:** For each labeled document, randomly assign $\mathrm{P}(c_j|d_i;\hat{\theta})$ for mixture components that correspond to the document's class label, to initialize each mixture component.
- Build an initial naive Bayes classifier, $\hat{\theta}$, from the labeled documents only. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$ (see Equations 5 and 6).
- Loop while classifier parameters improve ($0.05 < \Delta l_c(\theta|\mathcal{D};\mathbf{z})$, the change in complete log probability of the labeled and unlabeled data, and the prior) (see Equation 10):
    - **(E-step)** Use the current classifier, $\hat{\theta}$, to estimate the component membership of each document, *i.e.* the probability that each mixture component generated each document, $\mathrm{P}(c_j|d_i;\hat{\theta})$ (see Equation 7).

      [Multiple only]: Restrict the membership probability estimates of labeled documents to be zero for components associated with other classes, and renormalize.
    - **(M-step)** Re-estimate the classifier, $\hat{\theta}$, given the estimated component membership of each document. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$ (see Equations 5 and 6).

      [Weighted only]: When counting events for parameter estimation, word and document counts from unlabeled documents are reduced by a factor λ (see Equations 13 and 14).
- **Output:** A classifier, $\hat{\theta}$, that takes an unlabeled document and predicts a class label.

Table 2. The Algorithm described in this paper, and used to generate the experimental results in Section 6. The algorithm enhancements for EM-λ that vary the contribution of the unlabeled data (Section 5.3.1) are indicated by **[Weighted only]**. The optional use of multiple mixture components per class (Section 5.3.2) is indicated by **[Multiple only]**. Unmarked paragraphs are common to all variations of the algorithm.

domized E-step in which $\mathrm{P}(c_j|d_i;\hat{\theta})$ is sampled from a uniform distribution over mixture components for which $\mathrm{P}(t_a = y_i|c_j;\hat{\theta})$ is one.

When there are multiple mixture components per class, classification becomes a matter of probabilistically "classifying" documents into the mixture components, and then summing the mixture component probabilities into class probabilities:

$$\mathrm{P}(t_a|d_i;\hat{\theta}) = \sum_{c_j} \mathrm{P}(t_a|c_j;\hat{\theta}) \frac{\mathrm{P}(c_j|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_j;\hat{\theta})}{\sum_{r=1}^{|\mathcal{C}|} \mathrm{P}(c_r|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_r;\hat{\theta})}. \tag{15}$$

In this paper, we select the number of mixture components per class by cross-validation. Table 2 gives an outline of the EM algorithm with the extensions of this and the previous section.

Experimental results from this technique are described in Section 6.4. As shown there, when the data are not naturally modeled by a single component per class, the use of unlabeled data with EM degrades performance. However, when multiple mixture components per class are used, performance with unlabeled data and EM is superior to naive Bayes.

6. Experimental Results

In this section, we provide empirical evidence that combining labeled and unlabeled training documents using EM outperforms traditional naive Bayes, which trains on labeled documents alone. We present experimental results with three different text corpora: UseNet news articles (20 Newsgroups), web pages (WebKB), and newswire articles (Reuters).[5]

Results show that improvements in accuracy due to unlabeled data are often dramatic, especially when the number of labeled training documents is low. For example, on the 20 Newsgroups data set, classification error is reduced by 30% when trained with 300 labeled and 10000 unlabeled documents.

On certain data sets, however, (and especially when the number of labeled documents is high), the incorporation of unlabeled data with the basic EM scheme may reduce rather than increase accuracy. We show that the application of the EM extensions described in the previous section increases performance beyond that of naive Bayes.

6.1. Datasets and Protocol

The 20 Newsgroups data set (Joachims, 1997; McCallum et al., 1998; Mitchell, 1997), collected by Ken Lang, consists of 20017 articles divided almost evenly among 20 different UseNet discussion groups. The task is to classify an article into the one newsgroup (of twenty) to which it was posted. Many of the categories fall into confusable clusters; for example, five of them are `comp.*` discussion groups, and three of them discuss religion. When words from a stoplist of common short words are removed, there are 62258 unique words that occur more than once; other feature selection is not used. 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. The word counts of each document are scaled such that each document has constant length, with potentially fractional word counts. Our preliminary experiments with 20 Newsgroups indicated that naive Bayes classification was better with this word count normalization.

The 20 Newsgroups data set was collected from UseNet postings over a period of several months in 1993. Naturally, the data have time dependencies—articles nearby in time are more likely to be from the same thread, and because of occasional quotations, may contain many of the same words. In practical use, a classifier for this data set would be asked to classify future articles after being trained on articles from the past. To preserve this scenario, we create a test set of 4000 documents by selecting by posting date the last 20% of the articles from each newsgroup. An unlabeled set is formed by randomly selecting 10000 documents from those remaining. Labeled training sets are formed by partitioning the remaining 6000 documents into non-overlapping sets. The sets are created with equal numbers of documents per class. For experiments with different labeled set sizes, we create up to ten sets per size; obviously, fewer sets are possible for experiments with labeled sets containing more than 600 documents. The use of each non-overlapping training set comprises a new trial of the given experiment. Results are reported as averages over all trials of the experiment.

The WebKB data set (Craven et al., 1998) contains 8145 web pages gathered from university computer science departments. The collection includes the entirety of four departments, and additionally, an assortment of pages from other universities. The pages are divided into seven categories: student, faculty, staff, course, project, department and other. In this paper, we use the four most populous non-other categories: student, faculty, course and project—all together containing 4199 pages. The task is to classify a web page into the appropriate one of the four categories. For consistency with previous studies with this data set (Craven et al., 1998), when tokenizing the WebKB data, numbers were converted into a time or a phone number token, if appropriate, or otherwise a sequence-of-length-n token.

We did not use stemming or a stoplist; we found that using a stoplist actually hurt performance. For example, "my" is an excellent indicator of a student homepage and is the fourth-ranked word by information gain. We limit the vocabulary to the 300 most informative words, as measured by average mutual information with the class variable. This feature selection method is commonly used for text (Yang & Pederson, 1997; Koller & Sahami, 1997; Joachims, 1997). We selected this vocabulary size by running leave-one-out cross-validation on the training data to optimize classification accuracy.

The WebKB data set was collected as part of an effort to create a crawler that explores previously unseen computer science departments and classifies web pages into a knowledge-base ontology. To mimic the crawler's intended use, and to avoid reporting performance based on idiosyncrasies particular to a single department, we test using a leave-one-university-out approach. That is, we create four test sets, each containing all the pages from one of the four complete computer science departments. For each test set, an unlabeled set of 2500 pages is formed by randomly selecting from the remaining web pages. Non-overlapping training sets are formed by the same method as in 20 Newsgroups. Also as before, results are reported as averages over all trials that share the same number of labeled training documents.

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, 1998; Liere & Tadepalli, 1997) we build binary classifiers for each of the ten most populous classes to identify the news topic. 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.

In Reuters, classifiers for different categories perform best with widely varying vocabulary sizes (which are chosen by average mutual information with the class variable). This variance in optimal vocabulary size is unsurprising. As previously noted (Joachims, 1997), categories like "wheat" and "corn" are known for a strong correspondence between a small set of words (like their title words) and the categories, while categories like "acq" are known for more complex characteristics. The categories with narrow definitions attain best classification with small vocabularies, while those with a broader definition require a large vocabulary. The vocabulary size for each Reuters trial is selected by optimizing accuracy as measured by leave-one-out cross-validation on the labeled training set.

As with the 20 Newsgroups data set, there are time dependencies in Reuters. The standard 'ModApte' train/test split divides the articles by time, such that the later 3299 documents form the test set, and the earlier 9603 are available for training. In our experiments, 7000 documents from this training set are randomly selected to form the unlabeled set. From the remaining training documents, we randomly select up to ten non-overlapping training sets of ten positively labeled documents and 40 negatively labeled documents, as previously described for the other two data sets. We use non-uniform number of labelings across the classes because the negative class is much more frequent than the positive class in all of the binary Reuters classification tasks.

Results on Reuters are reported as precision-recall breakeven points, a standard information retrieval measure for binary classification. Accuracy is not a good performance metric here because very high accuracy can be achieved by always predicting the negative class. The task on this data set is less like classification than it is like filtering—find the few positive examples from a large sea of negative examples. Recall and precision capture the inherent duality of this task, and are defined as:

$$\text{Recall} = \frac{\text{\# of correct positive predictions}}{\text{\# of positive examples}} \tag{16}$$

$$\text{Precision} = \frac{\text{\# of correct positive predictions}}{\text{\# of positive predictions}}. \tag{17}$$

The classifier can achieve a trade-off between precision and recall by adjusting the decision boundary between the positive and negative class away from its previous default of $\mathrm{P}(c_j|d_i;\hat{\theta}) = 0.5$. The precision-recall breakeven point is defined as the precision and recall value at which the two are equal (*e.g.* Joachims, 1998).

The algorithm used for experiments with EM is described in Table 2.

In this section, when leave-one-out cross-validation is performed in conjunction with EM, we make one simplification for computational efficiency. We first run EM to convergence with all the training data, and then subtract the word counts of each labeled document in turn before testing that document. Thus, when performing cross-validation for a specific combination of parameter settings, only one run of EM is required instead of one run of EM per labeled example. Note, however, that there are still some residual effects of the held-out document.

The computational complexity of EM, however, is not prohibitive. Each iteration requires classifying the training documents (E-step), and building a new classifier (M-step). In our experiments, EM usually converges after about 10 iterations. The wall-clock time to read the document-word matrix from disk, build an EM model by iterating to convergence, and classify the test documents is less than one minute

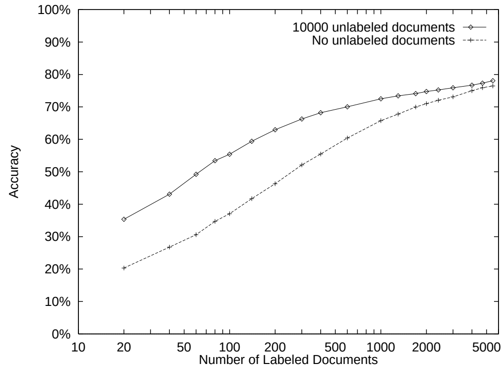

Figure 2. Classification accuracy on the 20 Newsgroups data set, both with and without 10,000 unlabeled documents. With small amounts of training data, using EM yields more accurate classifiers. With large amounts of labeled training data, accurate parameter estimates can be obtained without the use of unlabeled data, and the two methods begin to converge.

for the WebKB data set, and less than 15 minutes for 20 Newsgroups. The 20 Newsgroups data set takes longer because it has more documents and more words in the vocabulary.

6.2. EM with Unlabeled Data Increases Accuracy

We first consider the use of basic EM to incorporate information from unlabeled documents. Figure 2 shows the effect of using basic EM with unlabeled data on the 20 Newsgroups data set. The vertical axis indicates average classifier accuracy on test sets, and the horizontal axis indicates the amount of labeled training data on a log scale. 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 52% accuracy while EM achieves 66%. This represents a 30% reduction in classification error. Note that EM also performs well even with a very small number of labeled documents; with only 20 documents (a single labeled document per class), naive Bayes obtains 20%, EM 35%. As expected, when there is a lot of labeled data, and the naive Bayes learning curve is close to a plateau, having unlabeled data does not help nearly as much, because there is already enough labeled data to accurately estimate the classifier parameters. With 5500 labeled documents (275 per class), classification accuracy increases from 76% to 78%. Each of these results is statistically significant ($p < 0.05$).[6]

These results demonstrate that EM finds parameter estimates that improve classification accuracy and reduce the need for labeled training examples. For example, to reach 70% classification accuracy, naive Bayes requires 2000 labeled examples, while EM requires only 600 labeled examples to achieve the same accuracy.

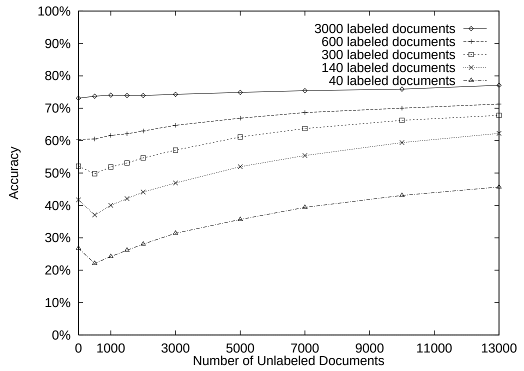

Figure 3. Classification accuracy while varying the number of unlabeled documents. The effect is shown on the 20 Newsgroups data set, with 5 different amounts of labeled documents, by varying the amount of unlabeled data on the horizontal axis. Having more unlabeled data helps. Note the dip in accuracy when a small amount of unlabeled data is added to a small amount of labeled data. We hypothesize that this is caused by extreme, almost 0 or 1, estimates of component membership, $\mathrm{P}(c_j|d_i, \hat{\theta})$, for the unlabeled documents (as caused by naive Bayes' word independence assumption).

In Figure 3 we consider the effect of varying the amount of unlabeled data. For five different quantities of labeled documents, we hold the number of labeled documents constant, and vary the number of unlabeled documents in the horizontal axis. Naturally, having more unlabeled data helps, and it helps more when there is less labeled data.

Notice that adding a small amount of unlabeled data to a small amount of labeled data actually hurts performance. We hypothesize that this occurs because the word independence assumption of naive Bayes leads to overly-confident $\mathrm{P}(c_j|d_i, \hat{\theta})$ estimates in the E-step, and the small amount of unlabeled data is distributed too sharply. (Without this bias in naive Bayes, the E-step would spread the unlabeled data more evenly across the classes.) When the number of unlabeled documents is large, however, this problem disappears because the unlabeled set provides a large enough sample to smooth out the sharp discreteness of naive Bayes' overly-confident classification.

We now move on to a different data set. To provide some intuition about why EM works, we present a detailed trace of one example from the WebKB data set. Table 3 shows the evolution of the classifier over the course of two EM iterations. Each column shows the ordered list of words that the model indicates are most "predictive" of the course class. Words are judged to be "predictive" using a weighted log likelihood ratio.[7] The symbol D indicates an arbitrary digit. At Iteration 0, the parameters are 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

Table 3. 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 trial. By the second iteration of EM, many common course-related words appear. The symbol D indicates an arbitrary digit.

Iteration 0	Iteration 1	Iteration 2
intelligence	DD	D
DD	D	DD
artificial	lecture	lecture
understanding	cc	cc
DDw	$D^{\star}$	$DD{:}DD$
dist	$DD{:}DD$	due
identical	handout	$D^{\star}$
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	DDam	hw
cognitive	yurttas	exam
logic	homework	problem
proving	kfoury	DDam
prolog	sec	postscript
knowledge	postscript	solution
human	exam	quiz
representation	solution	chapter
field	assaf	ascii

indicative of courses. The classifier corresponding to the first column achieves 50% accuracy; when EM converges, the classifier achieves 71% accuracy.

6.3. *Varying the Weight of the Unlabeled Data*

When graphing performance on this data set, we see that the incorporation of unlabeled data can also decrease, rather than increase, classification accuracy. The graph in Figure 4 shows the performance of basic EM (with 2500 unlabeled documents) on WebKB. Again, EM improves accuracy significantly when the amount of labeled data is small. When there are four labeled documents (one per class), traditional naive Bayes attains 40% accuracy, while EM reaches 55%. When there is a lot of labeled data, however, EM hurts performance slightly. With 240 labeled documents, naive Bayes obtains 81% accuracy, while EM does worse at 79%. Both of these differences in performance are statistically significant ($p < 0.05$), for three and two of the university test sets, respectively.

As discussed in Section 5.3.1, we hypothesize that EM hurts performance here because the data do not fit the assumptions of the generative model—that is, the mixture components that best explain the unlabeled data are not in precise correspondence with the class labels. It is not surprising that the unlabeled data can throw off parameter estimation when one considers that the number of unlabeled documents is much greater than the number of labeled documents (*e.g.* 2500 versus 240), and thus, even at the points in Figure 4 with the largest amounts of labeled

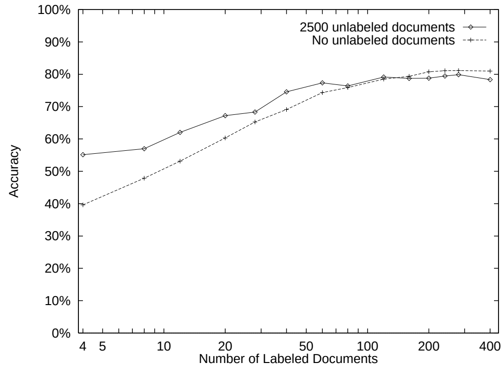

Figure 4. Classification accuracy on the WebKB data set, both with and without 2500 unlabeled documents. When there are small numbers of labeled documents, EM improves accuracy. When there are many labeled documents, however, EM degrades performance slightly—indicating a misfit between the data and the assumed generative model.

data, the great majority of the probability mass used in the M-step to estimate the classifier parameters actually comes from the unlabeled data.

To remedy this dip in performance, we use EM-λ to reduce the weight of the unlabeled data by varying λ in Equations 13 and 14. Figure 5 plots classification accuracy while varying λ to achieve the relative weighting indicated in the horizontal axis, and does so for three different amounts of labeled training data. The bottom curve is obtained using 40 labeled documents—a vertical slice in Figure 4 at a point where EM with unlabeled data gives higher accuracy than naive Bayes. Here, the best weighting of the unlabeled data is high, indicating that classification can be improved by augmenting the sparse labeled data with heavy reliance on the unlabeled data. The middle curve is obtained using 80 labeled documents—a slice near the point where EM and naive Bayes performance cross. Here, the best weighting is in the middle, indicating that EM-λ performs better than either naive Bayes or basic EM. The top curve is obtained using 200 labeled documents—a slice where unweighted EM performance is lower than traditional naive Bayes. Less weight should be given to the unlabeled data at this point.

Note the inverse relationship between the labeled data set size and the best weighting factor—the smaller labeled data set, the larger the best weighting of the unlabeled data. This trend holds across all amounts of labeled data. Intuitively, when EM has very little labeled training data, parameter estimation is so desperate for guidance that EM with all the unlabeled data helps in spite of the somewhat violated assumptions. However, when there is enough labeled training data to sufficiently estimate the parameters, less weight should be given to the unlabeled data. Finally, note that the best-performing values of λ are somewhere between the extremes, remembering that the right-most point corresponds to EM with the weighting used to generate Figure 4, and the left-most to regular naive Bayes. Paired t-tests across the trials of all the test universities show that the best-performing points on these curves are statistically significantly higher than either

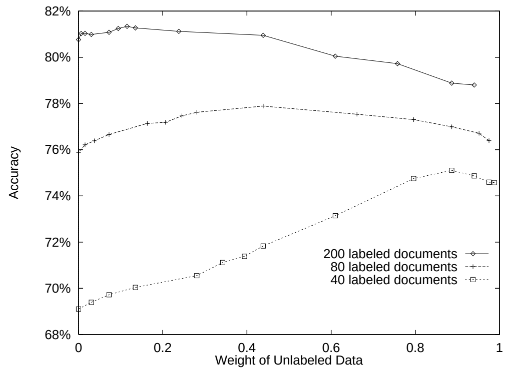

Figure 5. The effects of varying λ, the weighting factor on the unlabeled data in EM-λ. These three curves from the WebKB data set correspond to three different amounts of labeled data. When there is less labeled data, accuracy is highest when more weight is given to the unlabeled data. When the amount of labeled data is large, accurate parameter estimates are attainable from the labeled data alone, and the unlabeled data should receive less weight. With moderate amounts of labeled data, accuracy is better in the middle than at either extreme. Note the magnified vertical scale.

end point, except for the difference between the maxima and basic EM with 40 labeled documents ($p < 0.05$).

In practice the value of the tuning parameter λ can be selected by cross-validation. In our experiments we select λ by leave-one-out cross-validation on the labeled training set for each trial, as discussed in Section 6.1. Figure 6 shows the accuracy for the best possible λ, and the accuracy when selecting λ via cross-validation. Basic EM and naive Bayes accuracies are also shown for comparison. When λ is perfectly selected, its accuracy dominates the basic EM and naive Bayes curves. Cross-validation selects λ's that, for small amounts of labeled documents, perform about as well as EM. For large amounts of labeled documents, cross-validation selects λ's that do not suffer from the degraded performance seen in basic EM, and also performs at least as well as naive Bayes. For example, at the 240 document level seen before, the λ picked by cross-validation gives only 5% of the weight to the unlabeled data, instead of the 91% given by basic EM. Doing so provides an accuracy of 82%, compared to 81% for naive Bayes and 79% for basic EM. This is not statistically significantly different from naive Bayes, and is statistically significantly higher than basic EM for two of the four test sets (both $p < 0.05$). These results indicate that we can automatically avoid EM's degradation in accuracy at large training set sizes and still preserve the benefits of EM seen with small labeled training sets.

These results also indicate that when the training set size is very small improved methods of selecting λ could significantly increase the practical performance of EM-λ even further. Note that in these cases, cross-validation has only a few documents with which to choose λ. The end of Section 6.4 suggests some methods that may perform better than cross-validation.

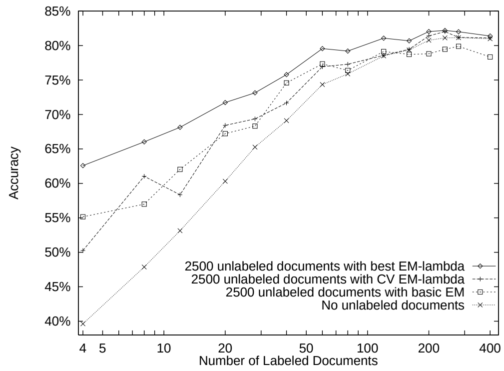

Figure 6. Classification accuracy on the WebKB data set, with modulation of the unlabeled data by the weighting factor λ. The top curve shows accuracy when using the best value of λ. In the second curve, λ is chosen by cross-validation. With small amounts of labeled data, the results are similar to basic EM; with large amounts of labeled data, the results are more accurate than basic EM. Thanks to the weighting factor, large amounts of unlabeled data no longer degrades accuracy, as it did in Figure 4, and yet the algorithm retains the large improvements with small amounts of labeled data. Note the magnified vertical axis to facilitate the comparisons.

6.4. Multiple Mixture Components per Class

Faced with data that do not fit the assumptions of our model, the λ-tuning approach described above addresses this problem by allowing the model to incrementally ignore the unlabeled data. Another, more direct approach, described in Section 5.3.2, is to change the model so that it more naturally fits the data. Flexibility can be added to the mapping between mixture components and class labels by allowing multiple mixture components per class. We expect this to improve performance when data for each class is, in fact, multi-modal.

With an eye towards testing this hypothesis, we apply 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 expect this task to strongly violate the assumption that all the data for the negative class are generated by a single mixture component. For this reason, we model the positive class with a single mixture component and the negative class with between one and forty mixture components, both with and without unlabeled data.

Table 4 contains a summary of results on the test set for modeling the negative class with multiple mixture components. The NB1 column shows precision-recall breakeven points from standard naive Bayes (with just the labeled data), that models the negative class with a single mixture component. The NB* column shows the results of modeling the negative class with multiple mixture components (again using just the labeled data). In the NB* column, the number of components has been selected to optimize the best precision-recall breakeven point. The median number of components selected across trials is indicated in parenthesis beside the

Table 4. Precision-recall breakeven points showing performance of binary classifiers on Reuters with traditional naive Bayes (NB1), multiple mixture components using just labeled data (NB*), basic EM (EM1) with labeled and unlabeled data, and multiple mixture components EM with labeled and unlabeled data (EM*). For NB* and EM*, the number of components is selected optimally for each trial, and the median number of components across the trials used for the negative class is shown in parentheses. Note that the multi-component model is more natural for Reuters, where the negative class consists of many topics. Using both unlabeled data and multiple mixture components per class increases performance over either alone, and over naive Bayes.

Category	NB1	NB*	EM1	EM*	EM* vs NB1	EM* vs NB*
acq	69.4	74.3 (4)	70.7	83.9 (10)	+14.5	+9.6
corn	44.3	47.8 (3)	44.6	52.8 (5)	+8.5	+5.0
crude	65.2	68.3 (2)	68.2	75.4 (8)	+10.2	+7.1
earn	91.1	91.6 (1)	89.2	89.2 (1)	-1.9	-2.4
grain	65.7	66.6 (2)	67.0	72.3 (8)	+6.3	+5.7
interest	44.4	54.9 (5)	36.8	52.3 (5)	+7.9	-2.6
money-fx	49.4	55.3 (15)	40.3	56.9 (10)	+7.5	+1.6
ship	44.3	51.2 (4)	34.1	52.5 (7)	+8.2	+1.3
trade	57.7	61.3 (3)	56.1	61.8 (3)	+4.1	+0.5
wheat	56.0	67.4 (10)	52.9	67.8 (10)	+11.8	+0.4

breakeven point. Note that even before we consider the effect of unlabeled data, using this more complex representation on this data improves performance over traditional naive Bayes.

The column labeled EM1 shows results with basic EM (*i.e.* with a single negative component). Notice that here performance is often worse than naive Bayes (NB1). We hypothesize that, because the negative class is truly multi-modal, fitting a single naive Bayes class with EM to the data does not accurately capture the negative class word distribution.

The column labeled EM* shows results of EM with multiple mixture components, again selecting the best number of components. Here performance is better than both NB1 (traditional naive Bayes) and NB* (naive Bayes with multiple mixture components per class). This increase, measured over all trials of Reuters, is statistically significant ($p < 0.05$). This indicates that while the use of multiple mixture components increases performance over traditional naive Bayes, the combination of unlabeled data and multiple mixture components increases performance even more.

Furthermore, it is interesting to note that, on average, EM* uses more mixture components than NB*—suggesting that the addition of unlabeled data reduces variance and supports the use of a more expressive model.

Tables 5 and 6 show the complete results for experiments using multiple mixture components with and without unlabeled data, respectively. Note that in general, using too many or too few mixture components hurts performance. With too few components, our assumptions are overly restrictive. With too many components, there are more parameters to estimate from the same amount of data. Table 7 shows the same results as Table 4, but for classification accuracy, and not precision-recall breakeven. The general trends for accuracy are the same as for precision-recall. However, for accuracy, the optimal number of mixture components for the negative class is greater than for precision-recall, because by its nature precision-

Table 5. Performance of EM using different numbers of mixture components for the negative class and 7000 unlabeled documents. Precision-recall breakeven points are shown for experiments using between one and forty mixture components. Note that using too few or too many mixture components results in poor performance.

Category	EM1	EM3	EM5	EM10	EM20	EM40
acq	70.7	75.0	72.5	77.1	68.7	57.5
corn	44.6	45.3	45.3	46.7	41.8	19.1
crude	68.2	72.1	70.9	71.6	64.2	44.0
earn	89.2	88.3	88.5	86.5	87.4	87.2
grain	67.0	68.8	70.3	68.0	58.5	41.3
interest	36.8	43.5	47.1	49.9	34.8	25.8
money-fx	40.3	48.4	53.4	54.3	51.4	40.1
ship	34.1	41.5	42.3	36.1	21.0	5.4
trade	56.1	54.4	55.8	53.4	35.8	27.5
wheat	52.9	56.0	55.5	60.8	60.8	43.4

Table 6. Performance of EM using different numbers of mixture components for the negative class, but with no unlabeled data. Precision-recall breakeven points are shown for experiments using between one and forty mixture components.

Category	NB1	NB3	NB5	NB10	NB20	NB40
acq	69.4	69.4	65.8	68.0	64.6	68.8
corn	44.3	44.3	46.0	41.8	41.1	38.9
crude	65.2	60.2	63.1	64.4	65.8	61.8
earn	91.1	90.9	90.5	90.5	90.5	90.4
grain	65.7	63.9	56.7	60.3	56.2	57.5
interest	44.4	48.8	52.6	48.9	47.2	47.6
money-fx	49.4	48.1	47.5	47.1	48.8	50.4
ship	44.3	42.7	47.1	46.0	43.6	45.6
trade	57.7	57.5	51.9	53.2	52.3	58.1
wheat	56.0	59.7	55.7	65.0	63.2	56.0

Table 7. Classification accuracy on Reuters with traditional naive Bayes (NB1), multiple mixture components using just labeled data (NB*), basic EM (EM1) with labeled and unlabeled data, and multiple mixture components EM with labeled and unlabeled data (EM*), as in Table 4.

Category	NB1	NB*	EM1	EM*	EM* vs NB1	EM* vs NB*
acq	86.9	88.0 (4)	81.3	93.1 (10)	+6.2	+5.1
corn	94.6	96.0 (10)	93.2	97.2 (40)	+2.6	+1.2
crude	94.3	95.7 (13)	94.9	96.3 (10)	+2.0	+0.6
earn	94.9	95.9 (5)	95.2	95.7 (10)	+0.8	-0.2
grain	94.1	96.2 (3)	93.6	96.9 (20)	+2.8	+0.7
interest	91.8	95.3 (5)	87.6	95.8 (10)	+4.0	+0.5
money-fx	93.0	94.1 (5)	90.4	95.0 (15)	+2.0	+0.9
ship	94.9	96.3 (3)	94.1	95.9 (3)	+1.0	-0.4
trade	91.8	94.3 (5)	90.2	95.0 (20)	+3.2	+0.7
wheat	94.0	96.2 (4)	94.5	97.8 (40)	+3.8	+1.6

Table 8. Performance of using multiple mixture components when the number of components is selected via cross-validation (EM*CV) compared to optimal selection (EM*) and straight naive Bayes (NB1). Note that cross-validation usually selects too few components.

Category	NB1	EM*	EM*CV	EM*CV vs NB1
acq	69.4	83.9 (10)	75.6 (1)	+6.2
corn	44.3	52.8 (5)	47.1 (3)	+2.8
crude	65.2	75.4 (8)	68.3 (1)	+3.1
earn	91.1	89.2 (1)	87.1 (1)	-4.0
grain	65.7	72.3 (8)	67.2 (1)	+1.5
interest	44.4	52.3 (5)	42.6 (3)	-1.8
money-fx	49.4	56.9 (10)	47.4 (2)	-2.0
ship	44.3	52.5 (7)	41.3 (2)	-3.0
trade	57.7	61.8 (3)	57.3 (1)	-0.4
wheat	56.0	67.8 (10)	56.9 (1)	+0.9

recall focuses more on modeling the positive class, where accuracy focuses more on modeling the negative class, because it is much more frequent. By allowing more mixture components for the negative class, a more accurate model is achieved.

One obvious question is how to select the best number of mixture components without having access to the test set labels. As with selection of the weighting factor, λ, we use leave-one-out cross-validation, with the computational short-cut that entails running EM only once (as described at the end of Section 6.1).

Results from this technique (EM*CV), compared to naive Bayes (NB1) and the best EM (EM*), are shown in Table 8. Note that cross-validation does not perfectly select the number of components that perform best on the test set. The results consistently show that selection by cross-validation chooses a smaller number of components than is best. By using the cross-validation with the computational short-cut, we bias the model towards the held-out document, which, we hypothesize, favors the use of fewer components. The computationally expensive, but complete, cross-validation should perform better.

Other model selection methods may perform better, while also remaining computationally efficient. These include: more robust methods of cross-validation, such as that of Ng (1997); Minimum Description Length (Rissanen, 1983); and Schuurman's metric-based approach, which also uses unlabeled data (1997). Research on improved methods of model selection for our algorithm is an area of future work.

7. Related Work

Expectation-Maximization is a well-known family of algorithms with a long history and many applications. Its application to classification is not new in the statistics literature. The idea of using an EM-like procedure to improve a classifier by "treating the unclassified data as incomplete" is mentioned by R. J. A. Little among the published responses to the original EM paper (Dempster et al., 1977). A discussion of this "partial classification" paradigm and descriptions of further references can be found in McLachlan and Basford's book on mixture models (1988, page 29).

Two recent studies in the machine learning literature have used 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, which, we hypothesize, exacerbate violations of the independence and mixture model assumptions.

Shahshahani and Landgrebe (1994) also theoretically investigate 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 versus 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 by Castelli and Cover (1995), unlabeled data does not improve the classification results in the absence of labeled data.

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. Work by Ghahramani and Jordan (1994) is another example in the machine learning literature of using EM with mixture models to fill in missing values. Whereas we focus on data where the class labels are missing, they focus on data where features other than the class labels are missing. The AutoClass project (Cheeseman & Stutz, 1996) investigates the combination of Expectation-Maximization with a naive Bayes generative model. The emphasis of their research is the discovery of novel clustering for unsupervised learning over unlabeled data.

Our use of multiple mixture components per class is an example of using mixtures to improve modeling of probability density functions. Jaakkola and Jordan (1998) provide a general discussion of using mixtures to improve mean field approximations (of which naive Bayes is an example).

Another paradigm that reduces the need for labeled training examples is active learning. In this scenario, the algorithm repeatedly selects an unlabeled example, asks a human labeler for its true class label, and rebuilds its classifier. Active learning algorithms differ in their methods of selecting the unlabeled example. Three such examples applied to text are "Query By Committee" (Dagan & Engelson, 1995; Liere & Tadepalli, 1997), relevance sampling and uncertainty sampling (Lewis & Gale, 1994; Lewis, 1995).

Recent work by some of the authors combines active learning with Expectation-Maximization (McCallum & Nigam, 1998). EM is applied to the unlabeled documents both to help inform the algorithm's choice of documents for labeling requests, and also to boost accuracy using the documents that remain unlabeled (as in this paper). Experimental results show that the combination of active learning and EM requires only slightly more than half as many labeled training examples to achieve the same accuracy as either active learning or EM alone.

Another effort to use unlabeled data in support of supervised learning is the work on co-training by Blum and Mitchell (1998). They consider a particular subclass of learning problems distinct from the problems addressed in this paper. In particular, they consider problems where a target function $f(x)$ is to be learned, and where the attributes describing each instance x can be partitioned into two sets such that each is sufficient to calculate f. This redundancy in the attributes describing x allows learning two distinct classifiers, then using them to train one another over unlabeled data. They present experimental results showing the success of this approach on a web page classification task, as well as a proof that under certain conditions the target function can be PAC learned given an initial weak classifier and unlabeled data only.

A variety of statistical techniques other than naive Bayes have been applied to text classification, including Support Vector Machines (Joachims, 1998), k nearest neighbor (Yang, 1994, 1999), TFIDF/Rocchio (Salton, 1991; Rocchio, 1971), exponential gradient and covering algorithms (Cohen & Singer, 1996). We use naive Bayes in this paper because is has a strong probabilistic foundation for Expectation-Maximization, and is more efficient for large data sets. Furthermore, 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.

8. Summary and Conclusions

This paper has presented a family of algorithms that address the question of when and how unlabeled data may be used to supplement scarce labeled data, 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 large volumes of unlabeled data. We have presented an algorithm that takes advantage of it and experimental results that show significant improvements by using unlabeled documents for training classifiers in three real-world text classification tasks.

When our assumptions of data generation are correct, basic EM can effectively incorporate information from unlabeled data. However, the full complexity of real-world text data cannot be completely captured by known statistical models. It is interesting then, to consider the performance of a classifier based on generative models that make incorrect assumptions about the data. In such cases, when the data is inconsistent with the assumptions of the model, our method for adjusting the relative contribution of the unlabeled data (EM-λ) prevents the unlabeled data from degrading classification accuracy.

In another augmentation to the basic EM scheme, we study the effect of multiple mixture components per class. This is an effort to relax the assumptions of the model, and make the generative model better match the data. Experimental results show improvements in classification, and suggest the exploration of even more complex mixture models that would correspond even better to textual data distributions. These results also recommend a study of improvements to the current cross-validation methods for selecting both the unlabeled data weight λ and the number of mixture components per class.

We believe that our algorithm and others using unlabeled data require a closer match between the data and the generative model than those using labeled data alone. If the intended target concept and model differ from the actual distribution of the data too strongly, then the use of unlabeled data will hurt instead of help performance. In future work we intend to make a closer theoretical and empirical study of the tradeoffs between the use of unlabeled data and inherent model inadequacies.

We also see several other interesting directions for future work with unlabeled data. Two other task formulations could also benefit from using EM: (1) active learning could use an explicit model of unlabeled data and incorporate EM, both to improve selection of examples for which to request a label and to improve classification accuracy using the examples that remain unlabeled at the end; initial study in this area has already begun (McCallum & Nigam, 1998); (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.

Furthermore, other problem domains share some similarities with text domains, and also have limited, expensive labeled data, but abundant and inexpensive unlabeled data. Robotics, vision, and information extraction are three such domains. Applying the techniques in this paper could improve performance 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 our anonymous reviewers for helpful suggestions and corrections. Doug Baker helped us format the Reuters data set. John Lafferty provided insightful discussions about EM. We thank Jason Rennie for comments on an earlier draft. This research was supported in part by the Darpa HPKB program under contract F30602-97-1-0215.

Notes

1. This assumption will be relaxed in Section 5.3.2 by making this a many-to-one correspondence. Other work (Li & Yamanishi, 1997) relaxes this assumption in a one-to-many fashion.
2. Throughout the paper we use standard notational shorthand for random variables, whereby $\mathrm{P}(X = x_i | Y = y_j)$ is written $\mathrm{P}(x_i | y_j)$ for random variables X and Y taking on values x_i and y_j.
3. Previous naive Bayes formalizations do not include this document length effect. In the most general case, document length should be modeled and parameterized on a class-by-class basis.
4. The Dirichlet is a commonly-used conjugate prior distribution over multinomials. Dirichlet distributions are discussed in more detail by, for example, Stolcke and Omohundro (1994).
5. 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.

6. For all statistical results in this paper, when the number of labeled examples is small, we have multiple trials, and use paired t-tests. When the number of labeled examples is large, we have a single trial, and report results instead with a McNemar test. These tests are discussed further by Dietterich (1998).
7. The weighted log likelihood ratio used to rank the words in Figure 3 is:

$$\mathrm{P}(w_t|c_j;\hat{\theta})\log\left(\frac{\mathrm{P}(w_t|c_j;\hat{\theta})}{\mathrm{P}(w_t|\neg c_j;\hat{\theta})}\right), \tag{18}$$

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 & Thomas, 1991).

References

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT '98)*, pp. 92–100.

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

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

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

Cover, T. M., & Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley and Sons, New York.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., & Slattery, S. (1998). Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the Fifteenth National Conference on Artificial Intellligence (AAAI-98)*, pp. 509–516.

Dagan, I., & Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. In *Machine Learning: Proceedings of the Twelfth International Conference (ICML '95)*, pp. 150–157.

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

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, *10*(7).

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *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*(1), 55–77.

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

Jaakkola, T. S., & Jordan, M. I. (1998). Improving the mean field approximation via the use of mixture distributions. In Jordan, M. I. (Ed.), *Learning in Graphical Models*. Kluwer Academic Publishers.

Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML '97)*, pp. 143–151.

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. In *Machine Learning: ECML-98, Tenth European Conference on Machine Learning*, pp. 137–142.

Koller, D., & Sahami, M. (1997). Hierarchically classifying documents using very few words. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML '97)*, pp. 170–178.

Lang, K. (1995). Newsweeder: Learning to filter netnews. In *Machine Learning: Proceedings of the Twelfth International Conference (ICML '95)*, pp. 331–339.

Larkey, L. S., & Croft, W. B. (1996). Combining classifiers in text categorization. In *SIGIR '96: Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 289–297.

Lewis, D. D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. In *SIGIR '92: Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 37–50.

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

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

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

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

Lewis, D. D., & Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. In *Third Annual Symposium on Document Analysis and Information Retrieval*, pp. 81–93.

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

Liere, R., & Tadepalli, P. (1997). Active learning with committees for text categorization. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, pp. 591–596.

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*. Tech. rep. WS-98-05, AAAI Press. http://www.cs.cmu.edu/~mccallum.

McCallum, A., Rosenfeld, R., Mitchell, T., & Ng, A. (1998). Improving text clasification by shrinkage in a hierarchy of classes. In *Machine Learning: Proceedings of the Fifteenth International Conference (ICML '98)*, pp. 359–367.

McCallum, A. K., & Nigam, K. (1998). Employing EM in pool-based active learning for text classification. In *Machine Learning: Proceedings of the Fifteenth International Conference (ICML '98)*, pp. 350–358.

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

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

Miller, D. J., & 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 9*, pp. 571–577.

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

Ng, A. Y. (1997). Preventing "overfitting" of cross-validation data. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML '97)*, pp. 245–253.

Pazzani, M. J., Muramatsu, J., & Billsus, D. (1996). Syskill & Webert: Identifying interesting Web sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pp. 54–59.

Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, *11*(2), 416–431.

Robertson, S. E., & Sparck-Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, *27*(3), 129–146.

Rocchio, J. (1971). Relevance feedback in information retrieval. In Salton, G. (Ed.), *The SMART Retrieval System:Experiments in Automatic Document Processing.* Prentice Hall, Englewood Cliffs, NJ.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Baysian approach to filtering junk e-mail. In *AAAI-98 Workshop on Learning for Text Categorization.* Tech. rep. WS-98-05, AAAI Press. http://robotics.stanford.edu/users/sahami/papers.html.

Salton, G. (1991). Developments in automatic text retrieval. *Science*, *253*(5023), 974–980.

Schuurmans, D. (1997). A new metric-based approach to model selection. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, pp. 552–558.

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

Shavlik, J., & Eliassi-Rad, T. (1998). Intelligent agents for web-based tasks: An advice-taking approach. In *AAAI-98 Workshop on Learning for Text Categorization.* Tech. rep. WS-98-05, AAAI Press. http://www.cs.wisc.edu/~shavlik/mlrg/publications.html.

Stolcke, A., & Omohundro, S. M. (1994). Best-first model merging for hidden Markov model induction. Tech. rep. TR-94-003, ICSI, University of California, Berkeley. http://www.icsi.berkeley.edu/techreports/1994.html.

Yang, Y. (1994). Expert network: Effective and efficient learning from human decisions in text categorization and retrieval.. In *SIGIR '94: Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 13–22.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*. To appear.

Yang, Y., & Pederson, J. O. (1997). Feature selection in statistical learning of text categorization. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML '97)*, pp. 412–420.