Employing EM and Pool-Based Active Learning for Text Classification

Andrew Kachites McCallum[‡†]
mccallum@justresearch.com
[‡]Just Research
4616 Henry Street
Pittsburgh, PA 15213

Kamal Nigam[†]
knigam@cs.cmu.edu
[†]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper shows how a text classifier's need for labeled training documents can be reduced by taking advantage of a large pool of unlabeled documents. We modify the Query-by-Committee (QBC) method of active learning to use the unlabeled pool for explicitly estimating document density when selecting examples for labeling. Then active learning is combined with Expectation-Maximization in order to "fill in" the class labels of those documents that remain unlabeled. Experimental results show that the improvements to active learning require less than two-thirds as many labeled training examples as previous QBC approaches, and that the combination of EM and active learning requires only slightly more than half as many labeled training examples to achieve the same accuracy as either the improved active learning or EM alone.

1 Introduction

Obtaining labeled training examples for text classification is often expensive, while gathering large quantities of unlabeled examples is usually very cheap. For example, consider the task of learning which web pages a user finds interesting. The user may not have the patience to hand-label a thousand training pages as interesting or not, yet multitudes of unlabeled pages are readily available on the Internet.

This paper presents techniques for using a large pool of unlabeled documents to improve text classification when labeled training data is sparse. We enhance the QBC active learning algorithm to select labeling requests from the entire pool of unlabeled documents, and explicitly use the pool to estimate regional document density. We also combine active learning with Expectation-Maximization (EM) in order to take advantage of the word co-occurrence information contained in the many documents that remain in the unlabeled pool.

In previous work [Nigam *et al.* 1998] we show that combining the evidence of labeled and unlabeled documents via EM can reduce text classification error by one-third. We treat the absent labels as "hidden variables" and use EM to fill them in. EM improves the classifier by alternately using the current classifier to guess the hidden variables, and then using the current guesses to advance classifier training. EM consequently finds the classifier parameters that locally maximize the probability of both the labeled and unlabeled data.

Active learning approaches this same problem in a different way. Unlike our EM setting, the active learner can request the true class label for certain unlabeled documents it selects. However, each request is considered an expensive operation and the point is to perform well with as few queries as possible. Active learning aims to select the most informative examples—in many settings defined as those that, if their class label were known, would maximally reduce classification error and variance over the distribution of examples [Cohn, Ghahramani, & Jordan 1996]. When calculating this in closed-form is prohibitively complex, the *Query-by-Committee* (QBC) algorithm [Freund *et al.* 1997] can be used to select documents that have high classification variance themselves. QBC measures the variance indirectly, by examining the disagreement among class labels assigned by a set of classifier variants, sampled from the probability distribution of classifiers that results from the labeled training examples.

This paper shows that a pool of unlabeled examples can be used to benefit both active learning and EM. Rather than having active learning choose queries by synthetically generating them (which is awkward with text), or by selecting examples from a stream (which inefficiently models the data distribution), we advocate selecting the best examples from the entire pool of unlabeled documents (and using the pool to explicitly model density); we call this last scheme *pool-based sampling*. In experimental results on a real-world text data set, this technique is shown to reduce the need for labeled documents by 42% over previous QBC approaches. Furthermore, we show that the *combination* of QBC and EM learns with fewer labeled examples than either individually—requiring only 58% as many labeled examples as EM alone, and only 26% as many as QBC alone. We also discuss our initial approach to a richer combination we call *pool-leveraged sampling* that interleaves active learning and EM such that EM's modeling of the unlabeled data informs the selection of active learning queries.

2 Probabilistic Framework for Text Classification

This section presents a Bayesian probabilistic framework for text classification. The next two sections add EM and active learning by building on this framework. We approach the task of text classification from a Bayesian learning perspective: we assume that the documents are generated by a particular parametric model, and use training data to calculate Bayes-optimal estimates of the model parameters. Then, we use these estimates to classify new test documents by turning the generative model around with Bayes' rule, calculating the probability that each class would have generated the test document in question, and selecting the most probable class.

Our parametric model is naive Bayes, which is based on commonly used assumptions [Friedman 1997; Joachims 1997]. First we assume that text documents are generated by a mixture model (parameterized by θ), and that there is a one-to-one correspondence between the (observed) class labels and the mixture components. We use the notation $c_j \in \mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$ to indicate both the jth component and jth class. Each component c_j is parameterized by a disjoint subset of θ. These assumptions specify that a document is created by (1) selecting a class according to the prior probabilities, $\mathrm{P}(c_j|\theta)$, then (2) having that class component generate a document according to its own parameters, with distribution $\mathrm{P}(d_i|c_j;\theta)$. We can characterize the likelihood of a document as a sum of total probability over all generative components:

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

Document d_i is considered to be an ordered list of word events. We write $w_{d_{ik}}$ for the word in position k of document d_i, where the subscript of w indicates an index into the vocabulary $V = \langle w_1, w_2, \ldots, w_{|V|}\rangle$. We make the standard naive Bayes assumption: that the words of a document are generated independently of context, that is, independently of the other words in the same document given the class. We further assume that the probability of a word is independent of its position within the document. Thus, we can express the class-conditional probability of a document by taking the product of the probabilities of the independent word events:

$$\mathrm{P}(d_i|c_j;\theta) = \mathrm{P}(|d_i|) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{ik}}|c_j;\theta), \tag{2}$$

where we assume the length of the document, $|d_i|$, is distributed independently of class. Each individual class component is parameterized by the collection of word probabilities, such that $\theta_{w_t|c_j} = \mathrm{P}(w_t|c_j;\theta)$, where $t \in \{1, \ldots, |V|\}$ and $\sum_t \mathrm{P}(w_t|c_j;\theta) = 1$. The other parameters of the model are the class prior probabilities $\theta_{c_j} = \mathrm{P}(c_j|\theta)$, which indicate the probabilities of selecting each mixture component.

Given these underlying assumptions of how the data are produced, the task of learning a text classifier consists of forming an estimate of θ, written $\hat{\theta}$, based on a set of training data. With labeled training documents, $\mathcal{D} = \{d_1, \ldots, d_{|\mathcal{D}|}\}$, we can calculate estimates for the parameters of the model that generated these documents. To calculate the probability of a word given a class, $\theta_{w_t|c_j}$, simply count the fraction of times the word occurs in the data for that class, augmented with a Laplacean prior. This smoothing prevents probabilities of zero for infrequently occurring words. These word probability estimates $\hat{\theta}_{w_t|c_j}$ are:

$$\hat{\theta}_{w_t|c_j} = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} N(w_t, d_i)\mathrm{P}(c_j|d_i)}{|V| + \sum_{s=1}^{|V|}\sum_{i=1}^{|\mathcal{D}|} N(w_s, d_i)\mathrm{P}(c_j|d_i)}, \tag{3}$$

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

$$\hat{\theta}_{c_j} = \frac{\sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}{|\mathcal{D}|}. \quad (4)$$

Given estimates of these parameters calculated from the training documents, it is possible to turn the generative model around and calculate the probability that a particular class component generated a given document. We formulate this by an application of Bayes' rule, and then substitutions using Equations 1 and 2:

$$\mathrm{P}(c_j|d_i;\hat{\theta}) = \frac{\mathrm{P}(c_j|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{ik}}|c_j;\hat{\theta})}{\sum_{r=1}^{|\mathcal{C}|} \mathrm{P}(c_r|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{ik}}|c_r;\hat{\theta})}. \quad (5)$$

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

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

3 EM and Unlabeled Data

When naive Bayes is given just a small set of labeled training data, classification accuracy will suffer because variance in the parameter estimates of the generative model will be high. However, by augmenting this small set with a large set of unlabeled data and combining the two pools with EM, we can improve the parameter estimates. This section describes how to use EM to combine these pools within the probabilistic framework of the previous section.

EM is a class of iterative algorithms for maximum likelihood estimation in problems with incomplete data [Dempster, Laird, & Rubin 1977]. Given a model of data generation, and data with some missing values, EM alternately uses the current model to estimate the missing values, and then uses the missing value estimates to improve the model. Using all the available data, EM will locally maximize the likelihood of the generative parameters, giving estimates for the missing values.

In our text classification setting, we treat the class labels of the unlabeled documents as missing values, and then apply EM. The resulting naive Bayes parameter estimates often give significantly improved classification accuracy on the test set when the pool of labeled examples is small [Nigam *et al.* 1998].[1] This use of EM is a special case of a more general missing values formulation [Ghahramani & Jordan 1994].

In implementation, EM is an iterative two-step process. The E-step calculates probabilistically-weighted class labels, $\mathrm{P}(c_j|d_i;\hat{\theta})$, for every unlabeled document using a current estimate of θ and Equation 5. The M-step calculates a new maximum likelihood estimate for θ using all the labeled data, both original and probabilistically labeled, by Equations 3 and 4. We initialize the process with parameter estimates using just the labeled training data, and iterate until $\hat{\theta}$ reaches a fixed point. See [Nigam *et al.* 1998] for more details.

4 Active Learning with EM

Rather than estimating class labels for unlabeled documents, as EM does, active learning instead requests the *true* class labels for unlabeled documents it selects. In many settings, an optimal active learner should select those documents that, when labeled and incorporated into training, will minimize classification error over the distribution of future documents. Equivalently in probabilistic frameworks without bias, active learning aims to minimize the expected classification variance over the document distribution. Note that Naive Bayes' independence assumption and Laplacean priors do introduce bias. However, variance tends to dominate bias in classification error [Friedman 1997], and thus we focus on reducing variance.

The Query-by-Committee (QBC) method of active learning measures this variance indirectly [Freund *et al.* 1997]. It samples several times from the classifier parameter distribution that results from the training

[1] When the classes do not correspond to the natural clusters of the data, EM can hurt accuracy instead of helping. Our previous work also describes a method for avoiding these detrimental effects.

data, in order to create a "committee" of classifier variants. This committee approximates the entire classifier distribution. QBC then classifies unlabeled documents with each committee member, and measures the disagreement between their classifications—thus approximating the classification variance. Finally, documents on which the committee disagrees strongly are selected for labeling requests. The newly labeled documents are included in the training data, and a new committee is sampled for making the next set of requests. This section presents each of these steps in detail, and then explains its integration with EM. Our implementation of this algorithm is summarized in Table 1.

Our committee members are created by sampling classifiers according to the distribution of classifier parameters specified by the training data. Since the probability of the naive Bayes parameters for each class are described by a Dirichlet distribution, we sample the parameters $\theta_{w_t|c_j}$ from the posterior Dirichlet distribution based on training data word counts, $N(\cdot,\cdot)$. This is performed by drawing weights, v_{tj}, for each word w_t and class c_j from the Gamma distribution: $v_{tj} = \mathrm{Gamma}(\alpha_t + N(w_t, c_j))$, where α_t is always 1, as specified by our Laplacean prior. Then we set the parameters $\theta_{w_t|c_j}$ to the normalized weights by $\theta_{w_t|c_j} = v_{tj}/\sum_s v_{sj}$. We sample to create a classifier k times, resulting in k committee members. Individual committee members are denoted by m.

We consider two metrics for measuring committee disagreement. The previously employed *vote entropy* [Dagan & Engelson 1995] is the entropy of the class label distribution resulting from having each committee member "vote" with probability mass $1/k$ for its winning class. One disadvantage of vote entropy is that it does not consider the confidence of the committee members' classifications, as indicated by the class probabilities $\mathrm{P}_m(c_j|d_i;\hat{\theta})$ from each member.

To capture this information, we propose to measure committee disagreement for each document using *Kullback-Leibler divergence to the mean* [Pereira, Tishby, & Lee 1993]. Unlike vote entropy, which compares only the committee members' top ranked class, KL divergence measures the strength of the certainty of disagreement by calculating differences in the committee members' class distributions, $\mathrm{P}_m(C|d_i)$.[2] Each committee member m produces a posterior class distribution, $\mathrm{P}_m(C|d_i)$, where C is a random variable over classes. KL divergence to the mean is an average of the KL divergence between each distribution and the mean of all the distributions:

$$\frac{1}{k}\sum_{m=1}^{k} D\left(\mathrm{P}_m(C|d_i)||\mathrm{P}_{avg}(C|d_i)\right), \tag{6}$$

where $\mathrm{P}_{avg}(C|d_i)$ is the class distribution mean over all committee members, m: $\mathrm{P}_{avg}(C|d_i) = (\sum_m \mathrm{P}_m(C|d_i))/k$.

KL divergence, $D(\cdot||\cdot)$, is an information-theoretic measure of the difference between two distributions, capturing the number of extra "bits of information" required to send messages sampled from the first distribution using a code that is optimal for the second. The KL divergence between distributions $\mathrm{P}_1(C)$ and $\mathrm{P}_2(C)$ is:

$$D(\mathrm{P}_1(C)||\mathrm{P}_2(C)) = \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}_1(c_j)\log\left(\frac{\mathrm{P}_1(c_j)}{\mathrm{P}_2(c_j)}\right). \tag{7}$$

- Calculate the density for each document. (Eq. 9)
- Loop while adding documents:
  - Build an initial estimate of $\hat{\theta}$ from the labeled documents only. (Eqs. 3 and 4)
  - Loop k times, once for each committee member:
    - + Create a committee member by sampling for each class from the appropriate Dirichlet distribution.
    - + *Starting with the sampled classifier apply EM with the unlabeled data. Loop while parameters change:*
      - · *Use the current classifier to probabilistically label the unlabeled documents. (Eq. 5)*
      - · *Recalculate the classifier parameters given the probabilistically-weighted labels. (Eqs. 3 and 4)*
    - + Use the current classifier to probabilistically label all unlabeled documents. (Eq. 5)
  - Calculate the disagreement for each unlabeled document (Eq. 7), multiply by its density, and request the class label for the one with the highest score.
- Build a classifier with the labeled data. (Eqs. 3 and 4).
- *Starting with this classifier, apply EM as above.*

Table 1: Our active learning algorithm. Traditional Query-by-Committee omits the EM steps, indicated by italics, does not use the density, and works in a stream-based setting.

[2]While naive Bayes is not an accurate probability estimator [Domingos & Pazzani 1997], naive Bayes classification scores are somewhat correlated to confidence; the fact that naive Bayes scores can be successfully used to make accuracy/coverage trade-offs is testament to this.

After disagreement has been calculated, a document is selected for a class label request. (Selecting more than one document at a time can be a computational convenience.) We consider three ways of selecting documents: stream-based, pool-based, and density-weighted pool-based. Some previous applications of QBC [Dagan & Engelson 1995; Liere & Tadepalli 1997] use a simulated stream of unlabeled documents. When a document is produced by the stream, this approach measures the classification disagreement among the committee members, and decides, based on the disagreement, whether to select that document for labeling. Dagan and Engelson do this heuristically by dividing the vote entropy by the maximum entropy to create a probability of selecting the document. Disadvantages of using *stream-based sampling* are that it only sparsely samples the full distribution of possible document labeling requests, and that the decision to label is made on each document individually, irrespective of the alternatives.

An alternative that aims to address these problems is *pool-based sampling*. It selects from among *all* the unlabeled documents in a pool the one with the largest disagreement. However, this loses one benefit of stream-based sampling—the implicit modeling of the data distribution—and it may select documents that have high disagreement, but are in unimportant, sparsely populated regions.

We can retain this distributional information by selecting documents using both the classification disagreement and the "density" of the region around a document. This *density-weighted pool-based sampling* method prefers documents with high classification variance that are also similar to many other documents. The stream approach approximates this implicitly; we accomplish this more accurately, (especially when labeling a small number of documents), by modeling the density explicitly.

We approximate the density in a region around a particular document by measuring the average distance from that document to all other documents. Distance, Y, between individual documents is measured by using exponentiated KL divergence:

$$Y(d_i, d_h) = e^{-\beta\, D(\mathrm{P}(W|d_h)\ ||\ (\lambda \mathrm{P}(W|d_i) + (1-\lambda)\mathrm{P}(W)))}, \tag{8}$$

where W is a random variable over words in the vocabulary; $\mathrm{P}(W|d_i)$ is the maximum likelihood estimate of words sampled from document d_i, (*i.e.*, $\mathrm{P}(w_t|d_i) = N(w_t, d_i)/|d_i|$); $\mathrm{P}(W)$ is the marginal distribution over words; λ is a parameter that determines how much smoothing to use on the encoding distribution (we must ensure no zeroes here to prevent infinite distances); and β is a parameter that determines the sharpness of the distance metric.

In essence, the average KL divergence between a document, d_i, and all other documents measures the degree of overlap between d_i and all other documents; exponentiation converts this information-theoretic number of "bits of information" into a scalar distance.

When calculating the average distance from d_i to all other documents it is much more computationally efficient to calculate the geometric mean than the arithmetic mean, because the distance to all documents that share no words words with d_i can be calculated in advance, and we only need make corrections for the words that appear in d_i. Using a geometric mean, we define density, Z of document d_i to be

$$Z(d_i) = e^{\frac{1}{|\mathcal{D}|}\sum_{d_h \in \mathcal{D}} \ln(Y(d_i, d_h))}. \tag{9}$$

We combine this density metric with disagreement by selecting the document that has the largest product of density (Equation 9) and disagreement (Equation 6). This density-weighted pool-based sampling selects the document that is representative of many other documents, and about which there is confident committee disagreement.

Combining Active Learning and EM

Active learning can be combined with EM by running EM to convergence after actively selecting all the training data that will be labeled. This can be understood as using active learning to select a better starting point for EM hill climbing, instead of randomly selecting documents to label for the starting point. A more interesting approach, that we term *pool-leveraged sampling*, is to interleave EM with active learning, so that EM not only builds on the results of active learning, but EM also informs active learning. To do this we run EM to convergence on each committee member before performing the disagreement calculations. The intended effect is (1) to avoid requesting labels for examples whose label can be reliably filled in by EM, and (2) to encourage the selection of examples that will help EM find a local maximum with higher classification accuracy. With more accurate committee members, QBC should pick more informative documents to label. The complete active learning algorithm, both with and without EM, is summarized in Table 1.

Unlike settings in which queries must be generated [Cohn 1994], and previous work in which the unlabeled data is available as a stream [Dagan & Engelson 1995; Liere & Tadepalli 1997; Freund *et al.* 1997], our assumption about the availability of a pool of unlabeled data makes the improvements to active learning possible. This pool is present for many real-world tasks in which efficient use of labels is important, especially in text learning.

5 Related Work

A similar approach to active learning, but without EM, is that of Dagan and Engelson [1995]. They use QBC stream-based sampling and vote entropy. In contrast, we advocate density-weighted pool-based sampling and the KL metric. Additionally, we select committee members using the Dirichlet distribution over classifier parameters, instead of approximating this with a Normal distribution. Several other studies have investigated active learning for text categorization. Lewis and Gale examine uncertainty sampling and relevance sampling in a pool-based setting [Lewis & Gale 1994; Lewis 1995]. These techniques select queries based on only a single classifier instead of a committee, and thus cannot approximate classification variance. Liere and Tadepalli [1997] use committees of Winnow learners for active text learning. They select documents for which two randomly selected committee members disagree on the class label.

In previous work, we show that EM with unlabeled data reduces text classification error by one-third [Nigam *et al.* 1998]. Two other studies have used EM to combine labeled and unlabeled data without active learning for classification, but on non-text tasks [Miller & Uyar 1997; Shahshahani & Landgrebe 1994]. Ghahramani and Jordan [1994] use EM with mixture models to fill in missing feature values.

6 Experimental Results

This section provides evidence that using a combination of active learning and EM performs better than using either individually. The results are based on data sets from UseNet and Reuters.[3]

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

The Newsgroups data set, collected by Ken Lang, contains about 20,000 articles evenly divided among 20 UseNet discussion groups [Joachims 1997]. We use the five `comp.*` classes as our data set. When tokenizing this data, we skip the UseNet headers (thereby discarding the subject line); tokens are formed from contiguous alphabetic characters, removing words on a stoplist of common words. Best performance was obtained with no feature selection, no stemming, and by normalizing word counts by document length. The resulting vocabulary, after removing words that occur only once, has 22958 words. On each trial, 20% of the documents are randomly selected for placement in the test set.

The 'ModApte' train/test split of the Reuters 21578 Distribution 1.0 data set consists of 12902 Reuters newswire articles in 135 overlapping topic categories. Following several other studies [Joachims 1998; Liere & Tadepalli 1997] we build binary classifiers for each of the 10 most populous classes. We ignore words on a stoplist, but do not use stemming. The resulting vocabulary has 19371 words. Results are reported on the complete test set as precision-recall breakeven points, a standard information retrieval measure for binary classification [Joachims 1998].

In our experiments, an initial classifier was trained with one randomly-selected labeled document per class. Active learning proceeds as described in Table 1. Newsgroups experiments were run for 200 active learning iterations, each round selecting one document for labeling. Reuters experiments were run for 100 iterations, each round selecting five documents for labeling. Smoothing parameter λ is 0.5; sharpness parameter β is 3. We made little effort to tune β and none to tune λ. For QBC we use a committee size of three (k=3); initial experiments show that committee size has little effect. All EM runs perform seven EM iterations; we never found classification accuracy to improve beyond the seventh iteration. All results presented are averages of ten runs per condition.

The top graph in Figure 1 shows a comparison of different disagreement metrics and selection strategies for QBC without EM. The best combination, density-weighted pool-based sampling with a KL divergence to the mean disagreement metric achieves 51% accuracy after acquiring only 30 labeled documents. To reach the same accuracy, unweighted pool-based sampling with KL disagreement needs 40 labeled documents. If we switch to stream-based, sampling, KL disagreement needs 51 labelings for 51% accuracy. Our random selection baseline requires 59 labeled documents.

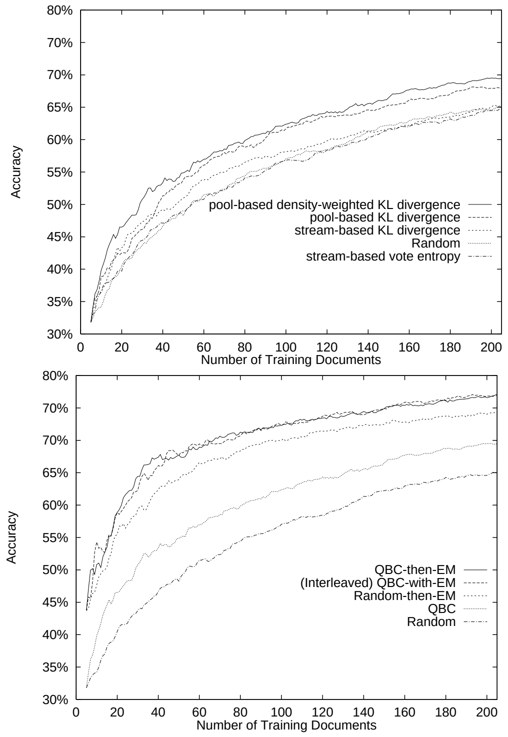

Figure 1: On the top, a comparison of disagreement metrics and selection strategies for QBC shows that density-weighted pool-based KL sampling does better than other metrics. On the bottom, combinations of QBC and EM outperform stand-alone QBC or EM. In these cases, QBC uses density-weighted pool-based KL sampling. Note that the order of the legend matches the order of the curves and that, for resolution, the vertical axes do not range from 0 to 100.

Surprisingly, stream-based vote entropy does slightly worse than random, needing 61 documents for the 51% threshold. Density-weighted pool-based sampling with a KL metric is statistically significantly better than each of the other methods ($p < 0.005$ for each pairing). It is interesting to note that the first several documents selected by this approach are usually FAQs for the various newsgroups. Thus, using a pool of unlabeled data can notably improve active learning.

In contrast to earlier work on part-of-speech tagging [Dagan & Engelson 1995], vote entropy does not perform well on document classification. In our experience, vote entropy tends to select outliers—documents that are short or unusual. We conjecture that this occurs because short documents and documents consisting of infrequently occurring words are the documents that most easily have their classifications changed by perturbations in the classifier parameters. In these situations, classification variance is high, but the difference in magnitude between the classification score of the winner and the losers is small. For vote entropy, these are prime selection candidates, but KL divergence accounts for the magnitude of the differences, and thus helps measure the confidence in the disagreement. Furthermore, incorporating density-weighting biases selection towards longer documents, since these documents have word distributions that are more representative of the corpus, and thus are considered "more dense." It is generally better to label long rather than short documents because, for the same labeling effort, a long document provides information about more words. Dagan and Engelson's domain, part-of-speech tagging, does not have varying length examples; document classification does.

Now consider the addition of EM to the learning scheme. Our EM baseline post-processes random selection with runs of EM (Random-then-EM). The most straightforward method of combining EM and active learning is to run EM after active learning completes (QBC-then-EM). We also interleave EM and active learning, by running EM on each committee member (QBC-with-EM). This also includes a post-processing run of EM. In QBC, documents are selected by density-weighted pool-based KL, as the previous experiment indicated was best. Random selection (Random) and QBC without EM (QBC) are repeated from the previous experiment for comparison.

The bottom graph of Figure 1 shows the results of combining EM and active learning. Starting with the 30 labeling mark again, QBC-then-EM is impressive, reaching 64% accuracy. Interleaved QBC-with-EM lags only slightly, requiring 32 labeled documents for 64% accuracy. Random-then-EM is the next best performer, needing 51 labeled documents. QBC, without EM, takes 118 labeled documents, and our baseline, Random, takes 179 labeled documents to reach 64% accuracy. QBC-then-EM and QBC-with-EM are not statistically significantly different ($p = 0.71$ N.S.); these two are each statistically significantly better than each of the other methods at this threshold ($p < 0.05$).

These results indicate that the combination of EM and active learning provides a large benefit. However, QBC interleaved with EM does not perform better than QBC followed by EM—not what we were expecting. We hypothesize that while the interleaved method tends to label documents that EM cannot reliably label on its own, these documents do not provide the most beneficial starting point for EM's hill-climbing. In ongoing work we are examining this more closely and investigating improvements.

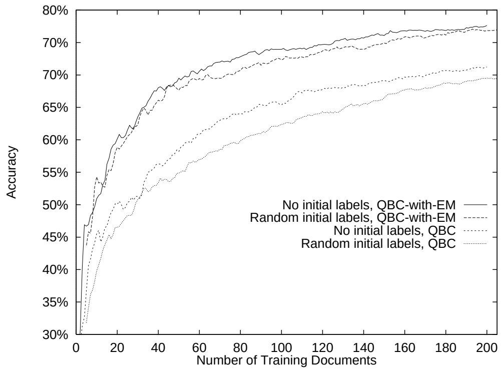

Figure 2: A comparison of random initial labeling and no initial labeling when documents are selected with density-weighted pool-based sampling. Note that no initial labeling tends to dominate the random initial labeling cases.

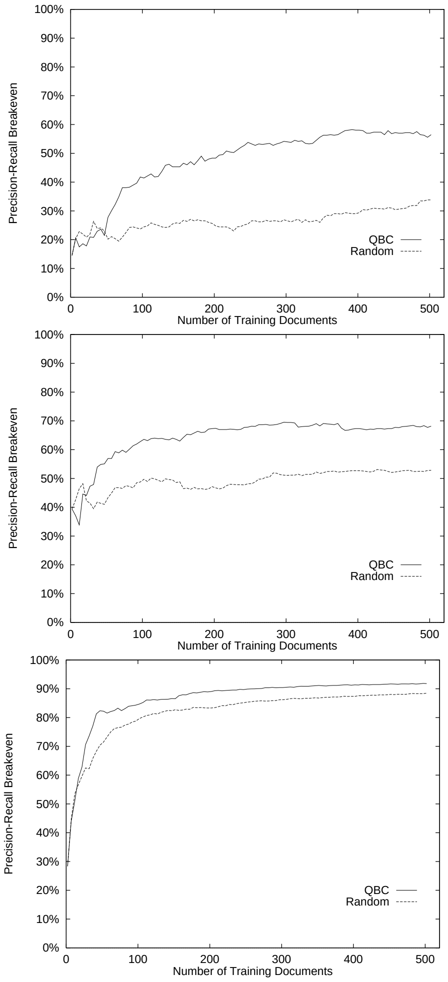

Figure 3: Active learning results on three categories of the Reuters data, corn, trade, and acq, respectively from the top and in increasing order of frequency. Note that active learning with committees outperforms random selection and that the magnitude of improvement is larger for more infrequent classes.

Another application of the unlabeled pool to guiding active learning is the selection of the initial labeled examples. Several previous implementations [Dagan & Engelson 1995; Lewis & Gale 1994; Lewis 1995] suppose that the learner is provided with a collection of labeled examples at the beginning of active learning. However, obtaining labels for these initial examples (and making sure we have examples from each class) can itself be an expensive proposition. Alternatively, our method can begin without any labeled documents, sampling from the Dirichlet distribution and selecting with density-weighted metrics as usual. Figure 2 shows results from experiments that begin with zero labeled documents, and use the structure of the unlabeled data pool to select initial labeling requests. Interestingly, this approach is not only more convenient for many real-world tasks, but also performs better because, even without any labeled documents, it can still select documents in dense regions. With 70 labeled documents, QBC initialized with one (randomly selected) document per class attains an average of 59% accuracy, while QBC initialized with none (relying on density-weighted KL divergence to select all 70) attains an average of 63%. Performance also increased with EM; QBC-with-EM rises from 69% to 72% when active learning begins with zero labeled documents. Each of these differences is statistically significant ($p < 0.005$). Both with and without EM, this method successfully finds labeling requests to cover all classes. As before, the first requests tend to be FAQs or similar, long, informative documents.

In comparison to previous active learning studies in text classification domains [Lewis & Gale 1994; Liere & Tadepalli 1997], the magnitude of our classification accuracy increase is relatively modest. Both of these previous studies consider binary classifiers with skewed distributions in which the positive class has a very small prior probability. With a very infrequent positive class, random selection should perform extremely poorly because nearly all documents selected for labeling will be from the negative class. In tasks where the class priors are more even, random selection should perform much better—making the improvement of active learning less dramatic. With an eye towards testing this hypothesis, we perform a subset of our previous experiments on the Reuters data set, which has these skewed priors. We compare Random against unweighted pool-based sampling (QBC) with the KL disagreement metric.

Figure 3 shows results for three of the ten binary classification tasks. The frequencies of the positive classes are 0.018, 0.038 and 0.184 for the corn (top), trade (middle) and acq (bottom) graphs, respectively. The class frequency and active learning results are representative of the spectrum of the ten classes. In all cases, active learning classification is more accurate than Random. After 252 labelings, improvements of accuracy over random are from 27% to 53% for corn, 48% to 68% for trade, and 85% to 90% for acq. The distinct trend across all ten categories is that the less frequently occurring positive classes show larger improvements with active learning. Thus, we conclude that our earlier accuracy improvements are good, given that with unskewed class priors, Random selection provides a relatively strong performance baseline.

7 Conclusions

This paper demonstrates that by leveraging a large pool of unlabeled documents in two ways—using EM and density-weighted pool-based sampling—we can strongly reduce the need for labeled examples. In future work, we will explore the use of a more direct approximation of the expected reduction in classification variance across the distribution. We will consider the effect of the poor probability estimates given by naive Bayes by exploring other classifiers that give more realistic probability estimates. We will also further investigate ways of interleaving active learning and EM to achieve a more than additive benefit.

Acknowledgments

We are grateful to Larry Wasserman for help on theoretical aspects of this work. We thank Doug Baker for help formatting the Reuters data set. Two anonymous reviewers provided very helpful comments. This research was supported in part by the Darpa HPKB program under contract F30602-97-1-0215.

References

Cohn, D.; Ghahramani, Z.; and Jordan, M. 1996. Active learning with statistical models. *Journal of Artificial Intelligence Research* 4:129–145.

Cohn, D. 1994. Neural network exploration using optimal experiment design. In *NIPS 6*.

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

Dagan, I., and Engelson, S. 1995. Committee-based sampling for training probabilistic classifiers. In *ICML-95*.

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

Domingos, P., and Pazzani, M. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* 29:103–130.

Freund, Y.; Seung, H.; Shamir, E.; and Tishby, N. 1997. Selective sampling using the query by committee algorithm. *Machine Learning* 28:133–168.

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

Ghahramani, Z., and Jordan, M. 1994. Supervised learning from incomplete data via an EM approach. In *NIPS 6*.

Joachims, T. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *ICML-97*.

Joachims, T. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. In *ECML-98*.

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

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

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

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

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

Miller, D. J., and Uyar, H. S. 1997. A mixture of experts classifier with learning based on both labelled and unlabelled data. In *NIPS 9*.

Nigam, K.; McCallum, A.; Thrun, S.; and Mitchell, T. 1998. Learning to classify text from labeled and unlabeled documents. In *AAAI-98*.

Pereira, F.; Tishby, N.; and Lee, L. 1993. Distributional clustering of English words. In *Proceedings of the 31st ACL*.

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