Pool-Based Active Learning for Text Classification

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

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

Abstract

This paper shows how a text classifier's need for labeled training documents can be reduced by employing a large pool of unlabeled documents. We modify the Query-by-Committee (QBC) method of active learning to use the unlabeled pool by 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 reduce the need for labelings by one-third over 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 EM or active learning alone.

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 by selecting labeling requests from the entire pool of unlabeled documents, and explicitly using the pool to estimate regional document density. We also combine active learning with Expectation-Maximization (EM) in order to take advantage of the extensive 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—consequently finding 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—those that if their class label were known, would maximally reduce classification error and variance over the distribution of examples [Cohn *et al.* 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 resulting from the labeled training examples.

This paper shows that a pool of unlabeled examples can be to used to good effect by 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 one-third 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 25% as many as QBC alone. We also discuss work in progress on 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.

Naive Bayes and EM

This section presents a Bayesian probabilistic framework for text classification and a method for incorporating unlabeled data within the framework by using Expectation-Maximization. Our parametric model is naive Bayes. First we assume that text documents are generated by a mixture model, parameterized by θ. The mixture model consists of generative components $c_j \in \mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$. We assume each component corresponds to a class; c_j indicates both the jth

mixture component and the jth class. Thus 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, $\mathrm{P}(d_i|\theta) = \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta)$.

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:

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

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

$$\hat{\theta}_{w_t|c_j} = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} N(w_t, d_i)\mathrm{P}(c_j|d_i)}{|V| + \sum_{s=1}^{|V|}\sum_{i=1}^{|\mathcal{D}|} N(w_s, d_i)\mathrm{P}(c_j|d_i)}, \qquad (2)$$

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

$$\hat{\theta}_{c_j} = \frac{\sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}{|\mathcal{D}|}. \qquad (3)$$

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

$$\mathrm{P}(c_j|d_i;\hat{\theta}) = \frac{\mathrm{P}(c_j|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{ik}}|c_j;\hat{\theta})}{\sum_{r=1}^{|\mathcal{C}|} \mathrm{P}(c_r|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{ik}}|c_r;\hat{\theta})}. \qquad (4)$$

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

When naive Bayes is given just a small set of labeled training data, classification accuracy suffers because parameter estimates of the generative model are poor. However, by augmenting this small set with a large set of unlabeled data and combining the two pools with EM, we can improve our parameter estimates. EM is a class of iterative algorithms for maximum likelihood estimation in problems with incomplete data [Dempster *et al.* 1977]. Given a model of data generation, and data with some missing values, EM converges to estimates of the missing values that locally maximize the likelihood of the generative parameters. By treating the class labels of the unlabeled data as missing values, and running EM on the entire data set, the resulting parameter estimates give higher classification accuracy for new documents when the pool of labeled examples is small [Nigam *et al.* 1998]. In practice, EM is an iterative two-step process. The E-step calculates probabilistically-weighted class labels, $\mathrm{P}(c_j|d_i)$, for every unlabeled document using a current estimate of θ and Equation 4. The M-step calculates a new maximum likelihood estimate for θ using all the labeled data, both original and probabilistically labeled, by Equations 2 and 3. We initialize the process with parameter estimates using just the labeled training data, and iterate until $\hat{\theta}$ reaches a fixed point.

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. Optimally, an active learner selects 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.

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 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. Our implementation of this algorithm is summarized in Table 1. This section presents each step of QBC in detail, and then explains its integration with EM.

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: $v_{tj} = \text{Gamma}(\alpha_t + N(w_t, c_j))$, 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 and 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 $\text{P}(c_j|d_i;\hat{\theta})$ from each member.

To capture this information, we propose to measure committee disagreement for each document using *Kullback-Leibler divergence to the mean* [Pereira *et al.* 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, $\text{P}_m(C|d_i)$.[1] Each committee member m produces a posterior class distribution, $\text{P}_m(C|d_i)$, 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:

[1]While naive Bayes is not an accurate probability estimator [Domingos and 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.

- Calculate the density for each document. (Eq. 8)
- Loop while adding documents:
  - Build an initial estimate of $\hat{\theta}$ from the labeled documents only. (Eqs. 2 and 3)
  - Loop k times, once for each committee member:
    - + Create a committee member by sampling for each class from the appropriate Dirichlet distribution. (Page 3)
    - + *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. 4)*
      - · *Recalculate the classifier parameters given the probabilistically-weighted labels. (Eqs. 2 and 3)*
    - + Use the current classifier to probabilistically label all unlabeled documents. (Eq. 4)
  - Calculate the disagreement for each unlabeled document (Eq. 6), multiply by its density, and request the class label for the one with the highest score.
- Build a classifier with the labeled data. (Eqs. 2 and 3).
- *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, and does not use the density.

$$\frac{1}{k}\sum_{m=1}^{k} D\left(\text{P}_m(C|d_i)||\text{P}_{avg}(C|d_i)\right). \qquad (5)$$

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

KL divergence, $D(\cdot||\cdot)$, is an information-theoretic measure of the inefficiency of sending messages sampled from the first distribution using a code that is optimal for the second. The KL divergence between distributions $\text{P}_1(C)$ and $\text{P}_2(C)$ is:

$$D(\text{P}_1(C)||\text{P}_2(C)) = \sum_{j=1}^{|\mathcal{C}|} \text{P}_1(c_j)\log\left(\frac{\text{P}_1(c_j)}{\text{P}_2(c_j)}\right). \qquad (6)$$

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. Previous applications of QBC [Dagan and Engelson 1995; Liere and 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 by heuristically scaling the vote entropy score to 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. Thus finding the very best requests is elusive.

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 third selection 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:

$$Y(d_i, d_h) = e^{-\beta\, D(\mathrm{P}(W|d_h)\ ||\ (\lambda \mathrm{P}(W|d_i)+(1-\lambda)\mathrm{P}(W)))}, \tag{7}$$

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 to which a class label on d_i informs the classifier about all other documents. 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.[2] Using a geometric mean, we define density, Z of document d_i to be

$$Z(d_i) = e^{\frac{1}{|\mathcal{D}|}\sum_{d_h \in \mathcal{D}} \ln(Y(d_i, d_h))}. \tag{8}$$

[2]In the same vein, using KL divergence to the mean instead of KL divergence to calculate Y would have avoided the need for the λ parameter, but, doing so would have precluded efficient calculation of the average.

We combine this density metric with disagreement by selecting the document that has the largest product of density (Equation 8) and disagreement (Equation 5). 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, 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 and Engelson 1995; Liere and Tadepalli 1997; Freund *et al.* 1997], our assumption about the availability of a pool of unlabeled data makes the leverage possible. This pool is present for many real-world tasks in which efficient use of labels is important, especially in text learning.

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 a 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 [Lewis and Gale 1994; Lewis 1995]. These pool-based techniques select queries based on only a single classifier instead of a committee, and thus cannot approximate classification variance reduction. Liere and Tadepalli [1997] use committees of Winnow learners for active text learning. They select documents for which two ran-

domly 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 and Uyar 1997; Shahshahani and Landgrebe 1994]. Ghahramani and Jordan [1994] use EM with mixture models to fill in missing feature values.

Experimental Results

This section provides empirical evidence that using a combination of active learning and EM does better than using either individually. 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. 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.

In our experiments an initial classifier was trained with one random document per class. Active learning proceeds as described in Table 1. Experiments were run for 200 active learning iterations. Smoothing parameter λ is 0.5; sharpness parameter β is 3. 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. Random selection is the standard baseline, requiring 133 labeled documents to reach 60% accuracy. Surprisingly, stream-based vote entropy does slightly worse than random, requiring 139 labeled documents. If we use stream-based KL divergence to the mean to more finely measure the disagreement, we improve on random somewhat, requiring 122 documents for 60% accuracy. If we then switch from stream-based sampling to pool-based sampling, each round selecting the document from the pool with the most disagreement, we get a large improvement, needing only 90 documents for pool-based KL divergence to the mean. When we then add density-weighting to the pool-based scheme, to explicitly model the distribution, the results are best, requiring only 81 labeled documents to achieve 60% accuracy. It is interesting to note

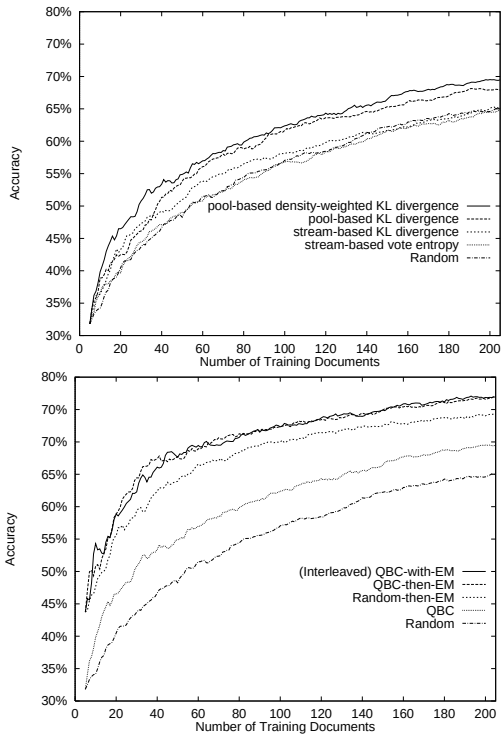

Figure 1: On the top, a comparison of disagreement metrics and selection strategies for QBC shows that density-weighted pool-based KL sampling does better than other metrics. On the bottom, combinations of QBC and EM outperform stand-alone QBC or EM. All have faster learning rates than random example selection. 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.

that the first several documents selected by density-weighted pool-based sampling are usually FAQs for the various newsgroups.

In contrast to earlier work on part-of-speech tagging [Dagan and Engelson 1995], vote entropy does not give good performance 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 rep-

resentative 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 we 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 appropriate. 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. As expected, Random selection and straight QBC give the slowest learning rates: 203 and 131 labeled documents to reach 65% accuracy respectively. Random-then-EM improves upon both; it needs 55 labelings to reach 65%. Interleaved QBC-with-EM is impressive, needing only 38 labelings. QBC-then-EM does slightly better than QBC-with-EM at this accuracy, needing 32 documents—less than 20% of the training data as random, less than 25% of the labeled examples as QBC alone, and 58% of the labeled examples as EM alone.

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.

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 using a more direct approximation of the expected reduction in classification variance across the distribution. We will test the hypothesis that the magnitude of active learning gains is related to the skew of the class priors. We will also further investigate ways of interleaving active learning and EM to achieve a more than additive benefit.

Acknowledgments

We thank Larry Wasserman for help on theoretical aspects of this work. This research was supported in part by the Darpa HPKB program under contract F30602-97-1-0215.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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