Active Learning with Committees: Preliminary Results in Comparing Winnow and Perceptron in Text Categorization

Ray Liere
lierer@research.cs.orst.edu

Prasad Tadepalli
tadepall@research.cs.orst.edu

Department of Computer Science, Oregon State University,
Dearborn Hall 303, Corvallis, OR 97331-3202, USA

Abstract

The availability of vast amounts of information on the World Wide Web has created a big demand for automatic tools to organize and index that information. Unfortunately, the paradigm of supervised machine learning is ill-suited to this task, as it assumes that the training examples are classified by a teacher – usually a human. In this paper, we describe an active learning method based on Query by Committee (QBC) that reduces the number of labeled training examples (text documents) required for learning by 1-2 orders of magnitude.

1. Introduction

The amount of textual information that is available in electronic form has grown exponentially since the advent of the World Wide Web. The Web contains large amounts of textual and other information in electronic form, and it is easy for almost anyone to add even more to this huge, semi-organized collection of information. Unfortunately, for most current learning methods based on the paradigm of supervised learning, this information is of little value unless it is first classified (labeled). This in turn requires human resources, which are expensive and often not readily available. An important characteristic of the Web is that unclassified examples are cheap and abundant, but the labeling is costly. This is not only true of text, but is also true of pictures, sound, and video. Ideally, we need a learning paradigm that can also make effective use of unlabeled examples. However, fully unsupervised learning is too unconstrained and ill-understood at this time to yield useful results in a complex domain. Instead, we have been working on developing methods that will significantly reduce the number of labeled examples needed in order to train the system without incurring unacceptable decreases in prediction accuracy.

We term our method *active learning with committees* (ALC), which is a form of *query by committee* (QBC). In active learning, the learning program exerts some control over the examples from which it learns [Cohn94], resulting in fewer examples being used as compared to supervised learning. Cohn, Atlas, and Ladner developed the theory for an active learning method called *selective sampling* and applied it to some small to moderate sized problems [Cohn94]. Lewis and Gale developed a similar method called *uncertainty sampling*, which is specifically meant for use in text categorization. Their method selects for labeling those examples whose membership is most unclear by using an approximation based on Bayes' Rule. They were able to show one to two orders of magnitude reduction in the number of examples needed to learn to categorize titles from the AP newswire [Lewis94].

QBC is a general approach to active learning which uses the degree of disagreement among all the hypotheses consistent with the data (i.e., the version space) to determine the likely informativeness of an example's label [Freund92, Seung92, Freund97]. Freund, et al. analyzed QBC in detail and showed that the number of examples required in this learning situation is logarithmic in the number of examples required in the passive learning setting [Freund92]. Dagan and Engelson proposed a similar method, termed *committee-based sampling*, for selecting examples to be labeled [Dagan95]. The informativeness of an example (and so the desirability of having it labeled) is indicated by the entropy of the predictions of the various hypotheses in the committee.

Active learning with committees (ALC) is similar to QBC, but only maintains a small finite set of hypotheses which are incrementally updated with training examples. In addition to reducing the number of training examples needed by an order of magnitude as in QBC, by taking a majority vote among the committee members, ALC also allows us to obtain accuracies that exceed those of any of the committee members.

The purpose of this paper is to present results of experiments that demonstrate the effectiveness of ALC in text categorization, using real-world data. We have performed 2 sets of experiments. The first, presented in more detail in [Liere97], looks at 4 different systems which vary in terms of whether or not they use active learning and whether or not they use committees for prediction. All systems use Winnow as the learning algorithm. These experiments indicate that active learning with committees can, as compared to supervised learning with a single learner, result in learning methods that use only 2.9% as many labeled examples but still achieve the same accuracy. The second set of experiments is currently in progress, and we report preliminary results in this paper. This set of experiments compares Winnow and Perceptron learners in both active committees mode and passive learning mode. Somewhat surprisingly, the results indicate that active learning with committees using the Perceptron provides better performance.

2. Active Learning with Committees

Our active learning with committees approach uses a form of QBC for deciding whether or not to see the label and Winnow or Perceptron for updating the hypotheses in the committee. Although it may not be surprising that the choice of good examples allows one to learn with fewer examples, it is not easy to know *how* to select good examples, especially in the presence of noise. Random selection of examples is no better than passive learning.

2.1 Deciding to See the Label

QBC maintains a committee of hypotheses consistent with the labeled examples it has seen so far – a representation of the version space. Each training example is presented to the algorithm unlabeled. An even number of hypotheses (usually 2) are chosen at random, given the attribute values, and asked to predict the label. If their predictions form a tie, then the example is assumed to be maximally informative, the algorithm requests the actual label from the teacher and updates the version space [Freund92, Seung92, Freund97]. QBC offers the benefit of a logarithmic reduction in the number of labeled training examples needed. However, QBC needs to maintain all possible hypotheses consistent with the training data – the version space – in some form [Seung92]. This is the committee. When data is noisy or when the hypothesis space is large or infinite as in text categorization, it is impractical to compactly represent the version space.

Our approach is to use a committee with a small number of hypotheses. Once presented with an unlabeled example, we do the following: two randomly chosen members of the committee are given the unlabeled example and asked to predict the label. If their predictions disagree, then we ask to see the actual label.

2.2 Updating the Hypotheses

After the label is seen, the learners adjust the hypotheses in the committee. Typically, each member of the committee learns individually. We use committees whose members are either all Winnow learners [Littlestone88] or all Perceptron learners [Weiss90]. Both Perceptron and Winnow maintain a hypothesis as a set of weights. Each document class is learned separately. A document is classified positive if the dot product of the weight vector and the feature vector that represents the document is greater than a threshold value. Both Perceptron and Winnow update the weight vector only when a document is misclassified. The main difference between Perceptron and Winnow lies in the way the weights are updated. The Perceptron learner adds (subtracts) a small constant to the weight of each active feature if a positive (negative) example is classified incorrectly. While the Perceptron learner uses additive updating of weights, the Winnow learner uses multiplicative updating.

Actually, "Winnow" refers to a quite large family of algorithms [Littlestone89]. We used a standard version of Winnow – `WINNOW2` in [Littlestone88], with some modifications from [Littlestone91]. This algorithm assigns an initial weight to each attribute and then adjusts those weights during learning, at a rate determined by two parameters α and β. The initial weights must be > 0 and will remain > 0 (due to the updating being multiplicative). This therefore limits the patterns that can be learned to those that can be learned using a separating hyperplane defined by all weights being positive. That is, the learning is based on those attributes that *are* in the document. This seems intuitively how a human classifies documents – by the words that are in the document (versus by those that are not). Our second set of experiments bears this out.

Once the learning process has been completed, the committee needs to make predictions for previously unseen test examples. We experimented with both using a single member of the committee and the majority vote and found that majority voting gives better test set accuracy.

3. Experimental Setup

3.1 Test Bed

All of our experiments were conducted using the titles of newspaper articles from the Reuters-22173 corpus [Reuters]. The Reuters corpus is a collection of 22,173 Reuters newswire articles ("documents") from 1987. Each article has been assigned to any number of categories, including none. There are 21,334 unique tokens in titles, and there are 679 categories. The Reuters-22173 corpus contains formatting errors, misspellings, and garbled/ missing/ reordered sections. This is good, in that it is typical of most real-world data.

A preprocessing step converts the raw data in the corpus into labeled examples. The presence or absence of each possible token in a particular article is used to construct the boolean attribute vector. Similarly, the labels indicate, for each category, whether or not that article is in that category. We do not do any prefiltering of the data to clean it up or to reduce its dimensionality.

3.2 Repeated Trials

A variety of approaches have been utilized in previous research using the Reuters corpus [Hayes90, Lewis91, Apte94]. Normally researchers use one of 3 standard corpus setups, and so it is predetermined which articles will be used for training, which will be used for testing, and which will not be used at all. We are mainly interested at this point in comparisons among various versions of our learning systems. Therefore, we compared their performance for categories most likely to have ample training data. We used the 10 most frequently occurring topic categories, as listed in [Lewis91], for our experiments. We performed repeated trials for each category, using randomly chosen training-test splits. We used the entire corpus, and split it into 21,000 training examples and 1,173 test examples. We used titles only for our tests.

We analyzed the data gathered while learning was occurring as well as the final results. The main results that we examined were the number of labeled training examples used, accuracy (fraction of documents correctly classified), elapsed processor time used, space used during learning, and space used during prediction. Our main statistical tool was the repeated measures analysis of variance test, hereafter "anova". The null hypothesis is that all of the systems actually perform the same on the average. We performed single factor tests, with the factor being the system.

How one initializes the weights is important in several learning algorithms, including Winnow and Perceptron. We compute initial positions of the hyperplanes so that they approximately bisect the space of all *possible* data points. We can compute this knowing only the number of attributes in the data. We chose this method because it allows the learners to start the learning process at a reasonable location, and it does not use any information about any actual data values. Individual committee members are randomly initialized to slightly different hyperplanes so that they represent different initial hypotheses. A difficulty encountered in initializing a committee of learners is that one wants the learners to be different (otherwise they will always agree), but not too different (or they will too often disagree), and not have the initialization in any way tailored to the specific category.

We used a committee with 7 members. This value was determined empirically. We found that using larger committees increased the space and time used by the algorithm without any significant increase in accuracy, since large amounts of duplicated effort occurred. We also found that using a very small committee resulted in the committee becoming very sensitive to the characteristics of 1 or perhaps 2 of the members, and the behavior of the committee began to approach that of a single member.

3.3 Experiment #1 and Results

In this experiment, we examined the performance of 4 different learning systems which vary in terms of whether or not they use active learning and whether or not they use committees for prediction. All systems use Winnow as the learning algorithm. Please see [Liere97] for further details.

The systems are:

- *active-majority*: the learner is a committee of Winnows which uses disagreement between two randomly chosen members to determine which labels to obtain from the teacher. Prediction is made by that same committee, using majority rule.
- *passive-majority*: the learner is a committee of Winnows which passively accepts all labels from the teacher. Prediction is made by that same committee, using majority rule.
- *active-single*: the learner is a committee of Winnows which uses disagreement between two randomly chosen members to determine which labels to obtain from the teacher. However, in the prediction phase, only a single member of the committee is used. For all predictions in a particular trial, a committee member chosen at random makes the predictions.
- *passive-single*: the learner is a single Winnow which passively accepts all labels from the teacher. Prediction is by that same Winnow. This can be thought of as the "base case" – a single supervised learner and predictor.

Figure 1 shows elapsed processor time as a function of the number of training examples used for each of the 4 systems. There is one dot on the graph for each trial, and a box encloses the dots for each system. Note that the dots form very tight clusters. Figure 1 dramatically illustrates that, in these experiments, the differences among the systems in terms of both the number of labeled examples used and the elapsed processor time were quite large. Figure 1 also shows that the variation in the behavior within each system, for both the number of training examples used and the elapsed processor time, was quite small. Anova indicates that which system one uses has a significant effect on both the number of training examples used ($p < 0.0001$) and on the elapsed processor time ($p < 0.0001$).

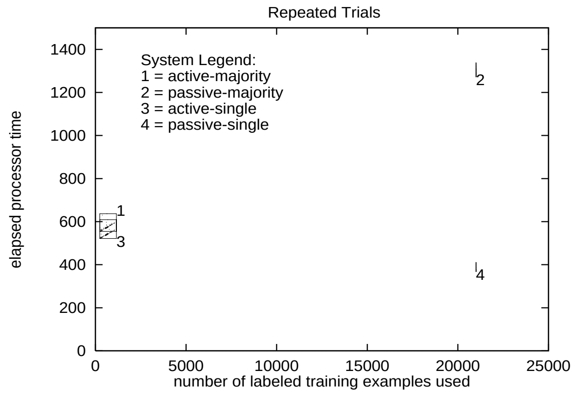

Figure 1: Elapsed Time versus Number of Training Examples Used

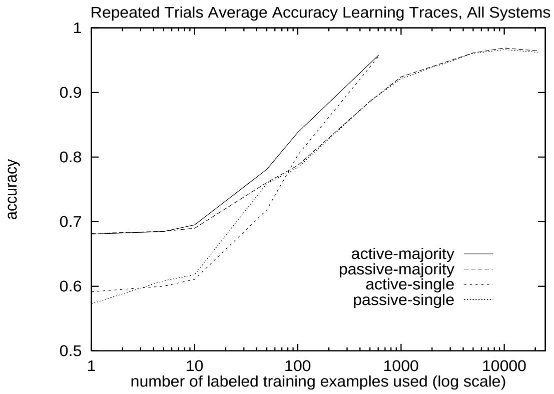

Figure 2: Average Accuracy by System

Figure 2 shows the average accuracy for each of the 4 systems as a function of the number of training examples used. This is a learning trace, showing how accuracy varies for each system as it learns. (Note the use of a log scale). We can see that the systems employing active learning use many fewer examples than those using supervised learning, which is consistent with the results in Figure 1. However, Figure 2 also shows that the 4 systems end up with very similar final accuracies – in the 95-96% range. Anova indicates that the system used does *not* have a significant effect on final accuracy. The fact that the systems are all about the same in terms of final accuracy justifies our looking at other system characteristics (such as number of training examples used and elapsed processor time) as metrics on which to base our comparison of the systems. One can also see in Figure 2 that active-majority is, during the learning process, more accurate than the other systems.

Of the systems tested, active learning with committees (the active-majority system) is the best approach when one has a limited supply of labeled examples. This approach achieves accuracies that are the same as those obtained by the other systems, but uses only 2.9% as many training examples as the supervised learners. And while all 4 systems reached essentially the same accuracy level, the path that each took to get there was different. Because it has the best average accuracy as learning progresses, the active-majority system is also the best one for applications in which learning is halted (and prediction commences) after a certain period of elapsed time, such as when interactive processing is occurring with a human being.

3.4 Experiment #2 and Results

This set of experiments is currently in progress, and we report preliminary results in this paper. This set of experiments compares Winnow and Perceptron learners, for active learning with committees, and for single learner supervised learning.

The 4 comparison systems are:

– *active-majority-winnow*: the learner is a committee of Winnows which uses disagreement between two randomly chosen members to determine which labels to obtain from the teacher. Prediction is made by that same committee, using majority rule.

– *active-majority-perceptron*: same as immediately above, but the learner is a committee of Perceptrons

– *passive-single-winnow*: the learner is a single Winnow which passively accepts all labels from the teacher. Prediction is by that same Winnow.

– *passive-single-perceptron*: same as immediately above, but the learner is a Perceptron

Figure 3 shows elapsed processor time as a function of the number of training examples used for each of the 4 systems. (The graph design is the same as for Figure 1, earlier). Note again that the dots form very tight clusters, and so we also find in this second set of experiments that differences among the systems in terms of both the number of labeled examples used and the elapsed processor time are quite large and that the variation in the behavior within each system, for both the number of training examples used and the elapsed processor time, is quite small. Anova indicates that which system one uses has a significant effect on both the number of training examples used ($p < 0.0001$) and on the elapsed processor time ($p < 0.0001$).

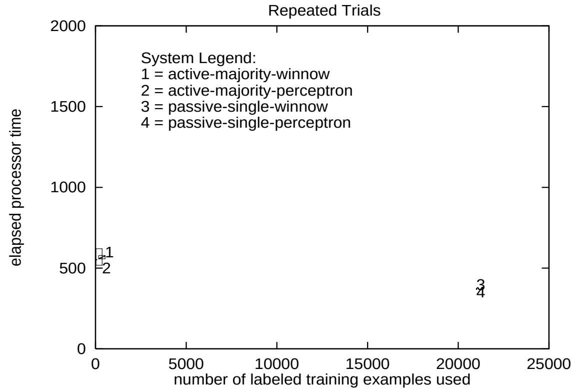

Figure 3: Elapsed Time versus Number of Training Examples Used

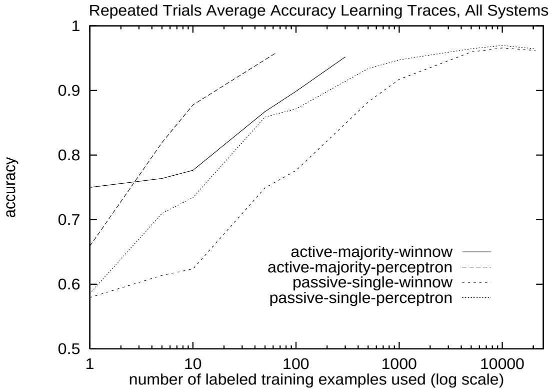

Figure 4: Average Accuracy by System

Figure 4 shows the average accuracy for each of the 4 systems as a function of the number of training examples used – it is of the same design as Figure 2. Again, we see that systems employing active learning use many fewer examples than those using supervised learning, consistent with Figure 3. Figure 4 also shows that the 4 systems end up with very similar final accuracies – in the 95-96% range. One can also see in Figures 3 and 4 that active-majority-perceptron is the best system, since during the learning process, it is more accurate than the other systems, and since it uses far fewer examples than the other systems. In fact, it uses less than 1% of the total number of examples used by the supervised learners.

In all honesty, we were puzzled by these results. We had expected Winnow to be the better performer. We investigated several possible explanations for our results. One thought was that the limitation (see earlier) on weight values that Winnow can assume had put it at a disadvantage when matched against the Perceptron, which can learn any linearly separable pattern. We performed two additional experiments. In one, we allowed Perceptron learners to learn until they reached 100% accuracy (or the maximum accuracy possible), using multiple epochs and α/β adjustments. Very few weights (typically less than 3%) were in fact negative, and most of them were only slightly negative ([-0.05,0.0] in a range of [-0.05,+0.80]). In another experiment, we coded and tested two different versions of so-called "balanced Winnow" algorithms – Winnows that can also learn any linearly separable pattern. These experiments have so far been inconclusive, in that it has been very difficult to initialize these committees so that they do not quickly degrade into always saying "yes" or always saying "no". We feel that this is at least in part due to the very high dimensionality (both methods effectively double the dimensionality of the input data).

But perhaps the real reason why the Perceptron does better than Winnow has to do with the sparseness of the instance space and the sparseness of the target rules. The instance space that we are dealing with is very sparse (most titles contain 2-10 tokens, out of 16,600 attributes), and the target rules are also sparse (most weights are close to 0.0). Winnow works best if the target rules are sparse but the instance space is not. Perceptron works best if the target rules are not sparse, but the instance space is. Our experiment #2 has both sparse instance space and sparse weights, and so could have gone either way. We plan to investigate this further by using the full text and also by artificially making the instances less sparse by adding noise.

4. Conclusions

The ease with which documents can be added to the World Wide Web presents many challenges, including automatically categorizing and indexing them. The current approaches of supervised machine learning are not suitable to this task because they are unable to exploit unclassified examples. Active Learning with Committees promises to reduce the number of labeled training examples needed by an order of magnitude or more without any significant loss in accuracy. The Query by Committee approach has some nice theoretical properties. Active Learning with

Committees adapts QBC to noisy situations and large hypothesis spaces. Some of the future problems include scaling it to full-text categorization and making it less sequential so that documents which are deemed informative may be classified off-line. We also foresee its application to multi-media, including pictures, sound, and video on the Web.

5. Acknowledgements

This research was partially supported by the National Science Foundation under grant number IRI-9520243. We thank Tom Bylander and David Lewis for suggesting an explanation of Winnow and Perceptron behaviors. The availability of the Reuters-22173 corpus [Reuters] and of the |STAT Data Manipulation and Analysis Programs [Perlman] has greatly assisted in our research.

6. References

[Apte94] Chidanand Apté, Fred Damerau, Sholom M. Weiss, Automated Learning of Decision Rules for Text Categorization, *ACM TOIS* 12(2):233-251, July 1994

[Cohn94] David Cohn, Les Atlas, Richard Ladner, Improving Generalization with Active Learning, *Machine Learning* 15(2):201-221, May 1994

[Dagan95] Ido Dagan, Sean P. Engelson, Committee-Based Sampling for Training Probabilistic Classifiers, in *Proceedings: ICML95*, 1995, p. 150-157

[Freund92] Yoav Freund, H. Sebastian Seung, Eli Shamir, Naftali Tishby, Information, Prediction, and Query by Committee, in *Proceedings: NIPS92*, p. 483-490

[Freund97] Yoav Freund, H. Sebastian Seung, Eli Shamir, Naftali Tishby, Selective Sampling Using the Query by Committee Algorithm, *Machine Learning* 28(2-3):133-168, August-September 1997

[Hayes90] Phillip J. Hayes, Peggy M. Andersen, Irene B. Nirenburg, Linda M. Schmandt, TCS: A Shell for Content-Based Text Categorization, in *Proceedings of the 6th IEEE CAIA*, 1990, IEEE, p. 320-326

[Lewis91] David D. Lewis, Representation and Learning in Information Retrieval, Ph.D. Thesis, University of Massachusetts at Amherst, COINS Technical Report 91-93, December 1991

[Lewis94] David D. Lewis, William A. Gale, A Sequential Algorithm for Training Text Classifiers, in *Proceedings: SIGIR'94*, p. 3-12

[Liere97] Ray Liere, Prasad Tadepalli, Active Learning with Committees for Text Categorization, in *Proceedings: AAAI-97*, p. 591-596

[Littlestone88] Nick Littlestone, Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm, *Machine Learning* 2(4):285-318, 1988

[Littlestone89] Nicholas Littlestone, Mistake Bounds and Logarithmic Linear-Threshold Learning Algorithms, University of California at Santa Cruz, UCSC-CRL-89-11, March 1989

[Littlestone91] Nick Littlestone, Redundant Noisy Attributes, Attribute Errors, and Linear-Threshold Learning Using Winnow, in *Proceedings: COLT'91*, p. 147-156

[Perlman] Gary Perlman, |STAT version 5.4, software and documentation, available from: `ftp:/archive.cis.ohio-state.edu/pub/stat/`

[Reuters] Reuters-22173 corpus, a collection of 22,173 indexed documents appearing on the Reuters newswire in 1987; Reuters Ltd, Carnegie Group, David Lewis, Information Retrieval Laboratory at the University of Massachusetts; available via ftp from: `ciir-ftp.cs.umass.edu:/pub/reuters1/corpus.tar.Z`

[Seung92] H. S. Seung, M. Opper, H. Sompolinsky, Query by Committee, in *Proceedings: COLT92*, p. 287-294

[Weiss90] Sholom M. Weiss, Casimir A. Kulikowski, *Computer Systems that Learn*, Morgan Kaufmann, 1990, p. 82-87