Active Learning using Adaptive Resampling

Vijay S. Iyengar
IBM Research Division
T.J. Watson Research Center
P.O. Box 218, Yorktown
Heights, NY 10598, USA

vsi@us.ibm.com

Chidanand Apte
IBM Research Division
T.J. Watson Research Center
P.O. Box 218, Yorktown
Heights, NY 10598, USA

apte@us.ibm.com

Tong Zhang
IBM Research Division
T.J. Watson Research Center
P.O. Box 218, Yorktown
Heights, NY 10598, USA

tzhang@watson.ibm.com

ABSTRACT

Classification modeling (a.k.a. supervised learning) is an extremely useful analytical technique for developing predictive and forecasting applications. The explosive growth in data warehousing and internet usage has made large amounts of data potentially available for developing classification models. For example, natural language text is widely available in many forms (e.g., electronic mail, news articles, reports, and web page contents). Categorization of data is a common activity which can be automated to a large extent using supervised learning methods. Examples of this include routing of electronic mail, satellite image classification, and character recognition. However, these tasks require labeled data sets of sufficiently high quality with adequate instances for training the predictive models. Much of the on-line data, particularly the unstructured variety (e.g., text), is unlabeled. Labeling is usually a expensive manual process done by domain experts. Active learning is an approach to solving this problem and works by identifying a subset of the data that needs to be labeled and uses this subset to generate classification models. We present an active learning method that uses adaptive resampling in a natural way to significantly reduce the size of the required labeled set and generates a classification model that achieves the high accuracies possible with current adaptive resampling methods.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.5.1 [**Pattern Recognition**]: Models; H.2.8 [**Database Management**]: Database Applications—*data mining*

General Terms

Data mining, machine learning, classification, active learning, adaptive resampling

1. INTRODUCTION

Supervised learning methods are being used to build classification models in various domains like finance, marketing, and healthcare [5]. Classification techniques have been developed within several scientific disciplines, including statistics, pattern recognition, machine learning, neural nets and expert systems [30]. The quality and the quantity of training data used by these supervised methods is an important factor in the prediction accuracy of the derived models. In many applications, getting data with the class labels is difficult and expensive since the labeling is done manually by experts. A frequently cited example is electronic mail routing based on categories. Training data is usually obtained by manually labeling a number of instances of mail. Another such example is categorizing web pages based on content.

One approach to solving this problem is to select the data that need to be labeled such that a small amount of labeled training data suffices to build a classifier with sufficient accuracy. Random sampling is clearly ineffective since the various classes can have very skewed distributions in the data and instances of the infrequent classes can get omitted from the random samples. Stratified sampling [8] is a method developed to address this problem with random samples. The unlabeled data is partitioned based on the attributes of each instance in the data. Sampling is done separately from each partition and can be biased based on the expected difficulty in classifying the data in each partition. However, it becomes more difficult to generate these partitions for high dimensional data and it is not clear how to effectively apply this approach on data typically seen in many real life applications.

Active learning is a term coined to represent methods where the learning algorithm assumes some control over the subset of the input space used in the modeling [9, 10]. In this paper, active learning will mean learning from unlabeled data, where an oracle can be queried for labels of specific instances, with the goal of minimizing the number of oracle queries required. Active learning has been proposed in various forms [2, 10, 11, 12, 17, 23, 24, 27]. We will discuss in more detail the earlier works in active learning related to the approach used in this paper.

One approach to active learning is *uncertainty sampling* in which instances in the data that need to be labeled are iteratively identified based on some measure that suggests that the predicted labels for these instances are uncertain. Vari-

ous methods for measuring uncertainty have been proposed. In [22], a single classifier is used that produces an estimate of the degree of uncertainty in its prediction. An iterative process then selects some fixed number of instances with maximum estimated uncertainty for labeling. The newly labeled instances are added to the training set and a classifier generated using this larger training set. This iterative process continues until the training set reaches a specified size. This method is generalized in [21] by using two classifiers, the first one to determine the degree of uncertainty and the second one to do the classification. In this work, a probabilistic classifier was chosen for the first task based on efficiency considerations and C4.5 rule induction was chosen for the second task.

Another related approach is called *Query by Committee* [27, 16]. In one version of the query by committee approach two classifiers consistent with the already labeled training data are randomly chosen. Instances of the data for which the two chosen classifiers disagree are then candidates for labeling. The emphasis here has been to prove theoretical results about this approach.

Adaptive resampling methods are being increasingly used to solve the classification problem in various domains with high accuracies [15, 7, 28]. In this paper, we use the term *adaptive resampling* to refer to methods like boosting that adaptively resample data biased towards the misclassified points in the training set and then combine the predictions of several classifiers. Various explanations have been put forth for the classification accuracies achieved by these techniques [26, 18]. Adaptive resampling methods like boosting are also useful in selecting *relevant* examples even though their original goal was to improve the performance of weak learning algorithms [14]. The application of boosting to selective labeling has been suggested in [14] without algorithmic details or experimental results. A related application of boosting to select a subset of labeled instances for nearest neighbor classifiers has been explored in [15]. The closest related work [1] combines the *Query by Committee* approach with bagging and boosting techniques. In this paper we use a more general formulation that separates the two roles for a classifier in such approaches. This allows us to plug in different classifiers (including an oracle) for one of these roles and gain additional insight on factors influencing the results achieved. Other differences between our method and [1] relate to practical aspects in the application that impact the computational requirements and will be discussed later in the paper.

This paper applies adaptive resampling to the active learning task in a direct way that will be described in the next section. The goal is to retain some of the advantages of adaptive resampling methods, e.g., accuracy and robustness of the generated models, and combine it with a reduction in the required size of the labeled training set. Comparisons will also be made between using either one or two classifiers in the adaptive resampling framework [21]. Experimental results using benchmarks from various domains are presented in the paper to illustrate the the sizes of the labeled training sets needed to get adequate classification accuracy.

2. DESCRIPTION OF OUR METHOD

Adaptive resampling (e.g., [15, 28]) selects instances from a labeled training set with the goal of improving the classification accuracy. The selection process adapts by biasing in favor of those instances that are misclassified by the ensemble of classifiers generated. We explore a direct application of this framework to choose which of the unlabeled instances should be labeled in an active learning task. Since the actual labels are unknown for these instances in an active learning task, guessed labels generated by a classifier will be used instead.

Method ALAR (Input: Unlabeled data U,
      Output: Labeled training set L,
      Output: Classifier C)
Choose initial subset to start process
1 Select an initial subset $S_0 \in$ U.
 Label instances in S_0. Remove S_0 from U and add it to L.
 A subset of instances selected for labeling in each phase
2 **For** each phase p
3   Guess labels G for each instance in U
    using classification method M1.
    Multiple rounds of adaptive resampling
4   Use adaptive resampling on training set L
    using classification method M2 to generate
    an ensemble E of classification models.
    Select subset of instances to add to training set
    for use by adaptive resampling in the next phase
5   **If** not last phase
6   Select subset $S_p \in$ U using weights W
    calculated for each instance in U using G and E.
   Remove S_p from U and add it to L.
 Build combined classifier using voting
7 Combine the ensemble E of classification models
 to form a resultant classifier C.
end ALAR

Figure 1: Description of Active Learning using Adaptive Resampling (comments are *italicized*)

Consider a more detailed description of the method (ALAR) given in Figure 1. It is assumed that apart from the unlabeled data U provided to the method, an *expert* is available to label any selected instance in U. The method produces as output a classifier C and a selectively labeled training set L that might have other uses (e.g., for use by another classifier).

Instances are selected from the unlabeled data U for labeling in an iterative process. The initial subset S_0 is typically chosen at random. Instances in S_0 are labeled by the expert and moved from U to the labeled training set L (statement 1). Additional instances from U will be labeled and added to L in phases. In each phase, the labeled training set L is used by a classification method M1 to guess the labels G for the unlabeled instances in U (statement 3). The set L with the instances labeled so far is used in an adaptive resampling framework using a classification method M2 to generate an ensemble E of classification models (statement 4). Many variations for adaptive resampling have been proposed and they differ in the details of weighting function for

resampling and the classification method used. The experimental results in this paper were generated using decision trees for the classification method M2. The resampling was done using the normalized version of the following weighting function w(i) for each instance i in L [28]:

$$w(i) = (1 + error(i)^3) \qquad (1)$$

where error(i) is the cumulative error for instance i over all the classification models in the ensemble E.

The ensemble E of classification models is used with the guessed classes G for the unlabeled data to select more instances in U for labeling in the next phase (statement 6). Intuitively, the weights W for selecting any instance in U for labeling should be biased towards those which are misclassified in the ensemble E assuming the validity of the guessed class labels G. In our experiments, we use Equation 1 again to compute the weights W, but with the cumulative error being calculated using the guessed class labels G as reference. A set of instances S_p is selected in each phase by sampling using the normalized version of weights W.

Typically, the iterative addition of instances from U to the labeled set L could continue until a specified size of L has been reached or the model quality improvements taper off. The final classifier C is generated by combining the classification models in the ensemble E (statement 7). We explore a couple of variations in the generation of C. In the first case, all the classification models in E are combined. In the second case, once the labeled training set L is complete, a new set of classification models is generated using adaptive resampling with this complete set L (earlier models in E are discarded). The second case corresponds to using our method to generate a labeled training set L and then using the adaptive resampling method with L. In our experiments, we use unweighted voting across the set of classification models being combined to produce the final classifier C [7, 28].

Two variations of the ALAR method will be considered in the experiments discussed in the next section. In the first approach, refered to as ALAR-vote-E, we combine (using unweighted voting) the ensemble of classification models E available in each phase for use as the classification model M1. This approach takes advantage of the reported effectiveness of voting methods (e.g., [15]) in providing guessed labels. In the second approach, refered to as ALAR-3-nn, two distinct classification methods are utilized. A nearest neighbor method (3-NN) is used for classification method M1. In both approaches decision trees are used for classification method M2. The comparison of the performance of these two approaches is interesting given earlier comparisons between one and two classifier methods (e.g., [21]).

Other important parameters that can be varied in the method in Figure 1 are the number of phases, number of points to be selected for labeling in each phase and the number of rounds of adaptive resampling with the training set of each phase. The values used for these parameters in our experiments will be given in the next section along with other experimental details.

3. EXPERIMENTS

This section presents the results of applying our method to benchmarks in various domains. The first benchmark *internet-ads* we will consider is based on an application to identify images that are Internet advertisements [6]. An application to remove advertisements after identification was evaluated using this benchmark by its donor in [19]. Three of the 1558 features encode the geometry of the image. Most of the remaining binary features capture occurrences of phrases in the URL, the anchor text, and text near the anchor text. In this paper, only the 2359 records in the benchmark without any missing data are used. The original paper [19] using this data reported results using the accuracy measure. The skewed distribution of the two classes *ad, nonad* leads us to use instead the usual information retrieval measures of recall and precision for the more infrequent class *ad*. All experiments with this benchmark are done using 10-fold cross validation.

In the first experiment we will use random sampling to create training sets of various sizes. For each training set created, two types of classification models are constructed and evaluated against the test set. The first type of model is a decision tree constructed using the tree package DMSK [29]. The second type of model is created using adaptive resampling of the training set with 100 DMSK trees. Figure 2 shows the results averaged over ten experiments for each partition in the 10-fold cross validation. The arithmetic mean of precision and recall is the metric displayed. The results obtained for the single tree are comparable to the results presented in [19]. The quality of precision/recall degrades substantially for the single tree from 89.4% to 71.3% when the randomly chosen training set size is reduced by a factor of ten to 212. On the other hand, adaptive resampling with the randomly chosen subsets (AR-random) is more robust. The precision/recall metric for AR-random with the entire training data is 92.3%, which is better to begin with. When the training set is cut in size randomly by a factor of ten the metric for AR-random degrades to 84.8%. Many of the earlier works in active learning give comparisons with classifiers like the single tree case shown in Figure 2. However, with the prevalence and success of adaptive resampling methods now, it is more interesting to compare the accuracy of active learning methods using AR-random as the baseline [1].

The improvement in prediction accuracy by using the ALAR method over AR-random is shown in Figure 3. The AR-random performance curve is repeated for comparison. The curves marked ALAR were achieved by using the ALAR method of Figure 1 with the following set of parameters. A total of 4 phases (after the initial addition of S_0) were used with equal number of instances being labeled in each phase. In each phase 25 rounds of adaptive resampling was done with the labeled training set available at that point. However, for the last phase after all the additions to the labeled training set this was increased to 100 rounds of adaptive resampling. The combined classifier was obtained by voting over all the 200 trees in the ensemble. This set of parameters was used for all the experiments in the paper except when noted otherwise.

The curves ALAR-vote-E and ALAR-3-nn depict the results achieved by two variations of the ALAR method. The
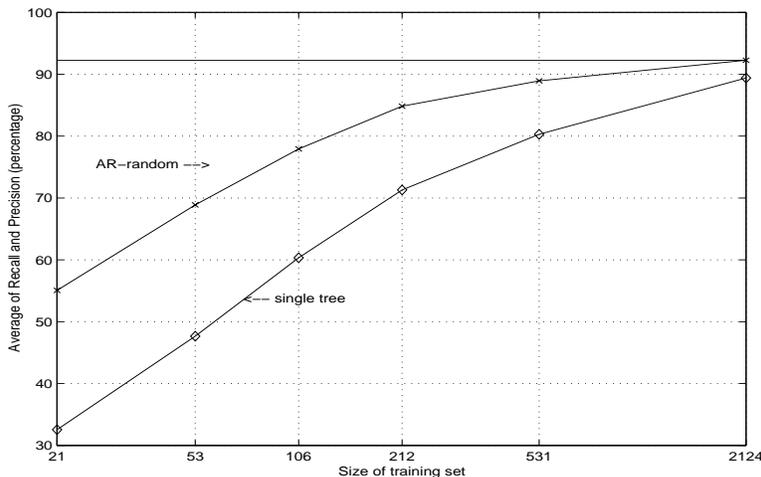
Figure 2: **Results using random sampling on benchmark** *internet-ads*

ALAR-vote-E curve was achieved by using the unweighted majority vote amongst the ensemble of models E for classification method M1. The ALAR-3-nn curve was achieved by using 3-NN as the classification method M1. The results in Figure 3 indicate that there is a very slight loss of accuracy using ALAR-vote-E and ALAR-3-nn even when the training set size is reduced by a factor of four. When further reductions are made in the size of the labeled training set, the accuracy of both methods (ALAR-vote-E and ALAR-3-nn) degrades, though it continues to remain better than AR-random. For this benchmark, ALAR-vote-E performs slightly better than ALAR-3-nn for most of the training set size range.

Another interesting curve plotted in Figure 3 is called ALAR-oracle. This curve is achieved by using an oracle for classification method M1. Obviously, this is not a practical solution since the labels for instances in U will not be known. However, the ALAR-oracle curve can be used to assess the impact of the accuracy of the classification methods used for M1 (e.g., 3-NN and vote-E) on the ALAR method. The gap between ALAR-oracle and ALAR-vote-E/ALAR-3-nn widens as the allowed size of the labeled set is reduced. This is caused in part by the quality of guesses in both ALAR-vote-E and ALAR-3-nn getting worse as the size of the labeled set available to them decreases. All the ALAR results can be impacted by changing the parameters for the ALAR method (e.g., number of phases, number of instances added for labeling in each phase). We have experimented with these parameters to some extent, but will use the same set of parameter values across all the benchmarks.

The next benchmark we will consider is *satimage* from the UCI Repository [6]. This benchmark contains spectral values for pixels in a satellite image (36 attributes) and the goal is to predict the soil type (6 classes). The given training set has 4435 points and the test set has 2000 points. The ALAR method was applied with the same set of parameters as described earlier and the results averaged over 10 experiments (on the given test set) are plotted in Figure 4. As before the AR-random curve is used as the baseline and the goal for accuracy is that achieved by AR-random (average error =

8.54%, $\sigma = 0.17\%$) with the entire training set of size 4435. Both ALAR-vote-E and ALAR-3-nn achieve comparable accuracy with only 2217 labeled instances. With 2217 labeled instances ALAR-3-nn achieves average error = 8.83%, $\sigma = 0.19\%$, and ALAR-vote-E achieves average error = 8.67%, $\sigma = 0.34\%$. Interestingly, both ALAR-3-nn and ALAR-vote-E achieve accuracy similar to ALAR-oracle for much of the training set size range.

The ALAR method (refer Figure 1) produces a labeled training set L of the specified size in addition to the classifier C. We explored the use of this labeled training set with this benchmark. Three different classifiers were used to compare three training sets: a ALAR-3-nn generated labeled set of size 2217, a random subset of size 2217, and the entire training set of size 4435. The three classifiers were 5-NN, adaptive resampling using 100 DMSK trees, and a single DMSK tree. Table 1 presents the average percentage errors and standard deviation (in parenthesis) over ten experiments. For this benchmark, the smaller labeled set produced by ALAR-3-nn can be used by these three classifiers to produce fairly accurate models when compared to the results using the entire training set. However, further investigations are needed to determine whether, in general, the labeled sets are useful with other classifiers.

The next benchmark is *letter-recognition* from the UCI Repository [6]. The 16 attributes capture statistical moments and edge counts for the english alphabets in various fonts with the goal of determining the displayed alphabet (26 classes). The benchmark specifies a training set with 16K instances and a test set with 4K instances. The results of applying the ALAR method are shown in Figure 5. Both ALAR-3-nn and ALAR-vote-E achieve the accuracy goal with only 8000 labeled instances.

The last benchmark used is the Mod-Apte split of the *Reuters* data set available from [20]. Only the top ten categories are considered. For each of them we solve the binary classification problem of being in or out of that category. We used the notion of *information gain* [31] to select a set of 500 attributes for each of the ten binary classification problems.
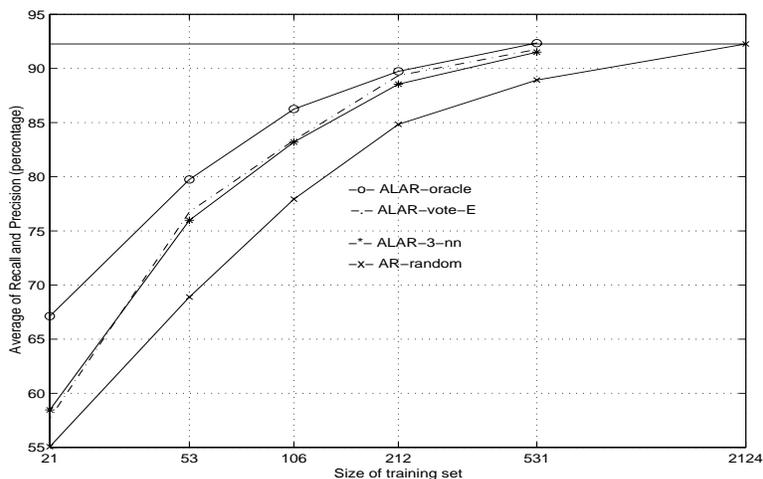
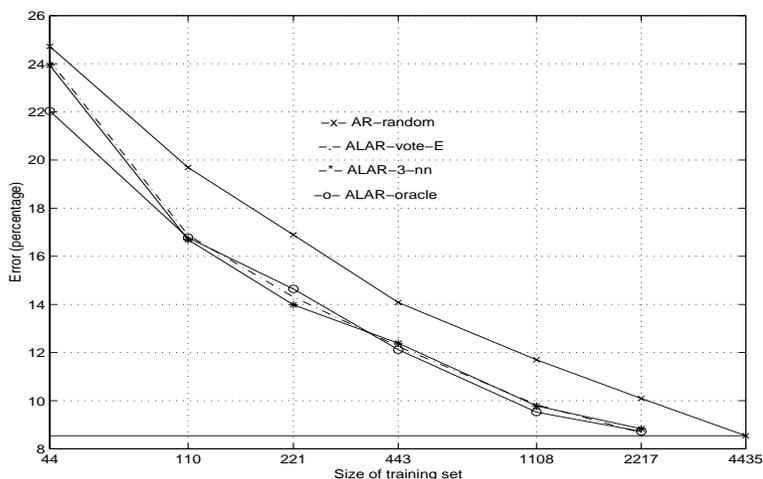**Figure 3: Results using ALAR methods on benchmark *internet-ads***

Figure 4: Results using ALAR methods on benchmark *satimage*

This feature selection method requires labels and hence is not applicable for truly unlabeled data [21]. Also, a reduction in the size of the labeled set in this experimental framework does not translate to a corresponding reduction in the labeled set needed for the Reuters classification problem. However, this experimental framework has been used in earlier works [24]. An internally available decision tree package customized for text applications was used for this benchmark. As is customary with this benchmark, we use the micro-average measure [3], in which the confusion matrices for the ten categories are added and overall precision and recall computed. Ten random runs were performed and the micro-average of the arithmetic mean of recall and precision is given in Figure 6. There is only a slight degradation in the accuracy with just 960 labeled instances using either ALAR-vote-E or ALAR-3-nn method.

4. DISCUSSIONS

The experimental results in the previous section indicate that the ALAR-3-nn and ALAR-vote-E methods perform similarly on those benchmarks. Clearly, there is no ev-

idence in our experiments to justify the added computational cost of a separate classification method like K-NN for M1. ALAR-vote-E is a more natural and direct way to apply adaptive resampling to the task of active learning when compared to ALAR-3-nn. On some of the benchmarks (*internet-ads, reuters*) the ALAR method using the oracle does significantly better than ALAR-vote-E, especially for the smaller sizes of the training set. Part of the explanation for this is that the quality of the guesses get worse as the size of the labeled training set decreases. However, variations in the behavior across the various benchmarks require further investigation.

It is hard to directly compare the results obtained using the ALAR methods with those obtained by earlier approaches to active learning. Clearly, the performance of any active learning method depends heavily on the benchmark and its usage. Earlier works on active learning also report significant reduction in the required size of labeled training set. However, the baseline target accuracy is chosen differently in each case. For example, in [21] the baseline target is

Classifier used	Random subset (size 2217)	ALAR-3-nn generated subset (size 2217)	Entire training set (size 4435)
5-NN	10.88 (0.47)	9.79 (0.17)	9.65
adaptive resampling using trees	10.09 (0.38)	8.63 (0.23)	8.54 (0.17)
Single tree	16.33 (0.76)	15.29 (0.7)	14.8

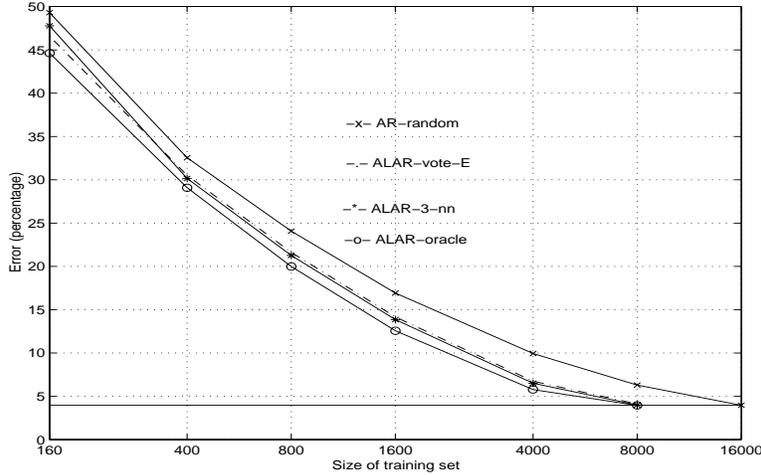Table 1: Use of ALAR-3-nn generated subset with some classifiers and comparisons

Figure 5: Results using ALAR methods on benchmark *letter-recognition*

set by the accuracy achieved by C4.5 rules on the full labeled set. As we have seen in Figure 2 adaptive resampling classification methods can significantly improve the baseline target over single tree classifiers. This has also been pointed out in the work in [1] which includes boosted results in the baseline.

Adaptive resampling with trees is a computationally intensive process and the ALAR method inherits this computational complexity if decision trees are chosen for the classification method M2. The values for the parameters of the ALAR method were chosen in our experiments based on computational complexity and accuracy considerations. Instances are chosen for labeling and added to the training set in phases. Each phase needs to have enough rounds of adaptive resampling to train the ensemble of classifiers adequately to the training set for that phase. Adding only one instance in each phase as in [1] would lead to too many phases and too many rounds of adaptive resampling. Hence, in our experiments the total number of rounds of adaptive resampling, which impacts the computational cost, was chosen to be comparable to earlier usage (e.g., [15]). Having chosen this, the number of phases is determined based on trading off having enough rounds per phase for adaptive resampling versus having enough phases with fine grain control for adding instances for labeling.

As mentioned above, computational considerations lead us

to select multiple instances for labeling in each phase. This opens up the issue of how these instances are chosen. One approach would be to extend the greedy method of picking one instance in [1] to picking multiple instances with the largest weights (W in Figure 1). Instead, we have used a randomized method by creating a probability function using the selection weights (Equation 1) and using it to pick multiple instances without replacement. The comparison for the benchmark *satimage* is given in Figure 7. For this benchmark the probabilistic method (ALAR-vote-E) performs better than the greedy method (Greedy-E) for smaller training set sizes. A plausible explanation is that picking multiple instances in a greedy fashion may be including more instances that are redundant for the modeling. Combining these methods to improve the selection process needs to be explored further.

In practice, the active learning process would be stopped by detecting diminishing improvement in the quality of the models being built. Convergence detection has been studied for the case of random sampling by estimating the slope of the learning curve [25]. The learning curve may not be well behaved in the active learning case making this task more complicated. This also makes the more general problem of determining a good schedule for adding labeled points harder than the random sampling case [25].

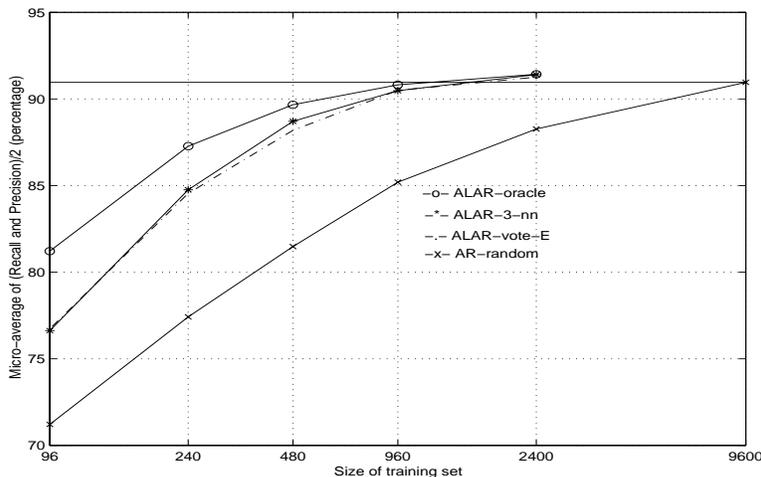There are other variations of this method still to be ex-

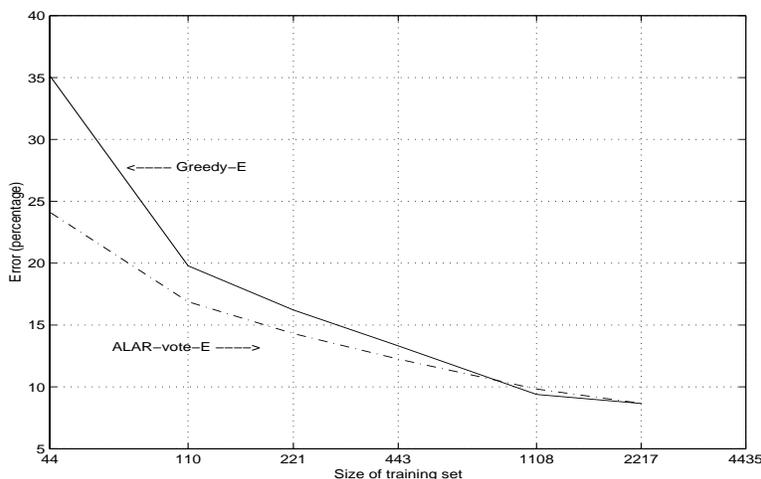Figure 6: **Results using ALAR methods on the top ten categories of the benchmark** *reuters*

Figure 7: **Comparing greedy and probabilistic selection methods on benchmark** *satimage*

plored. Use of simpler classification methods for M2 will be explored in future work. A related problem with the use of decision trees not addressed in this paper is that of attribute selection for unlabeled data [21]. Another variation to be explored is in the function (e.g., Equation 1) used for adaptive resampling relating importance of selecting an instance to some measure of error. The adaptive resampling literature has explored this and the related subject of overfitting any noisy labels in the training set [4, 13, 18]. The concern over overfitting of noise labels is not directly applicable in the active learning context since the error measure is computed using guessed labels.

5. CONCLUSIONS

Dealing with vast amounts of unlabeled data is a growing problem in many domains. We have presented a direct way of using adaptive resampling methods for selecting a subset of the instances for labeling. The experiments with various benchmarks indicate that this method is successful in significantly reducing the size of the labeled training set needed without sacrificing the classification accuracy when

compared with a state-of-the-art method like adaptive resampling with trees.

6. REFERENCES

[1] N. Abe and H. Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the International Conference on Machine Learning*, pages 1–9, 1998.

[2] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[3] C. Apte, F. Damerau, and S. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, July 1994.

[4] E. Bauer and R. Kohavi. An empirical comparison of

voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105–142, 1999.

[5] M. Berry and G. Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Support.* John Wiley and Sons, Inc., 1997.

[6] C. Blake, E. Keogh, and C. Merz. UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Science, URL=http://www.ics.uci.edu/~mlearn/-MLRespository.html, 1998.

[7] L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–849, 1998.

[8] W. Cochran. *Sampling Techniques.* John Wiley and Sons, Inc., 1977.

[9] D. Cohn, L. Atlas, and R.Ladner. Training connectionist networks with queries and selective sampling. In *Advances in Neural Information Processing Systems 2*. Morgan Kaufmann, 1990.

[10] D. Cohn, L. Atlas, and R.Ladner. Improved generalization with active learning. *Machine Learning*, 15:201–221, 1994.

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

[12] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with mixture models. In *Multiple model approaches to modeling and control.* Taylor and Francis, 1997.

[13] T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine Learning*, 40(2), August 2000.

[14] Y. Freund. Sifting informative examples from a random source. In *Advances in Neural Information Processing*, pages 85–89, 1994.

[15] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.

[16] Y. Freund, H. Seung, E. Shamir, and N. Tishby. Information, prediction, and query by committee. In *Advances in Neural Information Processing Systems 5*, pages 337–344. Morgan Kaufmann, 1992.

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

[18] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. Technical Report Technical Report, Stanford University, Dept. of Statistics, July 1998.

[19] N. Kushmerick. Learning to remove internet advertisements. In *Proceedings of the Third International Conference on Autonomous Agents*, pages 175–181, 1999.

[20] D. Lewis. Reuters 21578 data set. URL=http://www.research.att.com/lewis/-reuters21578.html.

[21] D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 148–156, 1994.

[22] D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the Seventeenth Annual ACM-SIGR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

[23] R. Liere and P. Tadepalli. Active learning with committees for text categorization. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 591–596, 1997.

[24] A. McCallum and K. Nigam. Employing em in pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 350–358, 1998.

[25] F. Provost, D. Jensen, and T. Oates. Efficient progressive sampling. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 23–32, 1999.

[26] R. Schapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[27] H. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth ACM Workshop on Computational Learning Theory*, pages 287–294, 1992.

[28] S. Weiss, C. Apte, F. Damerau, D. Johnson, F. Oles, T. Goetz, and T. Hampp. Maximizing text-mining performance. *IEEE Intelligent Systems and their applications*, 14(4):63–69, July/August 1999.

[29] S. Weiss and N. Indurkhya. Data-miner software kit (DMSK). URL=http://www.data-miner.com, 1998.

[30] S. M. Weiss and C. A. Kulikowski. *Computer Systems that Learn.* Morgan Kaufmann, 1991.

[31] Y. Yang and J. Pedersen. A comparitive study on feature selection in text categorization. In *ICML '97, Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420, 1997.