# Active Learning with Adaptive Heterogeneous Ensembles

Zhenyu Lu*, Xindong Wu*+, Josh Bongard*
*University of Vermont, Department of Computer Science, Burlington, VT 05401
+School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China
zlu@uvm.edu; xwu@cems.uvm.edu; jbongard@uvm.edu

***Abstract*—One common approach to active learning is to iteratively train a single classifier by choosing data points based on its uncertainty, but it is nontrivial to design uncertainty measures unbiased by the choice of classifier. Query by committee [1] suggests that given an ensemble of diverse but accurate classifiers, the most informative data points are those that cause maximal disagreement among the predictions of the ensemble members. However the method for finding ensembles appropriate to a given data set remains an open question. In this paper, the random subspace method is combined with active learning to create multiple instances of different classifier types, and an algorithm is introduced that adapts the ratio of different classifier types in the ensemble towards better overall accuracy. Here we show that the proposed algorithm outperforms C4.5 with uncertainty sampling, Naive Bayes with uncertainty sampling, bagging, boosting and the random subspace method with random sampling. To the best of our knowledge, our work is the first to adapt the ratio of classifiers in a heterogeneous ensemble for active learning.**

***Keywords*-Active Learning, Ensemble Learning, Adaptive Heterogeneous Ensembles;**

## I. Introduction

Classification techniques need both input and label information to build classifiers. In many data mining applications, data points without labels are abundant, but obtaining the labels often involves the work of human experts and thus is a costly and time-consuming process. For example, a time-consuming, expensive and/or dangerous procedure might be required to obtain class labels for patients in a medical data set. The objective of active learning is to query labels only for a carefully chosen subset of data points to reduce the cost of labeling while minimizing impact on classification accuracy.

Active learning can be divided into two categories. One common approach is to choose one classifier and select data points that help the training of this classifier, which normally includes choosing data points according to some confidence measure. This approach includes uncertainty sampling [2][3], in which data points that the current classifier is most uncertain about are considered informative. Many other methods have followed this approach [4][5][6], where uncertainty measures were developed for popular classifiers such as SVM [7] or $k$NN [8]. Recent work [9] introduced an uncertainty measure for C4.5 [10] and performed active learning with constructed data points.

As is known, there is no one classifier universally suitable for all classification tasks. Different classifiers perform best with a particular distribution of data points. Therefore any method following the uncertainty sampling approach must first select the appropriate classifier type and then design an uncertainty measure according to the choice of classifier type. Interesting recent work [11] exploits the clustering structures of data sets to suggest an active learning method that is independent of the choice of classifier type, but the effectiveness of the approach has only been tested with logistic regression.

The other approach to active learning is query by committee (QBC)[1], in which the most informative data points are those that cause maximal disagreement among the predictions of a committee of classifiers. QBC requires the committee of classifiers to be both accurate and diversified, which is also desired in ensemble learning [12]. Combining these two approaches is promising because it is known that ensemble learning methods [13][14][15][16] generalize better than a single learner [17] and there is no need to design different uncertainty measures for different ensemble methods. In [18], bagging [13] and boosting [14] were combined with QBC, but both ensemble methods require a fair amount of data to start the learning process, which is not desirable considering the cost of labeling data points. In a different approach, the estimation-exploration algorithm [19][20][21](EEA) uses stochastic optimization algorithms to optimize models and find disagreement-causing data points. Informative sampling [22] adapted the EEA for data mining, but its performance is constrained by the limited power of stochastic optimization. For some classifiers, a satisfactory stochastic optimization algorithm may not exist, and existing classifiers such as C4.5 and Naive Bayes [23] can not currently be used in this framework.

In this paper, the random subspace method (RSM)[15] is combined with QBC. As a well-established ensemble method, RSM offers good generalization and only requires one data point to start training. Although previous methods have focused on building homogeneous ensembles with QBC, homogeneous ensembles face similar problems as when using one classifier: the bias of the chosen classifier type. For example, decision trees form decision boundaries that consist of lines or planes that are orthogonal to one

of the input features, which makes them less suitable for a data set with continuous non-linear decision boundaries. Intuitively, using different classifiers to construct an ensemble would be helpful because it provides better diversity while each classifier remains partially accurate. Many methods have been suggested to construct heterogeneous ensembles. The ensembles were typically formed by including one instance of each classifier [24] or selecting members of an ensemble from a large library of classifiers trained with different parameters [25]. RSM is used here to construct the ensemble because it can create a large number of different instances of any classifier type.

This paper shows that active learning with heterogeneous ensembles outperforms homogeneous ensembles, and that different ratios of classifier types in an ensemble suit different data sets. The question then arises as to how to find a suitable ratio of classifier types for a given data set. An algorithm is introduced in this paper to adapt the ratio of different classifier types in a heterogeneous ensemble during training, which we term adaptive heterogeneous ensembles (AHE). The algorithm starts with a heterogeneous ensemble and alternates between a query phase and a training phase. In the query phase, classifiers in the current ensemble predict labels for a set of unlabeled data points, and the data point that induces the maximum prediction variance across the ensemble is chosen for label querying and then added to the training set. In the training phase, the ratio of different classifier types is adapted toward the ratio with better performance and a new ensemble of classifiers is trained according to the adapted ratio. In this work, we are especially interested in small ensemble sizes because it is computationally expensive to construct and evaluate a large new ensemble in each iteration. The major difference between our approach and [25] is that we are trying to find the best way to combine different classifier types for small ensembles rather than training a large library of classifiers and trying to find a more accurate subset of classifiers from it. To the best of our knowledge, our work is the first to adapt the ratio of classifiers in a heterogeneous ensemble for active learning.

In our current work, C4.5 and Naive Bayes were chosen as the classifier types, but AHE can work with any classification technique. Experiments on five data sets from the UCI Machine Learning Repository [26] show that the algorithm successfully finds the appropriate ratio for a data set and outperforms C4.5 with uncertainty sampling, Naive Bayes with uncertainty sampling, the same algorithm with random sampling, and bagging, boosting, and the random subspace method with random sampling.

The rest of the paper is organized as follows: Section 2 introduces the algorithm and the adaptive strategy, Section 3 provides the results and Section 4 concludes the paper.

Table I
THE AHE ALGORITHM

1 Initialization
- a) Randomly choose a data point from the pool, query its label and add it to the training set.
- b) Train an ensemble of heterogeneous classifiers on the current training set.

2 The Query Phase
- a) Each data point in the unlabeled data pool is passed to the ensemble.
- b) Each classifier in the current ensemble predicts the label of each data point in the pool.
- c) The data point that induces the most label prediction disagreement is queried for its label and added to the training set.

3 The Training Phase
- a) The ratio of classifier types in the ensemble is adapted towards the ratio with higher accuracy.
- b) A new ensemble of classifiers is generated according to the new adapted ratio.

4 Termination
- a) Step 2 and 3 are repeated until the ensemble meets some pre-defined criteria.
- b) Predictions are made by taking the majority vote of the resulting ensemble members.

## II. METHOD

The adaptive heterogeneous ensembles method alternates between two phases: the query phase chooses one data point for labeling and adds it to the training set, and the training phase adapts the ensemble towards a better combination of classifiers. One iteration of the AHE is defined as a query phase followed by a training phase. Table I outlines the general algorithm.

The algorithm starts by randomly selecting one data point from the current pool of training data, the label of this data point is queried, and the data point is added to the training set. In this work, the pool of training data is made available to the algorithm in a streaming manner similar to [27]. The whole potential training set is partitioned into chunks of equal size, and in each iteration one chunk of data serves as the pool for choosing one new data point.

There are two reasons to run the algorithm in a streaming manner. The first is for the efficiency of the algorithm. Active learning is especially desirable when the potential unlabeled training set is large. Therefore, scanning the whole pool of potential training data in each iteration to choose one data point is prohibitively inefficient. The second reason is that in this way the algorithm works on both static and streaming data.

After the first data point has been chosen out of the first chunk of data, a heterogeneous ensemble of classifiers is generated according to a predefined initial ratio. In this work, the first ensemble consists of five C4.5 and five Naive Bayes

classifiers, which provides the two classifier types with equal footing. Each classifier is trained on a 50% random subspace of the current training set.

During the first query phase, the second chunk of data is made available to the ensemble. Predictions are made by each classifier in the ensemble for each data point in the chunk. The predicted labels are recorded for each data point, and the data point with the largest variance of predicted labels is chosen. For a two class problem, the most informative data points are those that cause as close to half of the classifiers in the ensemble to predict the positive label as possible, and the remainder to predict the negative label. Ties are broken by randomly choosing a data point from the chunk. The label of the chosen point is queried, and this data point is added to the training set as the second data point.

The first query phase is followed by the first training phase. In this phase, the ratio of classifier types in the ensemble is first adapted, and then a new ensemble of classifiers is generated according to the new ratio, each again on a 50% random subspace on the current training set. Several adaptation strategies could work with AHE. The adaptation strategy in this paper works as follows:

1) The entire data set is partitioned into three sets: the training pool, the testing set and the adaptation set which is used for adapting the ratio of the ensemble.

2) Each time the ensemble is adapted, three accuracies are obtained for the ensemble against the adaptation set: the voting accuracy of the entire ensemble, the voting accuracy of the current ensemble with one randomly chosen C4.5 classifier removed, and the voting accuracy of the current ensemble with one randomly chosen Naive Bayes classifier removed.

3) The ratio of C4.5 and Naive Bayes classifiers are adapted according to the comparison of the three accuracies. If the accuracy with one C4.5 classifier removed is better than the accuracy with one Naive Bayes classifier removed and better than or equal to the accuracy with the entire ensemble, then the number of C4.5 classifiers is decreased by one, and the number of Naive Bayes classifier is increased by one. The relative numbers are adjusted similarly in the opposite case. If the entire ensemble achieves the best accuracy, then the ratio remains the same.

The algorithm then starts the second query phase: a third data point is chosen from the third chunk of data and added to the training set. The second training phase then adapts the ratio of the ensemble again, and a new ensemble is trained according to the new ratio on the updated training set.

The algorithm executes a predefined number of iterations or until the accuracy of the ensemble meets some pre-defined criteria.

## III. Results

In this section the experimental settings and data sets are introduced, and then the experimental results are reported. Two sets of experiments were conducted. In the first set of experiments, evidence is provided that heterogeneous ensembles consistently outperform homogeneous ensembles, and different combinations of C4.5 and Naive Bayes classifiers suit different data sets. Then the performance of the AHE was compared against the same algorithm with static ratios to show the effectiveness of the adaptation strategy. The second set of experiments compares the accuracy of AHE with uncertainty sampling and random sampling.

### A. Data sets and experimental settings

Experiments were conducted on five data sets from the UCI Machine Learning Repository [26]. All data sets were shuffled and partitioned into three sets: the training pool, the testing set and the adaptation set as mentioned before. For all data sets, parameters were chosen empirically as a trade-off between computation time and labeling cost, which are also criteria for choosing parameters in practice.

The first data set was one of the feature sets from the Multiple Features Data Set called mfeat-pixel[1]. This data set consists of features of handwritten digits extracted from a collection of Dutch utility maps. The mfeat-pixel data set contains 240 features and 10 possible output labels, with each representing a digit between “0” and “9”. For each class label there are 200 data points, thus the whole data set contains 2000 data points. There are no missing values in the data set. 1200 data points were randomly chosen for the training pool, 500 were randomly chosen to serve as the testing set, and the remaining 300 data points serve as the adaptation set. The training pool was partitioned into 400 chunks, with each containing three data points. One data point was drawn into the training set during each iteration of the algorithm. There totally 400 such iterations. Thus at the end of an execution of the algorithm, there were 400 data points in the training set.

The second data set was the Page Blocks Classification Data Set[2], henceforth referred to as “page”. The data set recorded 10 features of segmented blocks of a document, and the task is to classify a block as “text”, “horizontal line”, “picture”, “vertical line” and “graphic”. There are 5473 data points in this data set with no missing values, of which 4913(89.8%) data points belong to class “text”. 4000 data points were randomly chosen to serve as the training pool, 737 data points were randomly chosen to form the testing set, and the adaptation set contains the remaining 736 data points. Each chunk of the training pool contains five data points, and one data point was drawn into the training set

[1] http://archive.ics.uci.edu/ml/datasets/Multiple+Features

[2] http://archive.ics.uci.edu/ml/datasets/Page+Blocks+Classification

during each of the 800 iterations. Thus the final training set contains 800 data points.

The third data set tested from UCI was the Statlog (Landsat Satellite) Data Set[3],henceforth referred to as "satImg". The data set consists of the multi-spectral values of pixels in 3x3 neighborhoods in a satellite image, the objective of which is to predict the label associated with the central pixel in each neighborhood. There are a total of 6435 data points in the data set without missing values. Each data point has 36 features and six possible class labels. The training pool of this data set contains 4435 randomly chosen data points, the testing set contains 1000 randomly chosen data points, and the remaining 1000 data points make up the adaptation set. The training pool was partitioned into chunks of 10 data points. One data point was chosen out of a chunk in each iteration of the algorithm, and there were 400 such iterations. Thus the final training set contains 400 data points.

The fourth data set from UCI was the Spambase Data Set[4], henceforth referred to as "spam". 57 features were collected from 4601 emails in this data set to predict if a given email is "spam" or "non-spam", and there are no missing values. 3601 data points were randomly chosen as the training pool, 500 were randomly chosen for the testing set and the remaining 500 data points serve as the adaptation set. Each chunk of the training pool contains five data points. One data point was chosen out of a chunk in each iteration, and 700 iterations were conducted during each run. Thus there were 700 data points in the final training set.

The last data set was the Waveform Data Set (Version 2)[5]. This data set is an artificial data set generated with noise. Each data point in the data set has 40 features generated to predict three classes of waves. All of the features include noise, and the latter 19 features are all noise features with zero mean and unity variance. Of the 5000 data points, 4000 were randomly chosen as the training pool, 1000 were randomly chosen as the testing set, and the adaptation set contained the remaining 1000 data points. The training pool was partitioned into chunks of 10, one data point was chosen out of each chunk during each of the 400 iterations. Thus the final training set contained 400 data points.

All results report the means and standard deviations of 30 independent runs, and all accuracies were reported as error percentages. Each independent run works with a new seed to generate random numbers. The random numbers are used to randomly choose the first data point for AHE and uncertainty sampling, and to randomly choose all data points for random sampling. The C4.5 and Naive Bayes classifiers were the implementations by Weka [28] with default settings. The ensemble size was ten for each experiment.

[3]http://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)

[4]http://archive.ics.uci.edu/ml/datasets/Spambase

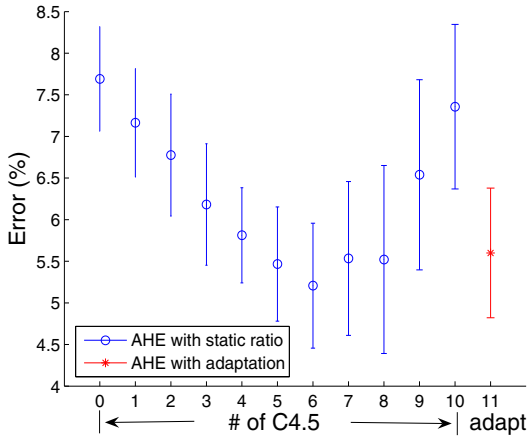
[5]http://archive.ics.uci.edu/ml/datasets/Waveform+Database+Generator+(Version+2)

### *B. Comparison with Static Ensembles*

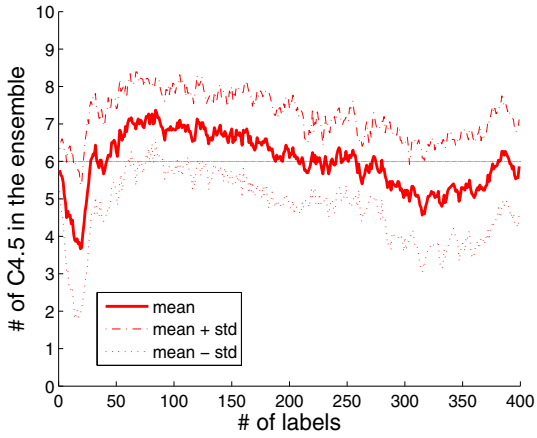
In this section, evidence is provided that heterogeneous ensembles work consistently better than homogeneous ensembles with active learning on the five tested data sets, and the effectiveness of AHE is reported in terms of both accuracy compared against AHE with static ratios and the algorithm's ability to converge to suitable ratios. The experiments on AHE with static ratios consist of the test of AHE on all possible starting ratios of C4.5 and Naive Bayes classifiers without adaptation: we tested the AHE with static ratios from zero C4.5 classifiers and 10 Naive Bayes classifiers to 10 C4.5 classifiers and zero Naive Bayes classifiers.

Figure 1 reports the results using data set "mfeat-pixel". In Figure 1(a), the error bars with circles represent AHE with static ratios (lines zero to 10 indicate the number of C4.5 classifiers in the ensembles), and the error bar with an asterisk represents AHE with adaptation. In Figure 1(b), ratio changes of AHE with adaptation are reported as the changes in the number of C4.5 classifiers in the ensemble over iterations. It can be seen that AHE with static ratios perform best when there are six C4.5 and four Naive Bayes classifiers in the ensemble. Although the accuracy of this ratio is only marginally better than employing equal numbers of C4.5 and Naive Bayes classifiers as well as seven C4.5 and three Naive Bayes classifiers, it is statistically significantly better than using homogeneous ensembles with all C4.5 classifiers or all Naive Bayes classifiers. AHE with adaptation converges suitably to close to six C4.5 classifiers in the ensemble as Figure 1(b) indicates, but is slightly though not significantly worse in accuracy than AHE with the best static ratio (compare line 11 to line 6 in Figure 1(a)). The reason is that the best combination of static ensembles aids AHE to choose the most informative data points for training as well as to make accurate predictions. Although AHE adapts the ensemble towards the best ratio, there might be costs associated with the quality of the chosen data during adaptation. This can clearly be seen before the 50th iteration in Figure 1(b) in which the ensemble first favors more Naive Bayes classifiers and then starts to increase the number of C4.5 classifiers. Fortunately, the cost appears marginal as the AHE with adaptation achieves statistically significantly better accuracy than homogeneous ensembles (compare line 11 to lines zero and 10 in Figure 1(a)).

The results for the "page" data set are reported in Fig. 2. Fig. 2(a) shows that this data set favors more C4.5 classifiers in the ensemble such that there are no significant accuracy differences when there are more than six C4.5 classifiers in the static ensembles. As is shown by Fig. 2(b), AHE with adaptation often converges appropriately to eight C4.5 classifiers, and the algorithm achieves equivalent accuracy compared to AHE with the optimal static ratios (compare line 11 to lines 6-10 in Fig. 2(a)).

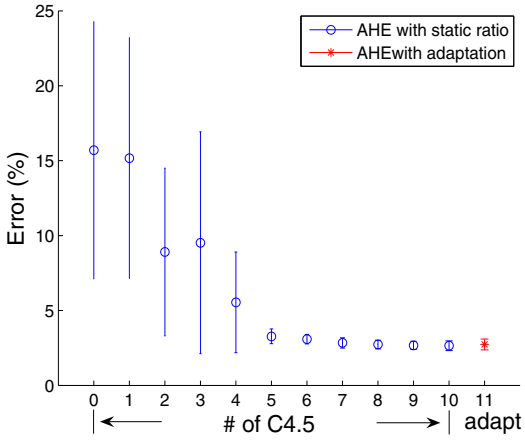



(a) Comparison between static and adaptive ensembles on the "mfeat-pixel" data set.

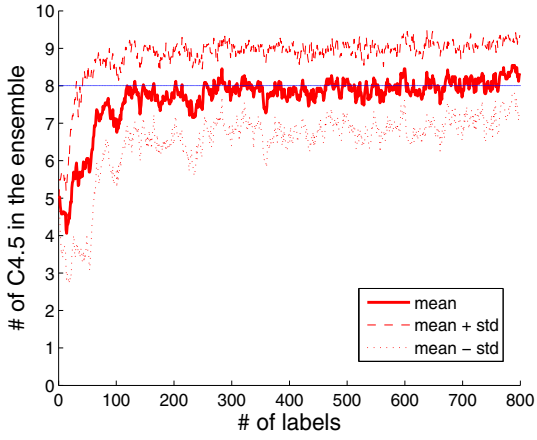



(b) Change in ratio of classifier types for the "mfeat-pixel" data set.

Figure 1. Ability of AHE to discover the optimal ratio for the "mfeat-pixel" data set.




(a) Comparison between static and adaptive ensembles for the "page" data set.




(b) Change in ratio of classifier types for the "page" data set.

Figure 2. Ability of AHE to discover the optimal ratio for the "page" data set.

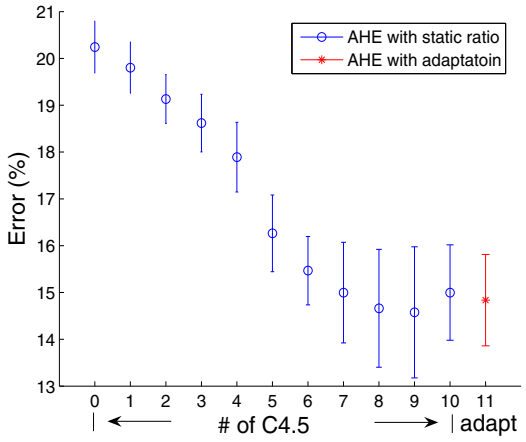
Like the "page" data set, the "satImg" data set is more accurately modeled if more C4.5 than Naive Bayes classifiers are employed in the ensemble (Fig. 3(a)), though eight or nine C4.5 classifiers are optimal. AHE with adaptation converges correctly to close to a mean of eight C4.5 classifiers as seen in Fig. 3(b), and the algorithm achieves comparable accuracy compared to the best classifier ratios in the static ensembles (compare line 11 to lines 8 and 9 in Fig. 3(a)).

The "spam" data set is also modeled better when a majority of C4.5 classifiers are employed. Fig. 4(a) indicates that the AHE which employ ensembles with six or more C4.5 classifiers all perform equivalently, and better than those that employ five or fewer (compare line 11 to lines 6-10 in Fig. 4(a)). It is shown in Fig. 4(b) that AHE with adaptation converges to on average seven C4.5 classifiers.

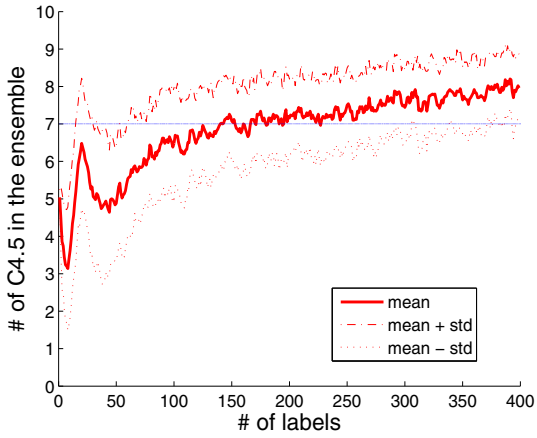
Unlike the previous three data sets, the "waveform" data set is more amenable to modeling by Naive Bayes classifiers when incorporated in the AHE. As is shown in Fig. 5(a), AHE with three C4.5 classifiers performs significantly or marginally better than other ensemble ratios. AHE with adaptation achieves comparable accuracy to the best static ratio of three C4.5 and seven Naive Bayes classifiers (compare line 11 to 3 in Fig. 5(a)), and Fig. 5(b) shows that AHE suitably converges to a mean of three C4.5 classifiers.

### C. Comparison against Random and Uncertainty Sampling

This section reports comparisons of AHE against random sampling and uncertainty sampling. For each of the tested data sets, two sets of experiments were conducted: AHE was first compared against AHE with random sampling, C4.5 with uncertainty sampling, and Naive Bayes with

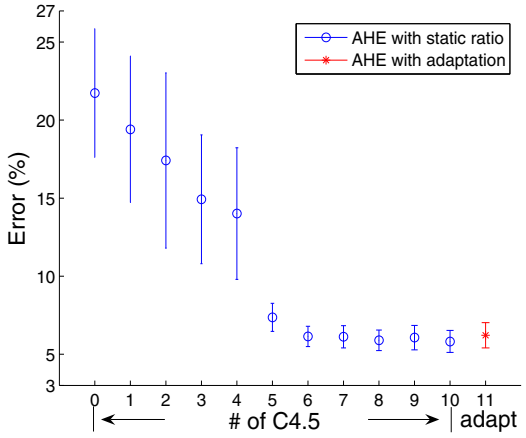



(a) Comparison between static and adaptive ensembles for the "satImg" data set.

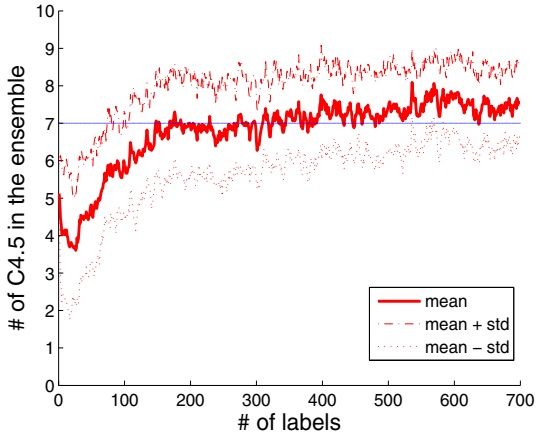



(b) Change in ratio of classifier types for the "satImg" data set.

Figure 3. Ability of AHE to discover the optimal ratio for the "satImg" data set.




(a) Comparison between static and adaptive ensembles for the "spam" data set.



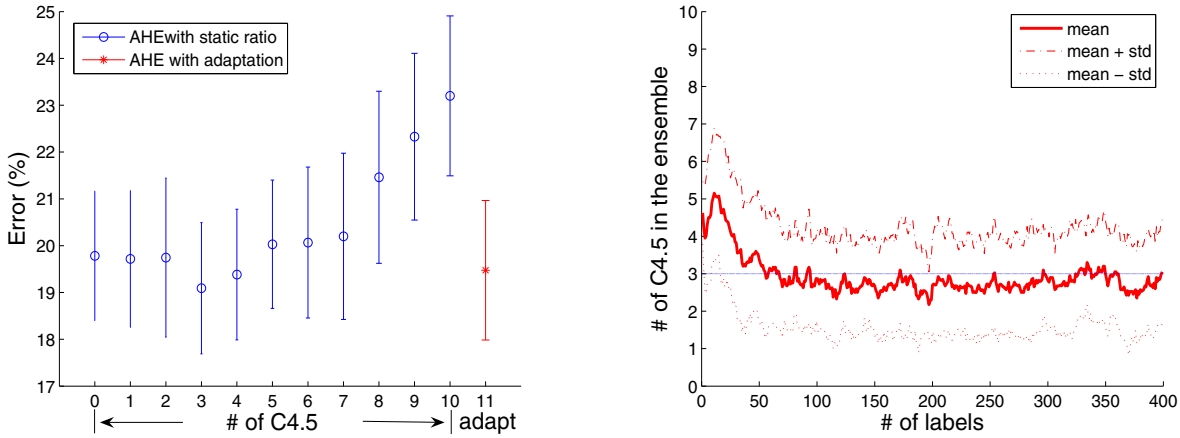

(b) Change in ratio of classifier types for the "spam" data set.

Figure 4. Ability of AHE to discover the optimal ratio for the "spam" data set.

uncertainty sampling over iterations, and then AHE was compared against bagging with random sampling, boosting with random sampling and the random subspace method with random sampling. AHE with random sampling was achieved by disabling the query phase: during each iteration, a single randomly-selected data point is drawn from each chunk. Adaptation of classifier ratio was retained in this algorithm variant. The first comparison is reported over iterations as the means and standard deviations of compared algorithms.

The same settings employed in the AHE were used for C4.5 with uncertainty sampling, except that during each iteration the data point with the lowest confidence output by Weka [28] was drawn from the data chunk into the training set, and a C4.5 classifier was then trained on this enlarged training set. Naive Bayes with uncertainty sampling was implemented similarly to C4.5 with uncertainty sampling. Both uncertainty sampling methods start with a randomly chosen data point as AHE does, which explains the standard deviations in accuracies of the two deterministic algorithms in the figures.

The second comparison was conducted because active learning methods with a poorly selected classifier might perform worse than a more accurate classifier with random sampling. Bagging, boosting and the random subspace method with C4.5 as base learners were therefore compared against AHE with random sampling. All methods were implemented by calling Weka with default settings. The three methods were trained on randomly-sampled training sets of the same size as those collected by the AHE. The



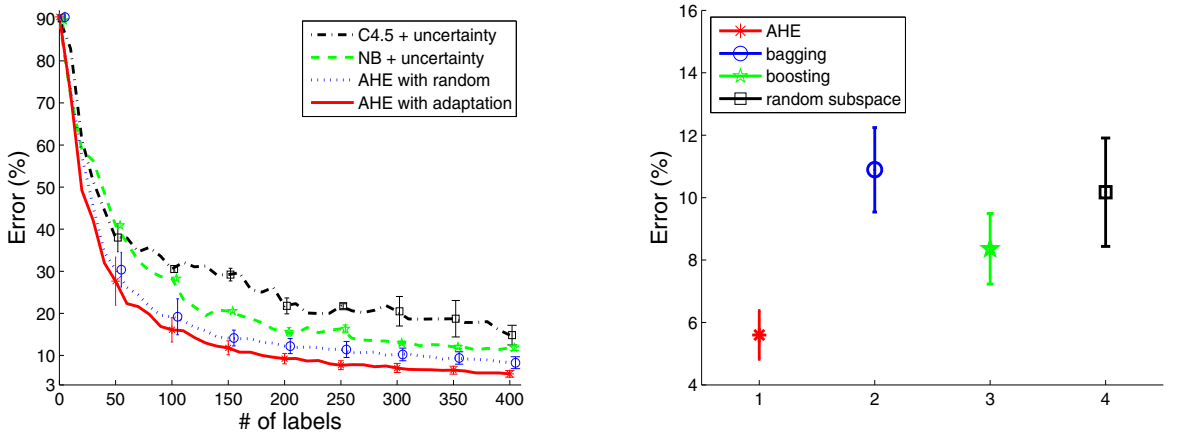

(a) Comparison between static and adaptive ensembles for the "waveform" data set.

(b) Change in ratio of classifier types for the "waveform" data set.

Figure 5. Ability of AHE to discover the optimal ratio for the "waveform" data set.

Table II
RELATIVE ERRORS FOR ALL METHODS ON ALL TESTED DATA SETS.

| | C4.5 (%) | NB (%) | AHE random (%) | bagging (%) | boosting (%) | RSM (%) | AHE (%) |
|---|---|---|---|---|---|---|---|
| mfeat-pixel | 14.83 ± 2.35 *** | 11.83 ± 0.66 *** | 8.28 ± 1.41 *** | 10.89 ± 1.35 *** | 8.36 ± 1.13 *** | 10.17 ± 1.74 *** | **5.6 ± 0.78** |
| page | 3.99 ± 0.4 *** | 23.64 ± 4.9 *** | 3.67 ± 0.54 *** | 3.54 ± 0.45 *** | 3.51 ± 0.53 *** | 3.64 ± 0.5 *** | **2.73 ± 0.36** |
| satImg | 23.89 ± 2.86 *** | 20.11 ± 0.47 *** | 17.33 ± 0.92 *** | 18.6 ± 1.7 *** | 17.8 ± 1.25 *** | 17.32 ± 1.31 *** | **14.84 ± 0.98** |
| spam | 12.29 ± 1.74 *** | 20.75 ± 0.35 *** | 7.21 ± 0.99 *** | 8.31 ± 0.56 *** | 6.83 ± 0.77 ** | 6.9 ± 0.87 ** | **6.21 ± 0.81** |
| waveform | 31.16 ± 3.21 *** | 20.81 ± 1 *** | 20.96 ± 1.54 *** | 23.67 ± 2.19 *** | 22.49 ± 1.47 *** | 22.82 ± 2.07 *** | **19.48± 1.49** |



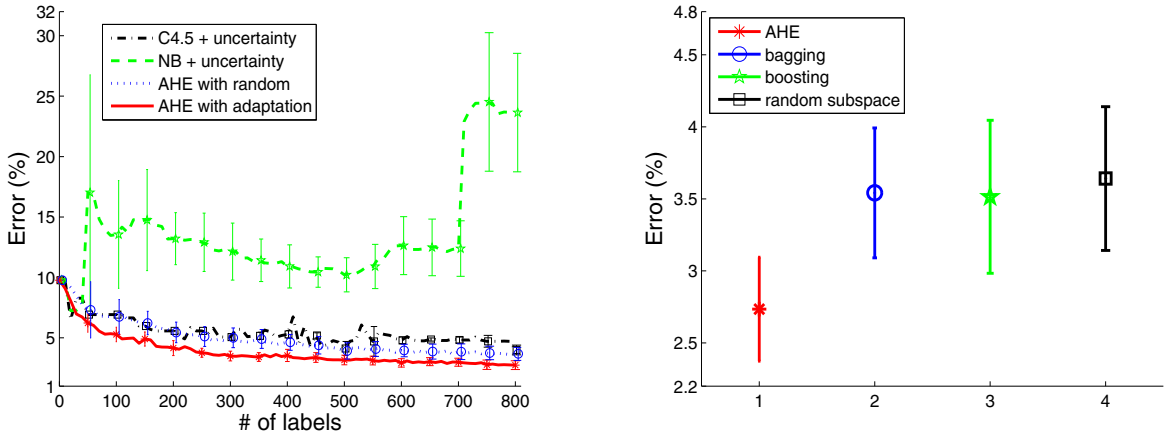

(a)

(b)

Figure 6. Comparison against random and uncertainty sampling for the "mfeat-pixel" data set.

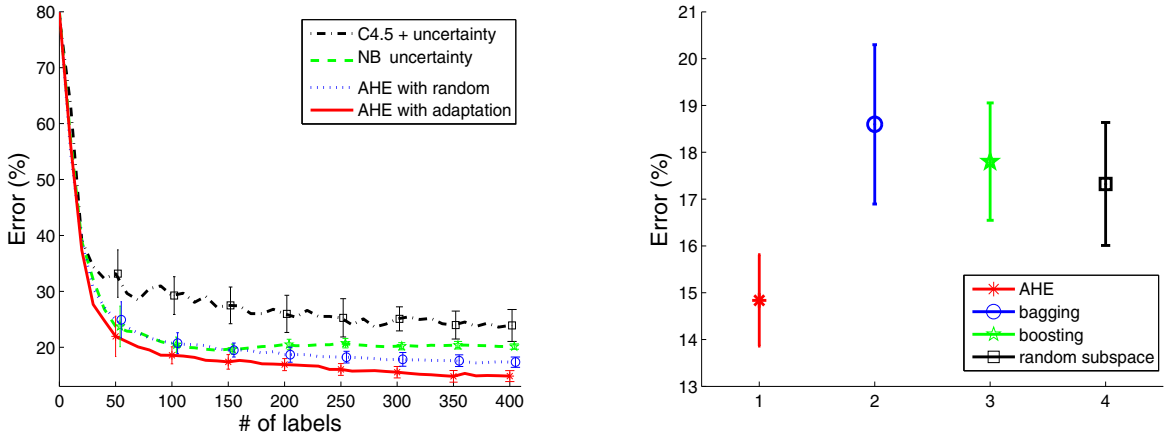
results are reported in Fig. 6-10, and Table II. Each cell in Table II reports the mean and standard deviation of error percentage over 30 independent runs and the significance level $p$ ("*" represents $p < 0.05$, "**" represents $p < 0.01$, and "***" represents $p < 0.001$) between the accuracy of AHE and the corresponding method from the Mann-Whitney U test [29].

Fig. 6 reports the results for the "mfeat-pixel" data set.



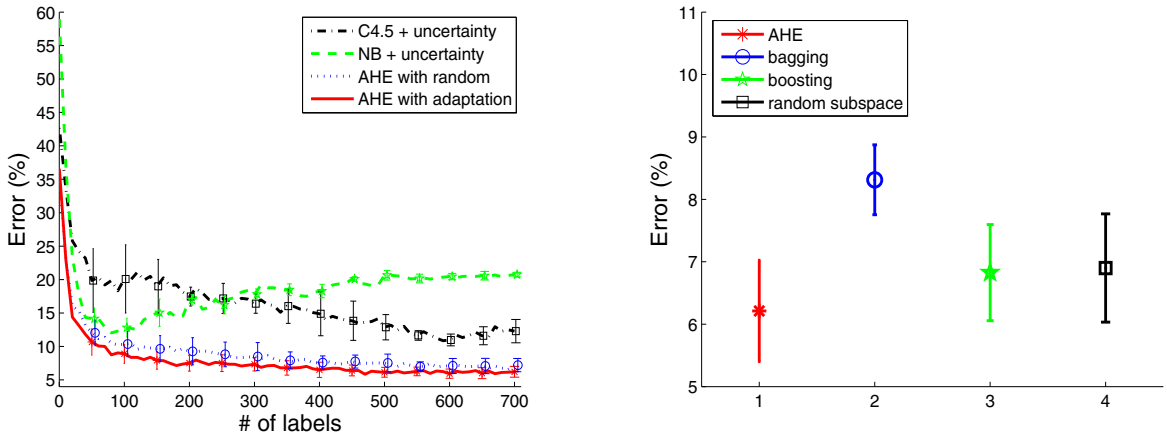

(a) (b)

Figure 7. Comparison against random and uncertainty sampling for the "page" data set.



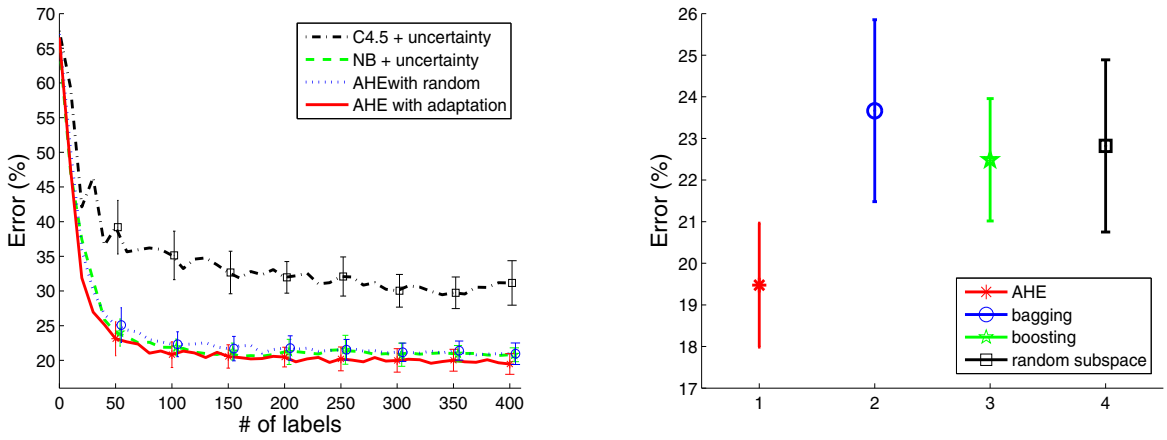

(a) (b)

Figure 8. Comparison against random and uncertainty sampling for the "satImg" data set.




(a) (b)

Figure 9. Comparison against random and uncertainty sampling for the "spam" data set.




(a) (b)

Figure 10. Comparison against random and uncertainty sampling for the "waveform" data set.

Fig. 6(a) indicates that AHE outperforms all competing methods, and Fig. 6(b) shows that AHE is statistically significantly more accurate than bagging, boosting and the random subspace method.

On the "page" data set, Fig. 7(a) indicates that AHE also outperforms all the other tested methods as well. It is interesting to note in Fig. 7(a) that the accuracy of Naive Bayes with uncertain sampling degenerates over the course of a run. This is because, as described in section III-A, the "page" data set is highly unbalanced. The Naive Bayes classifiers may achieve better accuracy early on in a run by simply predicting the majority label, but when more data points with minority labels are drawn into the training set because they tend to have low confidence, the Naive Bayes classifiers predict increasingly erroneous labels. Fig. 7(b) shows that AHE outperforms bagging, boosting and the random subspace method.

For the "satImg" data set, AHE is shown to be more accurate than all competing algorithms as shown by Figs. 8(a) and 8(b). For the "spam" data set, Fig. 9(a) shows that AHE significantly outperforms C4.5 and Naive Bayes with uncertainty sampling, and is slightly more accurate than AHE with random sampling. It can be seen in Fig. 9(b) that AHE is statistically significantly more accurate than bagging with random sampling, and performs slightly better than boosting and the random subspace method with random sampling. For the "waveform" data set, AHE is shown in Fig. 10(a) to significantly outperform C4.5 with uncertainty sampling, and slightly outperform AHE with random sampling and Naive Bayes with uncertainty sampling. It is shown in Fig. 10(b) that AHE again outperforms bagging, boosting and the random subspace method with random sampling.

In conclusion, AHE outperforms state-of-the-art classifiers that rely on uncertainty sampling as well as state-of-the-art ensemble methods that rely on random sampling, for all five of the tested data sets.

## IV. Conclusions and Future Work

In this paper, the random subspace method is employed to create multiple instances of different classifier types within an ensemble. The resulting heterogeneous ensemble is then exploited to draw informative training data from the entire data corpus using query by committee, and the expanded training set is then used to alter the ratio of classifier types toward one that is appropriate for the given data set. This process is repeated until some termination criteria are met. We term this combined approach *adaptive heterogeneous ensemble learning*. We show that our approach outperforms homogeneous ensembles such as bagging, boosting and the random subspace method, as well as heterogeneous ensembles with fixed classifier ratios, in which all methods use the same amount of training data. Our approach also outperforms C4.5 and Naive Bayes using uncertainty sampling (an active learning approach that does not require ensembles).

Although C4.5 and Naive Bayes were used as classifiers in this paper, AHE is a framework that can accommodate any classifier type. Future work will involve expanding AHE to include more classifiers in the ensemble and testing its performance against other classification methods on more data sets. We also plan to investigate more intelligent methods for adapting the ratio of classifier types and choosing subspaces for ensemble members. Finally, we plan to design methods that automatically determine parameters such as the size of the ensemble and the rate at which training data is drawn from the data corpus in static and streaming domains.

## Acknowledgment

This research is supported in part by the US National Science Foundation (NSF) under grants EPS-0701410 and

CCF-0905337, in part by the National Natural Science Foundation of China (NSFC) under award 60828005, and in part by the National Basic Research Program of China (973 Program) under award 2009CB326203.

## REFERENCES


[1] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the Fifth Workshop on Computational Learning Theory*, 1992, pp. 287–294.

[2] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of Research and Development in Information Retrieval*, 1994, pp. 3–12.

[3] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proceedings of the 11th International Conference on Machine Learning*, 1994, pp. 148–156.

[4] G. Schohn and D. Cohn, "Less is more: active learning with support vector machines," in *Proceedings of the 17th International Conference on Machine Learning*, 2000, pp. 839–846.

[5] M. Lindenbaum, S. Markovitch, and D. Rusakov, "Selective sampling for nearest neighbor classifiers," *Machine Learning*, vol. 54(2), pp. 125–152, 2004.

[6] S. Tong and D. Koller, "Support vector machine learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2001.

[7] N. Cristianini and J. S. Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[8] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. Pearson Addison-Wesley, 2006.

[9] C. X. Ling and J. Du, "Active learning with direct query construction," in *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 480–487.

[10] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann, 1993.

[11] S. Dasgupta and D. Hsu, "Hierarchical sampling for active learning," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 208–215.

[12] T. G. Dietterich, "Machine-learning research: Four current directions," *The AI Magazine*, vol. 18(4), pp. 97–136, 1998.

[13] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24(2), pp. 123–140, 1996.

[14] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55(1), pp. 119–139, 1997.

[15] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20(8), pp. 832–844, 1998.

[16] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial Intelligence*, vol. 137(1-2), pp. 239 – 263, 2002.

[17] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14(1), pp. 1–37, 2008.

[18] N. Abe and H. Mamitsuka, "Query learning strategies using boosting and bagging," in *Proceedings of the 15th International Conference on Machine Learning*, 1998, pp. 1–9.

[19] J. Bongard and H. Lipson, "Automating genetic network inference with minimal physical experimentation using coevolution," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2004, pp. 333–345.

[20] J. Bongard and H. Lipson, "Active coevolutionary learning of deterministic finite automata," *Journal of Machine Learning Research*, vol. 6, pp. 1651–1678, 2005.

[21] J. Bongard, V. Zykov, and H. Lipson, "Resilient machines through continuous self-modeling," *Science*, vol. 314, pp. 1118–1121, 2006.

[22] Z. Lu, A. I. Rughani, B. I. Tranmer, and J. Bongard, "Informative sampling for large unbalanced data sets," in *Proceedings of the Genetic and Evolutionary Computation Conference, Workshop on medical applications of genetic and evolutionary computation*, 2008, pp. 2047–2054.

[23] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine Learning*, vol. 29(2-3), pp. 103–130, 1997.

[24] S. Dzeroski and B. Zenko, "Is combining classifiers with stacking better than selecting the best one?" *Machine Learning*, vol. 54(3), pp. 255–273, 2004.

[25] R. Caruana, A. Munson, and A. Niculescu-Mizil, "Getting the most out of ensemble selection," in *Proceedings of the Sixth International Conference on Data Mining*, 2006, pp. 828–833.

[26] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html

[27] X. Zhu, P. Zhang, X. Lin, and Y. Shi, "Active learning from data streams," in *Proceedings of the Seventh IEEE International Conference on Data Mining*, 2007, pp. 757–762.

[28] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. San Francisco: Morgan Kaufmann, 2005.

[29] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1(6), pp. 80–83, 1945.