# Diverse Ensembles for Active Learning

**Prem Melville** MELVILLE@CS.UTEXAS.EDU
**Raymond J. Mooney** MOONEY@CS.UTEXAS.EDU
Department of Computer Sciences, University of Texas, 1 University Station C0500, Austin, TX 78712-0233

## Abstract

Query by Committee is an effective approach to selective sampling in which disagreement amongst an ensemble of hypotheses is used to select data for labeling. Query by Bagging and Query by Boosting are two practical implementations of this approach that use Bagging and Boosting, respectively, to build the committees. For effective active learning, it is critical that the committee be made up of consistent hypotheses that are very different from each other. DECORATE is a recently developed method that directly constructs such diverse committees using artificial training data. This paper introduces ACTIVE-DECORATE, which uses DECORATE committees to select good training examples. Extensive experimental results demonstrate that, in general, ACTIVE-DECORATE outperforms both Query by Bagging and Query by Boosting.

## 1. Introduction

The ability to actively select the most useful training examples is an important approach to reducing the amount of supervision required for effective learning. In particular, *pool-based sample selection*, in which the learner chooses the best instances for labeling from a given set of unlabeled examples, is the most practical approach for problems in which unlabeled data is relatively easily available (Cohn et al., 1994). A theoretically well-motivated approach to sample selection is *Query by Committee* (Seung et al., 1992), in which an ensemble of hypotheses is learned and examples that cause maximum disagreement amongst this committee (with respect to the predicted categorization) are selected as the most informative. Popular ensemble learning algorithms, such as Bagging and Boosting, have been used to efficiently learn effective committees for active learning (Abe & Mamitsuka, 1998). Meta-learning ensemble algorithms, such as Bagging and Boosting, that employ an arbitrary base classifier are particularly useful since they are general purpose and can be applied to improve any learner that is effective for a given domain.

An important property of a good ensemble for committee-based active learning is diversity. Only a committee of hypotheses that effectively samples the version space of all consistent hypotheses is productive for sample selection (Cohn et al., 1994). DECORATE (Melville & Mooney, 2003) is a recently introduced ensemble meta-learner that directly constructs diverse committees by employing specially-constructed artificial training examples. Extensive experiments have demonstrated that DECORATE constructs more accurate ensembles than both Bagging and ADABOOST when training data is limited. DECORATE has also been successfully used for the task of *active feature acquisition* (i.e., given a feature acquisition budget, identify the instances with missing values for which acquiring complete feature information will result in the most accurate model) (Melville et al., 2004).

This paper presents a new approach to active learning, ACTIVE-DECORATE, which uses committees produced by DECORATE to select examples for labeling. Extensive experimental results on several real-world datasets show that using this approach produces substantial improvement over using DECORATE with random sampling. ACTIVE-DECORATE requires far fewer examples than DECORATE, and on average also produces considerable reductions in error. In general, our approach also outperforms both Query by Bagging and Query by Boosting.

## 2. Query by Committee

Query by Committee (QBC) is a very effective active learning approach that has been successfully applied to

Appearing in *Proceedings of the 21$^{st}$ International Conference on Machine Learning*, Banff, Canada, 2004. 

different classification problems (McCallum & Nigam, 1998; Dagan & Engelson, 1995; Liere & Tadepalli, 1997). A generalized outline of the QBC approach is presented in Algorithm 1. Given a pool of unlabeled examples, QBC iteratively selects examples to be labeled for training. In each iteration, it generates a committee of classifiers based on the current training set. Then it evaluates the potential utility of each example in the unlabeled set, and selects a subset of examples with the highest expected utility. The labels for these examples are acquired and they are transfered to the training set. Typically, the utility of an example is determined by some measure of *disagreement* in the committee about its predicted label. This process is repeated until the number of available requests for labels is exhausted.

Freund et al. (1997) showed that under certain assumptions, Query by Committee can achieve an exponential decrease in the number of examples required to attain a particular level of accuracy, as compared to random sampling. However, these theoretical results assume that Gibbs algorithm is used to generate the committee of hypotheses used for sample selection. The Gibbs algorithm for most interesting problems is computationally intractable. To tackle this issue, Abe and Mamitsuka (1998) proposed two variants of QBC, Query by Bagging and Query by Boosting, where Bagging and AdaBoost are used to construct the committees for sample selection. In their approach, they evaluate the utility of candidate examples based on the *margin* of the example; where the margin is defined as the difference between the number of votes in the current committee for the most popular class label, and that for the second most popular label. Examples with smaller margins are considered to have higher utility.

**Algorithm 1** Generalized Query by Committee

**Given:**
$T$ - set of training examples
$U$ - set of unlabeled training examples
$BaseLearn$ - base learning algorithm
$k$ - number of selective sampling iterations
$m$ - size of each sample

1. Repeat $k$ times
2. Generate a committee of classifiers, $C^* = EnsembleMethod(BaseLearn, T)$
3. $\forall x_j \in U$, compute $Utility(C^*, x_j)$, based on the current committee
4. Select a subset $S$ of $m$ examples that maximizes utility
5. Label examples in $S$
6. Remove examples in $S$ from $U$ and add to $T$
7. Return $EnsembleMethod(BaseLearn, T)$

## 3. Active-Decorate

It is beneficial in QBC to use an ensemble method that builds a *diverse* committee, in which each hypothesis is as different as possible, while still maintaining consistency with the training data. Decorate is an ensemble method that explicitly focuses on creating ensembles that are diverse (Melville & Mooney, 2003; Melville & Mooney, 2004). A summary of the Decorate algorithm is provided in the following subsection. We propose a variant of Query by Committee, Active-Decorate, that uses Decorate (in Algorithm 1) to construct committees for sample selection.

To evaluate the expected utility of unlabeled examples, we also used the margins on the examples, as in Abe and Mamitsuka (1998). We generalized their definition, to allow the base classifiers in the ensemble to provide class probabilities, instead of just the most likely class label. Given the class membership probabilities predicted by the committee, the margin is then defined as the difference between the highest and second highest predicted probabilities.

### 3.1. Decorate

This section summarizes the Decorate algorithm; for further details see (Melville & Mooney, 2003; Melville & Mooney, 2004). The approach is motivated by the fact that combining the outputs of multiple classifiers is only useful if they disagree on some inputs (Krogh & Vedelsby, 1995). We refer to the amount of disagreement as the *diversity* of the ensemble, which we measure as the probability that a random ensemble member's prediction on a random example will disagree with the prediction of the complete ensemble.

Decorate was designed to use additional artificially-generated training data in order to generate highly diverse ensembles. An ensemble is generated iteratively, learning one new classifier at each iteration and adding it to the current ensemble. The ensemble is initialized with the classifier trained on the given data. The classifiers in each successive iteration are trained on the original data and also on some artificial data. In each iteration, a specified number of artificial training examples are generated based on a simple model of the data distribution. The category labels for these artificially generated training examples are chosen so as to differ maximally from the current ensemble's predictions. We refer to this artificial training set as the *diversity data*. We train a new classifier on the union of the original training data and the diversity data. If adding this new classifier to the current ensemble increases the ensemble training error, then this classifier is rejected, else it is added to the current ensemble. This process it repeated until the desired committee size is reached or a maximum number of iterations is exceeded. For this study the desired committee size and maximum number of iteration were set to 15 and 50 respectively.

The artificial data is constructed by randomly generating examples using an approximation of the training data distribution. For numeric attributes, a Gaussian distribution is determined by estimating the mean and standard deviation of the training set. For nominal attributes, the probability of occurrence of each distinct value is determined using Laplace estimates from the training data. Examples are then generated by randomly picking values for each feature based on these distributions, assuming attribute independence. In each iteration, the artificially generated examples are labeled based on the current ensemble. Given an example, we compute the class membership probabilities predicted by the current ensemble, replacing zero probabilities with a small $\epsilon$ for smoothing. Labels are then sampled from this distribution, such that the probability of selecting a label is inversely proportional to the current ensemble's predictions.

## 4. Experimental Evaluation

### 4.1. Methodology

To evaluate the performance of ACTIVE-DECORATE, we ran experiments on 15 representative data sets from the UCI repository (Blake & Merz, 1998). We compared the performance of ACTIVE-DECORATE with that of Query by Bagging (QBag), Query by Boosting (QBoost) and DECORATE, all using an ensemble size of 15. J48 decision-tree induction, which is the Weka (Witten & Frank, 1999) implementation of C4.5 (Quinlan, 1993), was used as the base learner for all methods.

The performance of each algorithm was averaged over two runs of 10-fold cross-validation. In each fold of cross-validation, we generated learning curves in the following fashion. The set of available training examples was treated as an unlabeled pool of examples, and at each iteration the active learner selected a sample of points to be labeled and added to the training set. For DECORATE, the examples in each iteration were selected randomly. The resulting curves evaluate how well an active learner orders the set of available examples in terms of utility. At the end of the learning curve, all algorithms see exactly the same training examples.

To maximize the gains of active learning, it is best to acquire a single example in each iteration. However to make our experiments computationally feasible, we choose larger sample sizes for the bigger data sets. In particular, we used a sample size of two for the *primary* dataset, and three for *breast-w, soybean, diabetes, vowel* and *credit-g*.

The primary aim of active learning is to reduce the amount of training data needed to induce an accurate model. To evaluate this, we first define the *target error rate* as the error that DECORATE can achieve on a given dataset, as determined by its error rate averaged over the points on the learning curve corresponding to the last 50 training examples. We then record the smallest number of examples required by a learner to achieve the same or lower error. We define the *data utilization ratio*, as the number of examples an active learner requires to reach the target error rate divided by the number DECORATE requires. This metric reflects how efficiently the active learner is using the data and is similar to a measure used by Abe and Mamitsuka (1998).

Another metric for evaluating an active learner is how much it improves accuracy over random sampling given a fixed amount of labeled data. Therefore, we also compute the percentage reduction in error over DECORATE averaged over points on the learning curve. As mentioned above, towards the end of the learning curve, all methods will have seen almost all the same examples. Hence, the main impact of active learning is lower on the learning curve. To capture this, we report the percentage error reduction averaged over only the 20% of points on the learning curve, where the largest improvements are produced. This is similar to a measure reported by Saar-Tsechansky and Provost (2001). When computing the error reduction of one system over another, the reduction is considered *significant* if the difference in the errors of the two systems averaged across the selected points on the learning curve is determined to be statistically significant according to paired t-tests ($p < 0.05$).

### 4.2. Results

The data utilization of the different active learners with respect to DECORATE is summarized in Table 1. We present the number of examples required for each system to achieve the target error rate and, in parentheses, the data utilization ratio. The smallest num-

*Table 1.* Data utilization with respect to Decorate

| Dataset | Total Size | Decorate | QBag | QBoost | ActiveDecorate | Target Error (%) |
|---|---|---|---|---|---|---|
| Soybean | 615 | 492(1.00) | 267(0.54) | 219(0.45) | **144(0.29)** | 6.59 |
| Vowel | 891 | 840(1.00) | - | - | **477(0.57)** | 3.81 |
| Statlog | 243 | 81(1.00) | 84(1.04) | 89(1.10) | **46(0.57)** | 19.21 |
| Hepatitis | 140 | 39(1.00) | 30(0.77) | 43(1.10) | **23(0.59)** | 16.96 |
| Primary | 305 | 238(1.00) | 202(0.85) | - | **164(0.69)** | 56.23 |
| Heart-c | 273 | 50(1.00) | 57(1.14) | 41(0.82) | **36(0.72)** | 20.97 |
| Sonar | 187 | 125(1.00) | 186(1.49) | 131(1.05) | **99(0.79)** | 18.39 |
| Heart-h | 265 | 49(1.00) | **31(0.63)** | 47(0.96) | 39(0.80) | 19.93 |
| Glass | 193 | 118(1.00) | **97(0.82)** | 101(0.86) | 100(0.85) | 27.00 |
| Diabetes | 691 | 234(1.00) | **114(0.49)** | 393(1.68) | 201(0.86) | 25.09 |
| Lymph | 133 | 27(1.00) | 40(1.48) | 40(1.48) | **24(0.89)** | 22.21 |
| Labor | 51 | 13(1.00) | 26(2.00) | 19(1.46) | **12(0.92)** | 15.14 |
| Iris | 135 | 32(1.00) | 33(1.03) | 125(3.91) | **30(0.94)** | 5.25 |
| Credit-g | 900 | 498(1.00) | **213(0.43)** | 243(0.49) | 495(0.99) | 26.36 |
| Breast-w | 629 | **30(1.00)** | 45(1.50) | 75(2.50) | 39(1.30) | 3.94 |
| *No. of Wins* | | 1 | 4 | 0 | 10 | |

ber of examples needed for each dataset is presented in bold font. On all but one dataset, Active-Decorate produces improvements over Decorate in terms of data utilization. Furthermore, Active-Decorate outperforms both the other active learners on 10 of the datasets. QBag and QBoost were unable to achieve the target error rate on *vowel*; and QBoost also failed to achieve the target error on *primary*. Furthermore, on several datasets QBag and QBoost required more training examples than Decorate. On average, Active-Decorate required 78% of the number of examples that Decorate used to reach the target error. It is important to note that Decorate itself achieves the target error with far fewer examples than available in the full training set, as seen by comparing to the total dataset sizes. Hence, improving on the data utilization of Decorate is a fairly difficult task. Figure 1 presents learning curves that clearly demonstrate the advantage of Active-Decorate. On one dataset, *breast-w*, Active-Decorate requires a few more examples than Decorate. This dataset exhibits a ceiling effect in learning, where Decorate manages to reach the target error rate using only 30 of the 629 available examples, making it difficult to improve on (Figure 2).

Our results on error reductions are summarized in Table 2. The significant values are presented in bold font. We observed that on almost all datasets, Active-Decorate produces substantial reductions in error over Decorate. Furthermore, on 8 of the datasets, Active-Decorate produces higher reductions in error than the other active-learning methods. Depending on the dataset, Active-Decorate produces a wide range of improvements, from moderate (4.16% on *credit-g*) to high (70.68% on *vowel*). On average, Active-Decorate produces a 21.15% reduction in error.

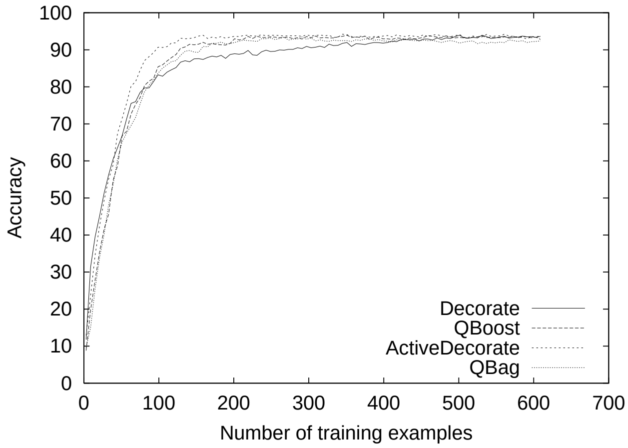


*Figure 1.* Comparing different active learners on *Soybean*.

## 5. Additional Experiments

### 5.1. Measures of Utility

There are two main aspects to any Query by Committee approach. The first is the method employed to construct the committee, and the second is the measure used to rank the utility of unlabeled examples given this committee. Thus far, we have only compared different methods for constructing the committees. Following Abe and Mamitsuka (1998), we ranked unlabeled examples based on the margin of the committee's prediction for the example.

An alternate approach is to use an information theo-

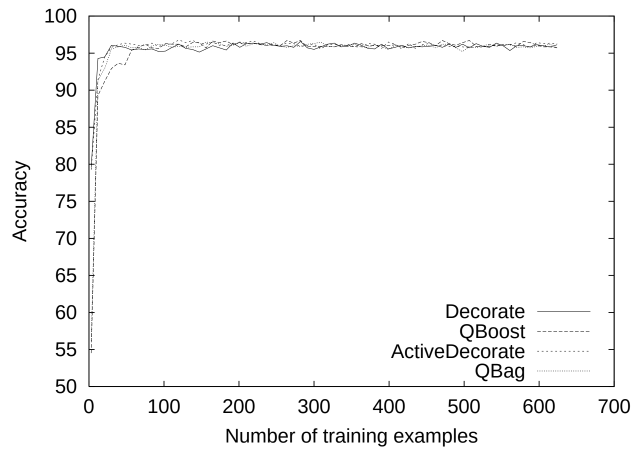


*Figure 2.* Ceiling effect in learning on *Breast-W.*

*Table 2.* Top 20% percent error reduction over Decorate

| Dataset | QBag | QBoost | ActiveDecorate |
|---|---|---|---|
| Soybean | **30.50** | **34.17** | **45.84** |
| Vowel | **22.65** | **42.09** | **70.68** |
| Statlog | **11.31** | **10.34** | **11.43** |
| Hepatitis | **12.13** | **16.68** | **19.31** |
| Primary | **3.23** | 0.43 | **5.74** |
| Heart-c | **15.40** | **19.40** | **12.56** |
| Sonar | 1.88 | **8.09** | **16.47** |
| Heart-h | **16.22** | **14.68** | **12.14** |
| Glass | **10.58** | **16.88** | **15.83** |
| Diabetes | **8.68** | **4.01** | **5.94** |
| Lymph | **19.65** | **28.51** | **18.84** |
| Labor | -2.61 | 12.55 | **36.33** |
| Iris | **22.78** | 1.22 | **22.53** |
| Credit-g | **9.43** | **6.71** | **4.16** |
| Breast-w | **15.12** | **18.89** | **19.51** |
| *Mean* | 13.13 | 15.64 | 21.15 |
| *No. of Wins* | 4 | 3 | 8 |

retic measure such as Jensen-Shannon (JS) divergence to evaluate the potential utility of examples (Cover & Thomas, 1991). JS-divergence is a measure of similarity between probability distributions (Gomez-Lopera et al., 2000). We can utilize this measure if the individual classifiers in the committee provide us with class membership probabilities, rather than just the most likely class. If $P_i(x)$ is the class probability distribution given by the $i$-th classifier for the example $x$ (which we will abbreviate as $P_i$) we can then compute the JS-divergence of an ensemble of size $n$ as:

$$JS(P_1, P_2, \ldots, P_n) = H(\sum_{i=1}^{n} w_i P_i) - \sum_{i=1}^{n} w_i H(P_i)$$

where $w_i$ is the vote weight of the $i$-th classifier in the ensemble;[1] and $H(P)$ is the Shannon entropy of the distribution $P = \{p_j, j = 1, \ldots, K\}$ defined as:

$$H(P) = -\sum_{j=1}^{K} p_j \, log \, p_j$$

Higher values for JS-divergence indicate a greater spread in the predicted class probability distributions, and it is zero if and only if the distributions are identical. We implemented a version of ACTIVE-DECORATE that selects the unlabeled examples with the highest JS-divergence. A similar measure was used for active learning for text categorization by McCallum and Nigam (1998). This measure incorporates more information about the predicted class distribution than using margins, and hence could result in the selection of more informative examples.

To test the effectiveness of using JS-divergence, we ran experiments comparing it to using the margin measure. The experiments were conducted as described in Section 4.1. Table 3 summarizes the results of the comparison of the two measures. All the error reductions are significant ($p < 0.05$), so we only present the better of the two columns in bold font. In terms of data utilization, the methods seem equally matched; JS-divergence performs better than margins on 8 of the 15 datasets. However, on the error reduction metric, using margins outperforms JS-divergence on 11 of the datasets. The results also show, that there are datasets on which JS-divergence and margins achieve the target error rate with comparable number of examples, but the error reduction produced by margins is higher. Figure 3 clearly demonstrates this phenomenon.

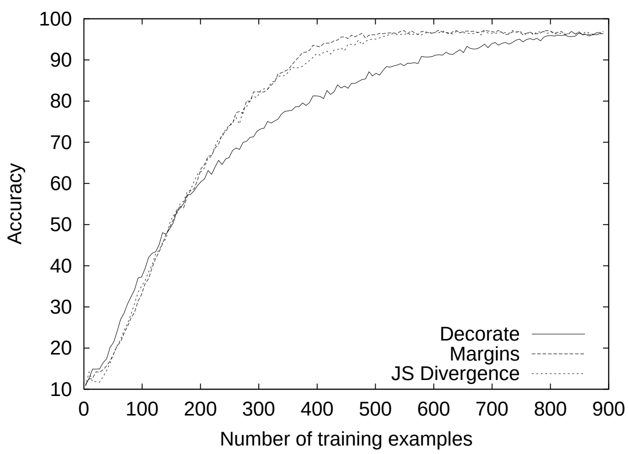


*Figure 3.* Comparing measures of utility: JS Divergence vs Margins on *Vowel.*

Note that while ACTIVE-DECORATE using either mea-

[1] DECORATE uses uniform vote weights, which are normalized to sum to one.

*Table 3.* Comparing measures of utility: Data utilization and top 20% error reduction with respect to Decorate.

| | Data Utilization | | %Error Reduction | |
|---|---|---|---|---|
| Dataset | Margin | JS Div. | Margin | JS Div. |
| Soybean | **144(0.29)** | 369(0.75) | **45.84** | 18.67 |
| Vowel | **477(0.57)** | 525(0.62) | **70.68** | 63.26 |
| Statlog | **46(0.57)** | 76(0.94) | 11.43 | **11.52** |
| Hepatitis | 23(0.59) | **19(0.49)** | **19.31** | 15.90 |
| Primary | **164(0.69)** | 212(0.89) | **5.74** | 3.84 |
| Heart-c | 36(0.72) | **28(0.56)** | 12.56 | **13.97** |
| Sonar | 99(0.79) | **94(0.75)** | 16.47 | **16.71** |
| Heart-h | 39(0.80) | **38(0.78)** | **12.14** | 10.81 |
| Glass | **100(0.85)** | 118(1.00) | **15.83** | 10.46 |
| Diabetes | 201(0.86) | **150(0.64)** | **5.94** | 5.03 |
| Lymph | 24(0.89) | **20(0.74)** | **18.84** | 12.18 |
| Labor | 12(0.92) | **10(0.77)** | **36.33** | 29.77 |
| Iris | **30(0.94)** | 41(1.28) | 22.53 | **23.01** |
| Credit-g | 495(0.99) | **330(0.66)** | **4.16** | 3.91 |
| Breast-w | **39(1.30)** | 45(1.50) | **19.51** | 19.20 |
| *Mean* | 0.78 | 0.83 | 21.15 | 17.22 |
| *# Wins* | 7 | 8 | 11 | 4 |

sure of utility produces substantial error reductions, in general using margins produces greater improvements. Using the JS-divergence measure tends to select examples that would reduce the uncertainty of the predicted class membership probabilities, which helps to improve classification accuracy. On the other hand, using margins focuses more directly on determining the decision boundary. This may account for its better performance. For making cost-sensitive decisions, it is very useful to have accurate class probability estimates (Saar-Tsechansky & Provost, 2001). In such cases, we conjecture that using JS-divergence could be a more effective approach.

### 5.2. Ensemble Diversity

By exploiting artificial examples, the DECORATE algorithm forces the construction of a *diverse* set of hypotheses that are consistent with the training data. We believe that this ensemble diversity is the key to the success of ACTIVE-DECORATE. We ran additional experiments to verify that DECORATE does indeed produce more diverse committees than Bagging or ADABOOST. As in (Melville & Mooney, 2004), we use the disagreement of ensemble members with the ensemble's prediction as a measure of diversity. More precisely, if $C_i(x)$ is the prediction of the $i$-th classifier for the label of $x$; $C^*(x)$ is the prediction of the entire ensemble, then the diversity of the $i$-th classifier on example $x$ is given by:

$$d_i(x) = \begin{cases} 0 & : \text{ if } C_i(x) = C^*(x) \\ w_i & : \text{otherwise} \end{cases}$$

*Table 4.* Comparing ensemble diversity: Win-loss records.

| | Number of Training Examples | | | | |
|---|---|---|---|---|---|
| | 10 | 15 | 20 | 25 | 30 |
| Decorate vs Bagging | 14-1 | 14-1 | 14-1 | 13-2 | 13-2 |
| Decorate vs AdaBoost | 15-0 | 14-1 | 14-1 | 14-1 | 14-1 |

Where $w_i$ is the vote weight of the $i$-th classifier. To compute the diversity of an ensemble of size $n$, on a set of examples of size $m$, we average the above term:

$$\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} d_i(x_j)$$

This measure estimates the probability that a classifier in an ensemble will disagree with the prediction of the ensemble as a whole.

The diversity of each ensemble method was evaluated using 10-fold cross-validation on all 15 datasets. To test performance on varying amounts of data, each system was evaluated on the testing data, after training on increasing subsets of the training data. We focused on points early on the learning curve, where we expect active learning to be most effective. The results (Table 4) are summarized in terms of significant win/loss records; where a win or loss is only counted if the difference in *diversity* (not accuracy) is determined to be significant at the 0.05 level by a paired $t$-test. These results confirm that in most cases DECORATE does indeed produce significantly more diverse ensembles than Bagging or ADABOOST.

### 5.3. Committees for Sample Selection vs. Prediction

All the active learning methods that we have described use committees to determine which examples to select. But in addition to using committees for sample selection, these methods also use the committees for prediction. So we are *not* evaluating which method selects the best queries for the *base learner*, but which combination of sample selection and ensemble method works the best. The fact that ACTIVE-DECORATE performs better than QBag may just be testament to the fact that DECORATE performs better than Bagging. However, we claim that not only does DECORATE produce accurate committees, but the committees produced are also more effective in sample selection. To verify this, we implemented an alternate version of ACTIVE-DECORATE, where at each iteration a committee constructed by Bagging is used to select the examples given to DECORATE. In this way, we separate the evaluation of the method used for sample

*Table 5.* Comparing different ensemble methods for selection for Active-Decorate: Percentage error reduction over Decorate.

| Dataset | Maximum Train Size | Select w/ Bagging | Select w/ AdaBoost | Select w/ Decorate |
|---|---|---|---|---|
| Soybean | 300 | **18.55** | **17.27** | **27.38** |
| Glass | 100 | **6.57** | **4.72** | **8.85** |
| Primary | 200 | 0.2 | **2.46** | **3.75** |
| Statlog | 100 | -1.79 | -1.18 | 1.73 |

selection from the method used for prediction. Similarly, we implemented a version of ACTIVE-DECORATE using ADABOOST to perform the sample selection.

We compared the three methods of sample selection for DECORATE on four of the datasets on which ACTIVE-DECORATE exhibited good performance. We generated learning curves as described in Section 4.1. However, we did not run the learning curve trials until all the available training data was exhausted, since the active learning methods need fewer examples to achieve the target error rates.

The error reductions over DECORATE averaged across all the points on the learning curve are presented in Table 5.[2] The significant error reductions are shown in bold. The table also includes the maximum training set size, which corresponds to the last point on the learning curve. The results show that, on 3 of the 4 datasets, using any of the ensemble sample selection methods in conjunction with DECORATE produces better results than DECORATE. Furthermore, DECORATE committees select more informative examples for training DECORATE than the other committee sample selection methods. These trends are clearly seen in Figure 4. A more extensive study needs to be done to add to these preliminary results. It would also be interesting to run similar experiments, using DECORATE ensembles to pick examples for training Bagging, ADABOOST, or J48.

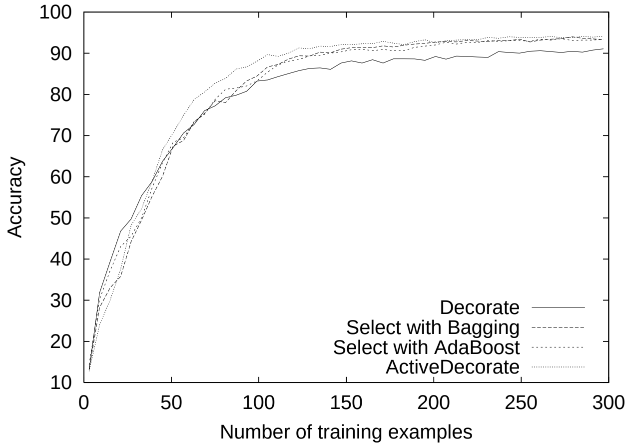


*Figure 4.* Comparing different ensembles methods for selecting samples for DECORATE on *Soybean*.

## 6. Related Work

In their QBC approach, Dagan and Engelson (1995) measure the utility of examples by *vote entropy*, which is the entropy of the class distribution, based on the majority votes of each committee member. McCallum and Nigam (1998) showed that *vote entropy* does not perform as well as JS-divergence for pool-based sample selection. Another recently developed effective committee-based active learner is Co-Testing (Muslea et al., 2000); however, it requires two redundant *views* of the data. Since most data sets do not have redundant views, Co-Testing has rather limited applicability. Another general approach to sample selection is *uncertainty sampling* (Lewis & Catlett, 1994); however, this approach requires a learner that accurately estimates the uncertainty of its decisions, and tends to over-sample the boundaries of its current incomplete hypothesis (Cohn et al., 1994). Finally, *expected-error reduction* methods for active learning (Cohn et al., 1996; Roy & McCallum, 2001; Zhu et al., 2003) attempt to statistically select training examples that are expected to minimize error on the actual test distribution. This approach has the advantage of avoiding the selection of outliers whose labeling will not improve accuracy on typical examples. However, this method is computationally intense, and must be carefully tailored to a specific learning algorithm (e.g. naive Bayes); and hence, cannot be used to select examples for an arbitrary learner. Active meta-learners like Query by Bagging/Boosting and ACTIVE-DECORATE have the advantage of being able to select queries to improve any learner appropriate for a given domain.

## 7. Conclusion

ACTIVE-DECORATE is a simple, yet effective approach to active learning. Experimental results show that, in general, this approach leads to more effective sample selection than Query by Bagging and Query by Boosting. On average, ACTIVE-DECORATE requires 78% of the number of training examples required by DECORATE with random sampling. Additional experiments support the hypothesis that for small training sets DECORATE produces more diverse ensembles than

[2]These results are not directly comparable to those in Table 2.

Bagging or AdaBoost. We believe this increased diversity is the key to Active-Decorate's superior performance.

Our results also show that using JS-divergence to evaluate the utility of examples is less effective for improving classification accuracy than using margins. JS-divergence may be a better measure when the objective is improving class probability estimates. This is an interesting area for future work.

## Acknowledgments

This research was supported by DARPA grants F30602-01-2-0571 and HR0011-04-1-007. We would like to thank Maytal Saar-Tsechansky and the anonymous reviewers for valuable comments.

## References

Abe, N., & Mamitsuka, H. (1998). Query learning strategies using boosting and bagging. *Proc. of 15th Intl. Conf. on Machine Learning (ICML-98)* (pp. 1–10).

Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, *15*, 201–221.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*, 129–145.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York, NY: Wiley.

Dagan, I., & Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. *Proc. of 12th Intl. Conf. on Machine Learning (ICML-95)* (pp. 150–157). San Francisco, CA: Morgan Kaufmann.

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

Gomez-Lopera, J. F., Martinez-Aroza, J., Robles-Perez, A. M., & Roman-Roldan, R. (2000). An analysis of edge detection by using the Jensen-Shannon divergence. *Journal of Mathematical Imaging and Vision*, *13*, 35–56.

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation and active learning. *Advances in Neural Information Processing Systems 7*.

Lewis, D. D., & Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. *Proc. of 11th Intl. Conf. on Machine Learning (ICML-94)* (pp. 148–156). San Francisco, CA: Morgan Kaufmann.

Liere, R., & Tadepalli, P. (1997). Active learning with committees for text categorization. *Proc. of 14th Natl. Conf. on Artificial Intelligence (AAAI-97)* (pp. 591–596). Providence, RI.

McCallum, A., & Nigam, K. (1998). Employing EM and pool-based active learning for text classification. *Proc. of 15th Intl. Conf. on Machine Learning (ICML-98)*. Madison, WI: Morgan Kaufmann.

Melville, P., & Mooney, R. (2003). Constructing diverse classifier ensembles using artificial training examples. *Proc. of 18th Intl. Joint Conf. on Artificial Intelligence* (pp. 505–510). Acapulco, Mexico.

Melville, P., & Mooney, R. J. (2004). Creating diversity in ensembles using artificial data. *Information Fusion: Special Issue on Diversity in Multiclassifier Systems*.

Melville, P., Saar-Tsechansky, M., Provost, F., & Mooney, R. (2004). Active feature acquisition for classifier induction. Submitted for review. Available at http://www.cs.utexas.edu/users/ml/publication/.

Muslea, I., Minton, S., & Knoblock, C. A. (2000). Selective sampling with redundant views. *Proc. of 17th Natl. Conf. on Artificial Intelligence (AAAI-2000)* (pp. 621–626).

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo,CA: Morgan Kaufmann.

Roy, N., & McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. *Proc. 18th Intl. Conf. on Machine Learning* (pp. 441–448). Morgan Kaufmann, San Francisco, CA.

Saar-Tsechansky, M., & Provost, F. J. (2001). Active learning for class probability estimation and ranking. *Proc. of 17th Intl. Joint Conf. on Artificial Intelligence (IJCAI-2001)* (pp. 911–920).

Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. *Proc. of the ACM Workshop on Computational Learning Theory*. Pittsburgh, PA.

Witten, I. H., & Frank, E. (1999). *Data mining: Practical machine learning tools and techniques with Java implementations*. San Francisco: Morgan Kaufmann.

Zhu, X., Lafferty, J., & Ghahramani, Z. (2003). Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. *Proc. of the ICML Workshop on the Continuum from Labeled to Unlabeled Data* (pp. 58–65).