Machine Learning, 54, 153–178, 2004


# Active Sampling for Class Probability Estimation and Ranking

MAYTAL SAAR-TSECHANSKY maytal.saar-tsechansky@bus.utexas.edu
*Department of Management Science and Information Systems, Red McCombs School of Business, The University of Texas at Austin, Austin, Texas 78712, USA*

FOSTER PROVOST fprovost@stern.nyu.edu
*Department of Information Operations & Management Sciences, Leonard N. Stern School of Business, New York University, 44 West Fourth Street, New York, NY 10012, USA*

**Editor:** Douglas Fisher

**Abstract.** In many cost-sensitive environments class probability estimates are used by decision makers to evaluate the expected utility from a set of alternatives. Supervised learning can be used to build class probability estimates; however, it often is very costly to obtain training data with class labels. Active learning acquires data incrementally, at each phase identifying especially useful additional data for labeling, and can be used to economize on examples needed for learning. We outline the critical features of an active learner and present a sampling-based active learning method for estimating class probabilities and class-based rankings. BOOTSTRAP-LV identifies particularly informative new data for learning based on the variance in probability estimates, and uses weighted sampling to account for a potential example's informative value for the rest of the input space. We show empirically that the method reduces the number of data items that must be obtained and labeled, across a wide variety of domains. We investigate the contribution of the components of the algorithm and show that each provides valuable information to help identify informative examples. We also compare BOOTSTRAP-LV with UNCERTAINTY SAMPLING, an existing active learning method designed to maximize classification accuracy. The results show that BOOTSTRAP-LV uses fewer examples to exhibit a certain estimation accuracy and provide insights to the behavior of the algorithms. Finally, we experiment with another new active sampling algorithm drawing from both UNCERTAINTY SAMPLING and BOOTSTRAP-LV and show that it is significantly more competitive with BOOTSTRAP-LV compared to UNCERTAINTY SAMPLING. The analysis suggests more general implications for improving existing active sampling algorithms for classification.

**Keywords:** active learning, cost-sensitive learning, class probability estimation, ranking, supervised learning, decision trees, uncertainty sampling, selective sampling

## 1. Introduction

Supervised classifier learning requires data with class labels. In many applications, procuring class labels can be costly. For example, to train diagnostic models experts may need to read many historical cases. To train document classifiers experts may need to read many documents and assign them labels. To train customer response models, consumers may have to be given costly incentives to reveal their preferences.

Active learning acquires labeled data incrementally, using the model learned "so far" to select particularly helpful additional training examples for labeling. When successful,

active learning methods reduce the number of instances that must be labeled to achieve a particular level of accuracy. Most existing methods and particularly empirical approaches for active learning address classification problems—they assume the task is to assign cases to one class (from a fixed set of classes).

Many applications, however, require more than simple classification. In particular, probability estimates are central in decision theory, allowing a decision maker to incorporate costs/benefits for evaluating alternatives. For example, in targeted marketing the estimated probability that a customer will respond to an offer is combined with the estimated profit (Zadrozny & Elkan, 2001) to evaluate various offer propositions. Other applications require ranking cases by the likelihood of class membership, to improve the response rate to offer propositions, or to add flexibility for user processing.[1] For example, documents can be ranked by their probability of being of interest to the user, and offers to consumers may be presented/proposed in order of the probability of purchase or of the expected benefit to the seller. For these reasons we focus on learning class probability estimation (CPE) models.

In this paper we consider active learning to produce accurate CPEs and class-based rankings from fewer labeled training examples. We assume a (unspecified) cost is associated with acquiring labels specifically rather than with the generation or the obtaining of training examples. Figure 1 shows the desired behavior of an active learner. The horizontal axis represents the information needed for learning, i.e., the number of labeled training examples, and the vertical axis represents the error rate of the probabilities produced by the learned model. Each *learning curve* shows how error rate decreases as more training data are used. The upper curve represents the decrease in error from sampling examples randomly for labeling and training; the lower curve represents sampling actively. The two curves form a "banana" shape: very early on, the curves are comparable because a model is not yet available to guide the active sampling. The active sampling curve soon accelerates, because of the careful choice of training examples. Given enough data, random sampling eventually catches up.

We introduce a new sampling-based active learning technique, BOOTSTRAP-LV, for learning CPEs. BOOTSTRAP-LV uses bootstrap samples (Efron & Tibshirani, 1993) of available

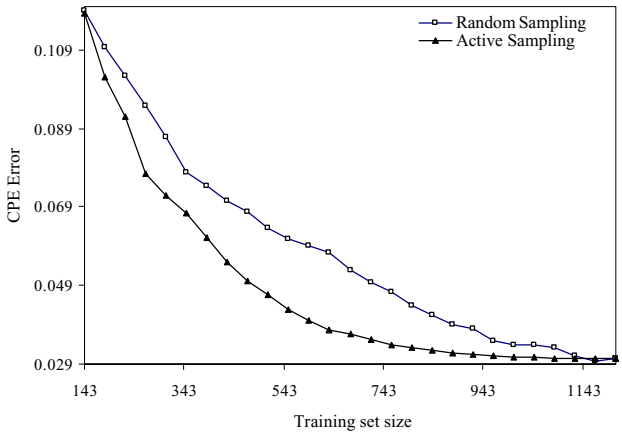


*Figure 1.* Learning curves for active sampling vs. random sampling.

labeled data to examine the variance in the probability estimates for not-yet-labeled data, and employs a weight-sampling procedure to select particularly informative examples for labeling and learning. We show empirically across a range of data sets that BOOTSTRAP-LV decreases the number of labeled instances needed to achieve accurate probability estimates, or alternatively that it increases the accuracy of the probability estimates for a fixed number of training data. An analysis of the algorithm's characteristics and performance reveals the contributions of its components. The results of the analysis lead to the design of a new algorithm for active sampling that is more competitive with BOOTSTRAP-LV than a popular existing method and has computational advantages over BOOTSTRAP-LV. This final result further demonstrates how the components of the BOOTSTRAP-LV algorithm contribute to its efficacy and highlights why existing algorithms do not perform well for CPE.

## 2. Active learning and the Bootstrap-LV algorithm

The fundamental notion of active sampling has a long history in machine learning. To our knowledge, the first to discuss it explicitly were Simon and Lea (1974) and Winston (1975). Simon and Lea describe how machine learning is different from other types of problem solving, because learning involves the simultaneous search of two spaces: the hypothesis space and the instance space. The results of searching the hypothesis space can affect how the instance space will be sampled. Porter and Kibler (1986) address the symbiosis between learning and problem solving, and propose a learning apprentice system that learns problem-solving rules. Their method reduces reliance on the teacher to provide examples by acting only when the system is unable to determine what to do next. Winston (1975) discusses how the best examples to select next for learning are "near misses," instances that miss being class members for only a few reasons. Subsequently, theoretical results showed that the number of training data can be reduced substantially if they are selected carefully (Angluin, 1988). The term *active learning* was coined later to describe induction where the algorithm controls the selection of potential unlabeled training examples (Cohn et al., 1994).

A generic algorithm for active learning is shown in figure 2. A learner first is applied to an initial set $L$ of labeled examples (usually selected at random or provided by an expert). Subsequently, sets of $M$ examples are selected in phases from a set of unlabeled examples

**Input**: an initial labeled set $L$, an unlabeled set $UL$, an inducer $I$,
a stopping criterion, and an integer $M$ specifying the number of actively selected examples in each phase.

1 While stopping criterion not met
/* perform next phase: */
2 Apply inducer $I$ to $L$
3 For each example $\{x_i \mid x_i \in UL\}$ compute $ES_i$, the effectiveness score
4 Select a subset $S$ of size M from $UL$ based on $ES_i$
5 Remove $S$ from $UL$, label examples in $S$, and add $S$ to $L$

**Output:** estimator $E$ induced with $I$ from the final labeled set $L$

*Figure 2*. Generic active learning algorithm.

*UL*, until some predefined condition is met (e.g., the labeling budget is exhausted). If *UL* is very large a subset of randomly sampled examples from *UL* may be used as a substitute for the complete set (Roy & McCallum, 2001). In each phase, each candidate example $x_i \in UL$ is assigned an effectiveness score $ES_i$ based on an objective function, reflecting its contribution to subsequent learning. Examples then are selected for labeling based on their effectiveness scores. Often, multiple examples, rather than a single example, are selected in each phase due to computational constraints. Once examples are selected, their labels are obtained (e.g., by querying an expert) before being added to $L$, to which the learner is applied next.

The objective of active learning is to select examples that will reduce the generalization error of the model the most. The *generalization error* is the expected error across the entire example space. Therefore when evaluating a training example an optimal active learning approach must evaluate the expected reduction in generalization error if the example were to be added to the training set from which the model would be induced (Roy & McCallum, 2001). The example that is expected to reduce the generalization error the most should be added to the training set. Unfortunately, as we discuss below, assessing the expected reduction of CPE generalization error is not straightforward.

We are interested in an active learning scheme that will apply to arbitrary learners, thus computational considerations may prohibit us from examining the models resulting from adding each potential unlabeled example to the training set (as prescribed by Roy and McCallum (2001)). We therefore resort to an indirect estimation of potential training examples' informative value. Also, we consider the potential of each training example to help improve the estimation of *other* examples in the space, which we describe in detail below.

Given the generic framework presented in figure 2, BOOTSTRAP-LV embodies a particular instantiation of steps 3 and 4. The description we provide here pertains to binary classification problems.

Since our goal is to reduce the class probability estimation (CPE) error, it is useful to understand the error's sources. A model's estimation $\hat{f}(x \mid T)$ for a particular input $x$ depends upon the sample $T$ from which the model is induced, and therefore can be treated as a random variable. Let $f(x)$ be the underlying function describing the probability of class membership for a case described by input $x$. One indication of the *quality* of the current class probability estimate $\hat{f}(x \mid T)$ for example $x$ given a training set $T$ is the expected estimation (absolute) error, reflecting the discrepancy between the estimated probability and the true probability, i.e., $| f(x) - \hat{f}(x \mid T)|$. We may infer from the discrepancy whether additional information is needed to improve the model's estimation. Note that unfortunately in our inductive learning setting we typically do not know the class *probability*, $f(x)$, for an input $x$, even when we do know the true class of a particular instance described by $x$.

A common formulation (Friedman, 1997) of the estimation error decomposes the expected squared estimation error into the sum of two terms: $E_T[(f(x) - \hat{f}(x \mid T)]^2 = E_T[\hat{f}(x \mid T) - E_T\hat{f}(x \mid T)]^2 + [f(x) - E_T\hat{f}(x \mid T)]^2$; $E_T(\cdot)$ represents expectation across training sets $T$. The first term in the sum is referred to as the *variance* of the estimation and reflects the sensitivity of the estimation to the training sample. The second term is referred to as the (squared) *bias,* reflecting the extent to which the induced model can approximate the target function $f(x)$ (Friedman, 1997). Calculating both the estimation error and the estimation bias requires knowing the actual probability function, $f(x)$, which as mentioned

**Input**: an initial labeled set $L$ sampled at random, an unlabeled set $UL$, an inducer $I$, a stopping criterion, and a sample size $M$.

2 for (s=1;until stopping criterion is met; s++)

3 Generate $k$ bootstrap subsamples $B_j$, $j = 1,\ldots,k$ from $L$

4 Apply inducer $I$ on each subsample $B_j$ and induce estimator $E_j$

5 For all examples $\{x_i \mid x_i \in UL\}$ compute $D_s(x_i) = \frac{\left\{\sum_{j=1}^{k}\left[\left(p_j(x_i) - \bar{p}_i\right)^2\right]\right\} / \bar{p}_{i,\min}}{R}$

6 Sample from the probability distribution $D_s$, a subset N of $M$ examples from $UL$ without replacement

7 Remove N from $UL$, label examples in N, and add them to $L$

8 end for

**Output:** estimator $E$ induced with $I$ from $L$

*Figure 3*. The BOOTSTRAP-LV algorithm.

above is not available for an inductive learning algorithm to consider. Therefore it is impossible to compute them directly. The estimation variance, however, reflects a behavior of the estimation *procedure* without reference to the underlying probability function. Therefore, in order to reduce the estimation error the BOOTSTRAP-LV algorithm estimates and then tries to reduce the estimation variance. The estimation variance for a certain input is referred to as the "local variance" (LV) to differentiate it from the model's expected variance over the entire input space. We ignore the bias, or alternatively assume the bias is zero.

The BOOTSTRAP-LV algorithm, shown in figure 3, first estimates the local variance of each potential training example. If the LV is high, the algorithm infers that this input is not well captured by the model given the available data. The local variance also reflects the potential error reduction if this variance were reduced as more examples become available for the learner. BOOTSTRAP-LV then employs the LV estimations together with a specialized sampling procedure to identify the examples that are particularly likely to reduce the *average* estimation error across the entire example space (i.e., the generalization error) the most. We first describe the estimation of the local variance. We then will discuss the sampling procedure.

Given that an efficient closed-form estimation of the local variance may not be obtained for arbitrary learners, we estimate it empirically. The variance stems from the estimation being induced from a random sample. We therefore emulate a series of samples by generating a set of $k$ *bootstrap* subsamples (Efron & Tibshirani, 1993) $B_j$, $j = 1, \ldots, k$ from $L$. We generate a set of models by applying the inducer $I$ to each bootstrap sample $B_j$, resulting in $k$ estimators $E_j$, $j = 1, \ldots, k$. To calculate the estimated variance, for each example in $x_i \in UL$, we estimate the variance among CPEs predicted by the estimators $\{E_j\}$. Finally, each example in $x_i \in UL$ is assigned an effectiveness score that is proportional to its local variance.

The local variance provides an indication of the potential error reduction for each individual training example. However, it does not necessarily provide an indication of how much would be learned about other examples in the space. Recall that our objective is to reduce the generalization error by training a model with fewer, particularly informative examples; a training example therefore must affect the estimation error of other examples in the example space. It may be that an example with a very high variance is not well captured by the model, but is an outlier, not similar to any other examples in the space.

Many existing active learning algorithms select examples in order of their effectiveness score, such that the examples with the highest scores are selected for labeling first. Let us refer to this approach as *Direct Selection*. Direct selection ignores information about how the class probability estimation error of other examples in the space may be affected by adding the example to the training set. This information, however, is essential to evaluate the expected effect an example may have on the generalization error.

Random sampling is often referred to in the active learning literature as "non-informed" learning (e.g., Cohn, Atlas, & Ladner, 1994; Lewis & Gale, 1994). Nevertheless, random sampling is powerful because it allows the incorporation of information about the distribution of examples even when this information is not known explicitly. For example, consider the case when examples for labeling are sampled at random. An example may inform the learning about other examples in the space if it is similar to these examples. Consider a set of similar examples. With random sampling, the larger this set the more likely it is that an example from this set is sampled, providing information about a larger number of examples. Note that this property is obtained without having to capture explicitly how examples are similar to each other.

In order to reduce the error across the example space, BOOTSTRAP-LV incorporates sampling into the selection of training examples by *weight sampling* examples for labeling. In particular, the probability of each example to be sampled is proportional to its effectiveness score, i.e., its local variance. Specifically, the distribution from which examples are sampled is given by $D_s(x_i) = \frac{\{\sum_{j=1}^{k}[(p_j(x_i)-\bar{p}_i)^2]\}/\bar{p}_{i,\min}}{R}$, where $p_j(x_i)$ denotes the estimated probability an estimator $E_j$ assigns to the event that example $x_i$ belongs to one of the two classes (the choice of performing the calculation for either class is arbitrary because the variance for both classes is equal); $\bar{p}_i = \frac{\sum_{j=1}^{k} p_j(x_i)}{k}$; $\bar{p}_{i,\min}$ is the average probability estimation assigned to the minority class by the estimators $\{E_j\}$, and $R$ is a normalizing factor $R = \sum_{i=1}^{size(UL)}\{\sum_{j=1}^{k}[(p_j(x_i) - \bar{p}_i)^2]\}/\bar{p}_{i,\min}$, so that $D_s$ is a distribution.

There is one additional technical point of note. Consider the case where the classes are not represented equally in the training data. When high variance exists in regions of the domain for which the minority class is assigned high probability, it is likely that the region is relatively better understood than regions with *the same variance* but for which the majority class is assigned high probability. In the latter case, the class probability estimation may be exhibiting high variance due simply to lack of representation of the minority class in the training data, and would benefit from sampling more from this subset of examples. Therefore the estimated variance is divided by the average value of the minority-class probability estimates $\bar{p}_{i,\min}$. The minority class is determined once from the initial random sample.

## 3. Related work

Cohn, Ghahramani, and Jordan (1996) propose an active learning approach for statistical learning models, generating queries (i.e., training examples) from the input space to be used as inputs to the learning algorithm. This approach directly evaluates the effectiveness score, i.e., the informative contribution of each example to the learning task. At each phase the expectation of the variance of the model over the example space is used to generate

the example that minimizes this variance. Since it requires a computation in closed form of the learner's variance, this approach is impracticable for arbitrary models. In addition, queries are generated whereas here we are interested in identifying informative examples from an existing set of available unlabeled examples (a subset of the set of possible queries).

When an efficient closed-form estimation of the expected generalization error is not available, the models that result from adding each potential training example to the training set can be induced in order to estimate the expected changes in generalization error. Roy and McCallum (2001) propose this approach for building classifiers. At each phase they update the current model with each additional training example for each possible label and calculate an effectiveness score, measured as class entropy, as an estimate of the improvement in classification error. They then select the example bringing about the greatest expected reduction of entropy. The algorithm was shown to be effective, reducing the number of examples needed to obtain a certain level of accuracy. For many learning algorithms, however, the induction of a new model for each possible training example may be prohibitively expensive. A critical requirement for their approach, therefore, allowing it to be computationally tractable, is that the learning algorithm allow efficient incremental updates of the model, such as the naïve Bayes algorithm used to classify text documents in their experiments (Roy & McCallum, 2001).

When an efficient closed-form computation of the error or incremental model updating is not possible, various active learning approaches compute alternative effectiveness scores. For example, the QUERY BY COMMITTEE (QBC) algorithm (Seung, Opper, & Smopolinsky, 1992) was proposed to select training examples actively for training a binary classifier. Examples are sampled at random, generating a "stream" of potential training examples, and each example is considered informative (and thus is labeled) if classifiers sampled from the current version space disagree regarding its class prediction. The QBC algorithm employs disagreement as a binary effectiveness score, designed to capture whether or not uncertainty exists regarding class prediction given the current labeled examples.

McCallum and Nigam (1998) note that a disadvantage of the "stream-based" QBC approach lies in the decision as to whether to label an example being "made on each document (i.e., example) individually, irrespective of the alternatives." An attractive method would be to compare the estimation uncertainty of all the unlabeled training examples, allowing one to select at each phase the example(s) with the largest classification uncertainty. Various other approaches have been developed within the Query By Committee framework that identify informative examples for constructing classifiers and which use a variety of measures that quantify the level of uncertainty or the likelihood of classification error given the current labeled data. In particular, these effectiveness scores quantify the estimated informative value of each example and thereby obtain a ranking of the examples' informative values. Subsequently the example(s) with the highest effectiveness score(s) is (are) selected.

For instance, Abe and Mamitsuka (1998) use bagging and boosting to generate a committee of classifiers and quantify disagreement as the margin (i.e., the difference in weight assigned to either class). Examples with the minimum margin are selected for labeling. The final classifier is composed of an ensemble of classifiers whose votes are used for class prediction. UNCERTAINTY SAMPLING (Lewis & Gale, 1994) was designed to select informative

examples to construct binary classifiers by adopting the uncertainty notion underlying the QBC approach, but instead of generating a committee of hypotheses to estimate uncertainty the algorithm employs a single probabilistic classifier. Examples whose probabilities of class membership are closest to 0.5 are selected for labeling first. UNCERTAINTY SAMPLING has several attractive properties, which we return to below.

These methods are not designed to improve CPEs or rankings, which is our concern in this paper; as indicated by their effectiveness scores, most are designed to improve classification. In addition, as opposed to the approach we propose in this paper, these methods do not incorporate the effect of a potential additional training example on other examples in the example space. Particularly, they disregard the potential of a training example to reduce the error of the estimation for other examples. Examples for which the current estimation is most uncertain may have no significant contribution to reducing the estimation error of other examples in the instance space. The failure to account for this effect was noted by Argamon-Engelson and Dagan (1999) as well as by McCallum and Nigam (1998) who proposed to incorporate an instance density measure explicitly into the effectiveness score, where the density measure reflects how similar are other examples in the space to the one examined. The underlying assumption is that the proposed similarity measure captures the relative effect an example would have on reducing the classification error of other examples in the space. The approach was shown to be effective in selecting informative examples for document classification. Yet the proposed density measure is specific to document items, where similarity measures are available (e.g., TF/IDF). It is not clear what an appropriate density measure would be for an arbitrary domain.[2]

Our approach uses weight sampling, by which we argue it implicitly incorporates properties of the domain to support the selection of examples more likely to be informative regarding other examples in the space. Note that weight sampling also is employed in the AdaBoost algorithm (Freund & Shapire, 1996) on which Iyengar, Apte, and Zhang (2000) base their active learning approach. Their algorithm results in an ensemble of classifiers where weight sampling is used both to select examples from which successive classifiers in the ensemble are generated as well as to select examples for labeling. Iyengar et al. note that better results were obtained when examples were sampled compared to when examples are selected by order of their error measure. They propose to study this phenomenon further and hypothesize that sampling allows their approach to avoid selecting the same examples repeatedly. We argue that in addition weight sampling acts to increase the likelihood of selecting examples that are particularly informative for reducing the generalization error. As we discuss in the previous paragraph, selecting examples should address the relevance of each training example to other examples in order to identify examples that will better decrease the average estimation error (i.e., the generalization error). Moreover, whereas the domain-specific approach of McCallum and Nigam modeled the example space explicitly and incorporated a measure of space density into the effectiveness score (McCallum & Nigam, 1998), the weight-sampling mechanism can be applied seamlessly for arbitrary domains.

In sum, BOOTSTRAP-LV employs an effectiveness score that identifies examples whose CPE has large variance with respect to the training data used. It uses this measure to indicate the potential improvement in class probability estimation error, rather than classification

accuracy. BOOTSTRAP-LV estimates local variance empirically, enabling its computation with an arbitrary modeling scheme. Lastly, we use a sampling mechanism to complement the selection of examples for learning. We argue that weight sampling can help account for the informative value an example confers to other examples in the space.

## 4. Experimental evaluation

The experiments we describe here examine BOOTSTRAP-LV's performance over a range of domains in order to assess its general ability to identify particularly useful examples for learning. In Section 4.1 we present our experimental setting. Sections 4.2 and 4.3 present and discuss our results when learning simple probability estimation trees, and when learning bagged probability estimation trees, respectively. We discuss additional evaluation measures in Section 4.4. In Section 4.5 we compare BOOTSTRAP-LV with UNCERTAINTY SAMPLING, an active learning approach designed to improve classification accuracy, in order to provide insight into the operation of the algorithm and its advantage compared to existing approaches. Finally, we present experiments with a new active learning algorithm inspired by the empirical investigation that provide further insight into the elements of the BOOTSTRAP-LV algorithm.

### *4.1. Experimental setting*

We applied BOOTSTRAP-LV to 20 data sets, 17 from the UCI machine learning repository (Blake & Merz 1998) and 3 used previously to evaluate rule-learning algorithms (Cohen & Singer, 1999). Data sets with more than two classes were mapped into two-class problems. For these data sets the minority class was associated with one class and all remaining classes were mapped to the second class.

For these experiments we use tree induction to produce class probability estimates.[3] In particular, for the experiments presented here, the underlying probability estimator is a Probability Estimation Tree (PET), an unpruned C4.5 decision tree (Quinlan, 1993) for which the Laplace correction (Cestnik, 1990) is applied at the leaves. Not pruning and using the Laplace correction had been shown to improve the CPEs produced by PETs (Bauer & Kohavi, 1999; Provost, Fawcett, & Kohavi, 1998; Provost & Domingos, 2000; Perlich, Provost, & Simonoff, 2003).

As models are learned from more data, performance typically improves as a learning curve; BOOTSTRAP-LV aims to obtain comparable performance with fewer labeled data (recall figure 1). To evaluate the predictive quality of the CPE models induced by BOOTSTRAP-LV it would be desirable to compare against the true class probability values, for example, computing the mean absolute error with respect to the actual probabilities. However, these data sets contain only class membership information; the true class probabilities are unknown. Instead, we compare the probabilities assigned by the model induced with BOOTSTRAP-LV at each phase with those assigned by a "best" estimator, $E_B$, as surrogates to the true probabilities, where $E_B$ is induced from the entire set of available training examples $L \cup UL$ (where the labels of all examples are known to us). In particular, we induce $E_B$ using bagged PETs, which have shown to produce superior probability estimates

compared to individual PETs (Bauer & Kohavi, 1999; Provost, Fawcett, & Kohavi, 1998; Provost & Domingos, 2000; Perlich, Provost, & Simonoff, 2003). We then calculate the mean absolute error, denoted BMAE (Best-estimate Mean Absolute Error), for an estimator $E$ with respect to $E_B$'s estimation. BMAE is given by $BMAE = \frac{\sum_{i=1}^{N} |p_{E_B}(x_i) - p_E(x_i)|}{N}$, where $p_{E_B}(x_i)$ is the estimated probability given by $E_B$; $p_E(x_i)$ is the probability estimated by $E$, and $N$ is the number of (test) examples examined.

We compare the performance of BOOTSTRAP-LV with a method, denoted RANDOM, where estimators are induced with the same inducer and the same training-set size, but for which examples are sampled at random. We compare across different sizes of the labeled set L. In order not have very large sample sizes, $M$, for large data sets and very small ones for small data sets, we applied different numbers of sampling phases for different data sets, varying between 10 and 30; for a given data set at each phase the same number of examples was added to L. Results are averaged over 10 random, three-way partitions of a data set into an initial labeled set, an unlabeled set, and a test set against which the estimators are evaluated. For control, the same partitions were used by both RANDOM and BOOTSTRAP-LV.

The banana curve in figure 4 shows the relative performance for the *Car* data set. The figure shows that the error of the estimator induced with BOOTSTRAP-LV decreases faster initially, exhibiting lower error for fewer examples. This demonstrates that examples actively added to the labeled set are more informative (on average), allowing the inducer to construct a better estimator for a certain number of training examples. Note that for visibility the algorithms' performances with the initial labeled set (for which all algorithms perform identically) are not shown.

Evaluations of active learning algorithms often present only the initial part of the learning curve to demonstrate the efficacy of the algorithm. We summarize the comparative performance of the competing algorithms instead across the entire leaning curve. In particular, the objective of BOOTSTRAP-LV is to enable learning with fewer examples in order to obtain a certain level of CPE accuracy. For each data set we calculate a set of measures

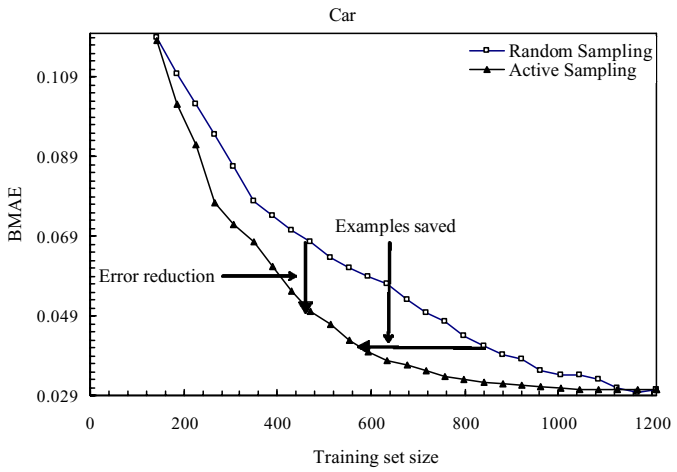


*Figure 4*. Learning behavior of BOOTSTRAP-LV and RANDOM for the Car data set.

pertaining to the saving obtained with BOOTSTRAP-LV in terms of the number of examples that did not need to be labeled when using BOOTSTRAP-LV instead of RANDOM. The number of examples saved by BOOTSTRAP-LV for a certain performance level is demonstrated in figure 4. For each sampling phase of the algorithm we calculate the difference in the number of examples needed by BOOTSTRAP-LV to obtain the exhibited error level and the number needed by RANDOM to obtain the same error level. We calculate the average saving across all sampling phases, referred to as "average saving," as well as the saving as a percentage of the number of examples needed by RANDOM (i.e., the percentage of examples saved if BOOTSTRAP-LV is used instead of RANDOM), referred to as "average relative saving." For instance, in the Car domain (figure 4) the average saving is 155 examples and the average relative saving is 23.3% of the examples needed by RANDOM

Because of the natural banana shape of the learning curves, even for the ideal case the performance of estimators induced from any two samples cannot be considerably different at the final sampling phases, as most of the available examples have been used by both sampling methods and therefore the samples obtained by the methods become increasingly similar. An average across all phases provides an indication of whether BOOTSTRAP-LV produces superior estimations. However, it is even more telling to examine the improvement at the "fat" part of the banana (where the benefit of active learning is concentrated). To allow a stable assessment, instead of presenting the saving exhibited by BOOTSTRAP-LV in the single, best sampling phase, we present the average saving of the top 20% of the sampling phases. We call this "top-20% saving." We also present the top-20% saving as a percentage of the examples needed by RANDOM, referred to as "top-20% relative saving." For instance, in the Car domain the top-20% saving is 281 examples or 35.4% of the examples needed by RANDOM. We also present the percentage of the sampling phases in which a saving was obtained, that is, where RANDOM needed more examples to obtain the error level exhibited by BOOTSTRAP-LV. We refer to this as the percentage of phases with savings.

Finally, for each data set we also present the error reduction achieved by BOOTSTRAP-LV with respect to RANDOM for the same number of training examples. This also is demonstrated in figure 4. We calculate the average error reduction for the 20% of the phases in which the largest error reduction is observed, and we refer to the latter as the top-20% error reduction. For the Car domain the top-20% error reduction is 31.3%.

### *4.2. Results: Bootstrap-LV versus random sampling*

For some data sets BOOTSTRAP-LV exhibits even more dramatic results than those presented for the Car data set above; figure 5 shows results for the Pendigits data set (the most impressive "win"). BOOTSTRAP-LV achieves its almost minimal level of error at about 4000 examples. RANDOM requires more than 9300 examples to obtain this error level. It is important to note that an active learning algorithm's performance is particularly interesting in the initial sampling phases demonstrating the performance that can be obtained for a relatively small portion of the data and therefore a small labeling cost. Similarly to the results presented in figure 4, in the initial phases the error exhibited by the model induced from BOOTSTRAP-LV's selection of training examples is reduced substantially faster than when examples are sampled randomly.

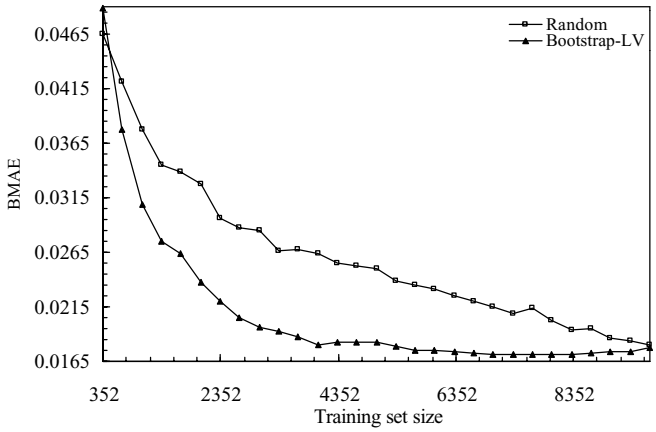


*Figure 5*. CPE learning curves for the Pendigits data set. BOOTSTRAP-LV accelerates error reduction considerably in the initial sampling phases.

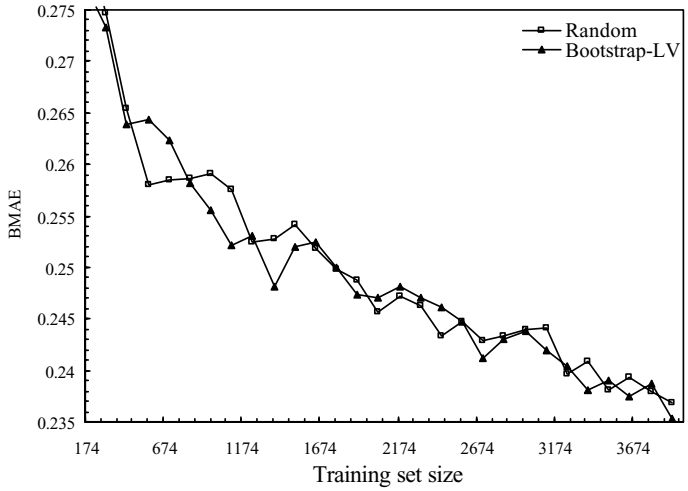


*Figure 6*. CPE learning curves for the weather data set where Bootstrap-lv does not provide improvement in CPE.

For 5 of the 20 data sets, BOOTSTRAP-LV did not succeed in accelerating learning much or at all, as is shown for the Weather data set in figure 6. Note that the accuracy was comparable to that obtained with random sampling: neither curve consistently resides above the other. This is discussed further below.

Table 1 presents a summary of our results for all the data sets. The second column shows the percentage of phases with savings. The third and fourth columns show the top-20% relative saving and the top-20% saving (respectively). The fifth and sixth columns of Table 1 show the average relative saving and the average saving across all sampling phases by applying BOOTSTRAP-LV. The seventh column presents the top-20% error reduction.

*Table 1*. Improvement in examples needed and improvement in error using BOOTSTRAP-LV versus RANDOM.

| Data set | Phases with savings (%) | Top-20% relative saving (%) | Top-20% saving (#) | Avg. relative saving (%) | Avg. saving (#) | Top-20% relative error reduction (%) |
|---|---|---|---|---|---|---|
| **Abalone** | **92.5** | **76.9** | **1152** | **34.9** | **574** | **10.1** |
| **Adult** | **96** | **30.2** | **585** | **17.8** | **302** | **6.6** |
| **Breast cancer-w** | **100** | **51.6** | **110** | **23.8** | **44** | **9.3** |
| **Car** | **89.6** | **35.4** | **281** | **23.3** | **155** | **31.3** |
| **Coding1** | **80** | **47.1** | **475** | **16.2** | **228** | **2.5** |
| **Connect-4** | **100** | **75.4** | **1939** | **45.5** | **984** | **9.5** |
| **Contraceptive** | **93.7** | **42.3** | **129** | **18.4** | **55** | **5.7** |
| German* | 57.1 | 46.5 | 113 | 5.8 | 7 | 5.9 |
| **Hypothyroid** | **100** | **69.0** | **1233** | **64.6** | **705** | **41.1** |
| **kr-vs-kp** | **100** | **27.1** | **57** | **18.1** | **37** | **25.5** |
| **Letter-a**** | **72.4** | **24.8** | **529** | **14.5** | **229** | **10.4** |
| Letter-vowel† | 50 | 12.8 | 429 | 2.1 | 121 | 3.4 |
| Move1 | 65 | 68.4 | 75 | 17.2 | 23 | 3.9 |
| **ocr1** | **93.7** | **42.9** | **168** | **24.5** | **83** | **21.7** |
| **Optdigits** | **94.4** | **50.0** | **762** | **24.5** | **412** | **32.6** |
| **Pendigits** | **100** | **68.6** | **5352** | **61.0** | **3773** | **29.9** |
| **Sick-euthyroid** | **93.1** | **70.2** | **924** | **45.2** | **600** | **26.2** |
| Solar-flare | 64.2 | 41.5 | 58 | 13.5 | 25 | 6.3 |
| Weather | 41.6 | 35.9 | 438 | −10.4 | −46 | 1.7 |
| **Yeast** | **75** | **58.7** | **159** | **23.6** | **79** | **4.9** |

*German credit database.
**letter-recognition, letter a.
†letter-recognition, vowels.

In summarizing these results, to be conservative we regard the two methods to be comparable if the percent of phases with saving is $50\% \pm 15\%$. Thus our first condition for BOOTSTRAP-LV to be deemed superior is that it exhibits superior performance in at least 65% of the phases examined. In addition, in order for BOOTSTRAP-LV to be superior we require that the average relative saving be at least 5% or higher (and symmetrically for RANDOM to be superior the average percentage gain must be –5% or lower). As can be seen in Table 1 (in bold), in 15 out of the 20 data sets BOOTSTRAP-LV exhibited superior performance. Particularly, in all but one of these data sets the percentage of phases with savings is 75% or above. In 13 of those the top-20% relative saving was 30% or more, and in 9 data sets BOOTSTRAP-LV used 50% or less of the number of examples needed by RANDOM to achieve the same accuracy level. For the Sick-euthyroid data set, for example, BOOTSTRAP-LV gradually improves until it is saving more than 70% of the examples (i.e., needing fewer than 30% of the examples required by RANDOM to obtain the same level of accuracy). Since the latter results pertain to the average improvement obtained for the top-20% phases, the maximal savings are even greater.

The measures pertaining to the number of examples saved and the error reduction complement each other and can provide interesting insight. For instance, the number of examples saved can help evaluate the "difficulty" of error reduction, as reflected by the number of examples required by RANDOM to obtain such reduction. For example, although the top-20% relative error reduction for Connect-4 is less than 10%, Table 1 shows that RANDOM needs 984 additional examples on average to obtain the same improvement.

For a single data set (Weather) BOOTSTRAP-LV exhibited a negative average saving. However, the percentage of phases with savings, showing that BOOTSTRAP-LV uses fewer examples in 41% of phases examined, and figure 6, both indicate that the two methods indeed exhibit comparable learning curves for this data set.

An examination of the learning curves for the data sets in which BOOTSTRAP-LV exhibits insignificant or no improvement reveals that training examples chosen at random seem to contribute to error reduction at an almost constant rate. As shown for the Weather data set in figure 6 and for the data sets in figure 7, the learning curves for all these data sets but one (letter-vowel) have an atypical shape, where additional examples bring an almost constant reduction in error, rather than the expected decreasing marginal error reduction. This may indicate that training examples are equally informative regardless of what or how many examples have been already used for training. An intelligent selection of training examples,

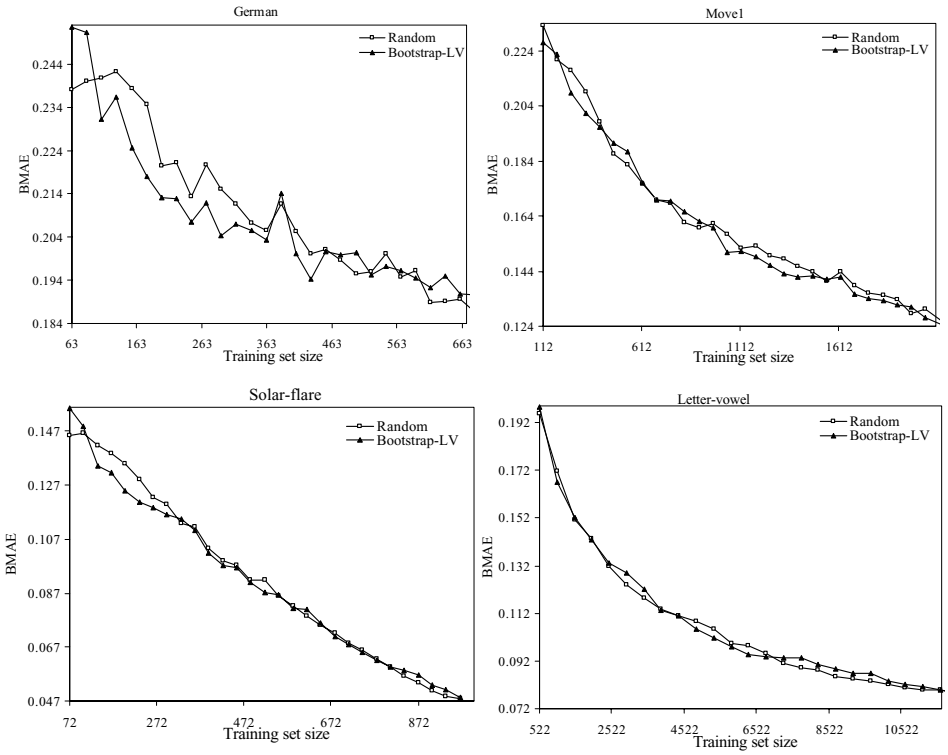


*Figure 7.* Learning curves for data sets where BOOTSTRAP-LV and RANDOM show comparable performance.

therefore, is not likely to improve learning, and will produce results comparable to those obtained with random selection.

### *4.3. Experiments with bagged-PETs*

In order to verify that BOOTSTRAP-LV is effective not solely with PETs, we also experimented with a different CPE learner. Bagged-PETs creates an ensemble of bagged (Brieman, 1996) trees, where each tree is induced from a different bootstrap (Efron & Tibshirani, 1993) sample. The trees are used to estimate the class probability of an instance by averaging the CPEs of the individual PETs in the ensemble. Bagged-PETs are substantially more complex than simple PETs, but have been shown generally to produce superior CPEs compared to simple PETs (Bauer & Kohavi, 1999; Provost, Fawcett, & Kohavi, 1998; Provost & Domingos, 2000; Perlich, Provost, & Simonoff, 2003).

BOOTSTRAP-LV's performance for bagged-PETs concurs with the results obtained for individual PETs. Particularly, for 15 of the data sets BOOTSTRAP-LV exhibited a percentage of phases with savings of more than 65% (in 13 of those the percentage of phases with savings is more than 75%). The top-20% relative saving was 25% or greater in 11 of those data sets. Only in two data sets is the percentage of phases with savings less than 40%.

Figure 8 shows a comparison between BOOTSTRAP-LV and RANDOM for simple PETs and for bagged-PETs. The overall error exhibited by the bagged-PETs is lower than for the simple PETs, and for both models BOOTSTRAP-LV achieves its lowest error with considerably fewer examples than are required for RANDOM.

### *4.4. Other evaluation criteria*

We also evaluated BOOTSTRAP-LV using alternative performance measures: the mean squared error measure used by Bauer and Kohavi (1999), as well as the area under the

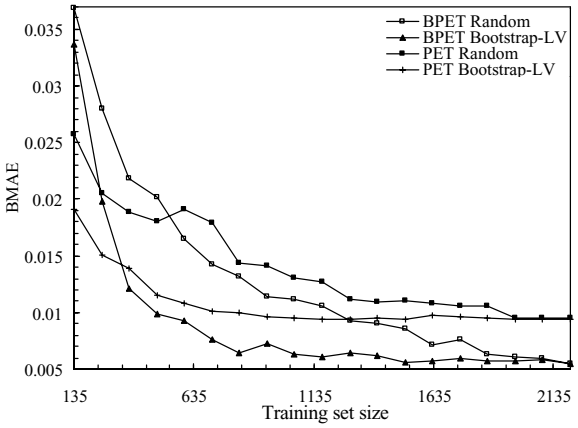


*Figure 8.* CPE learning curves for the Hypothyroid data set showing the performance of BOOTSTRAP-LV and RANDOM with bagged PETs and with simple PETs.

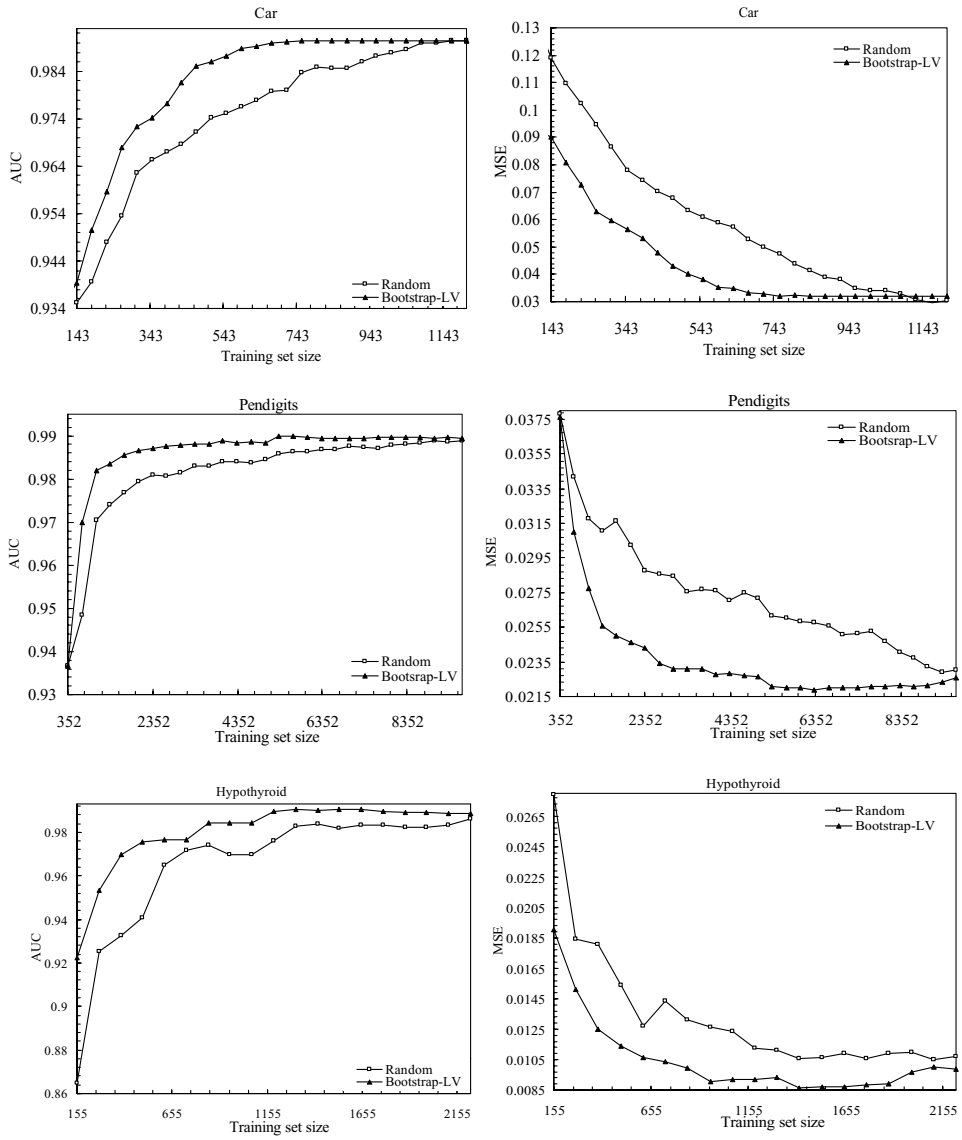


*Figure 9*. Learning curves for area under the ROC curve and MSE, comparing BOOTSTRAP-LV and RANDOM.

ROC curve (denoted AUC) (Bradley, 1997), which specifically evaluates ranking accuracy. The results for these measures agree with those obtained with BMAE. For example, BOOTSTRAP-LV generally leads to fatter ROC curves with fewer examples. Figure 9 presents learning curves of both measures for the Car, Pendigits and Hypothyroid data sets, whose learning curves using BMAE were presented earlier.

### *4.5. Comparisons with uncertainty sampling*

We now compare BOOTSTRAP-LV with an active learning algorithm previously shown to improve classification accuracy; improved classification accuracy may also result from improved class probability estimation error. The comparison shows that focusing on improving CPEs indeed adds value, and also provides interesting insight into the properties of the algorithms.

For the comparison we selected the well-known UNCERTAINTY SAMPLING algorithm (Lewis & Gale, 1994), proposed for the active learning of binary classifiers. Our choice was based on the generality of the algorithm, allowing it to be applied with an arbitrary modeling scheme (that produces CPEs) and an arbitrary data set. In addition, UNCERTAINTY SAMPLING focuses on identifying training examples and does not change the classifier architecture. In contrast, some active learning algorithms result in an ensemble of classifiers (Abe & Mamitsuka, 1998; Iyegar, Apte, & Zhang, 2000). Comparing these to active learning for single classifiers (with active or random sampling) confounds the effects of active learning and producing ensembles.[4] UNCERTAINTY SAMPLING allows us to compare the selection mechanism of the two algorithms over a wide range of domains. We present a summary of the comparison results in Table 2, where all the measures are the same as in Table 1, except that the baseline comparison is UNCERTAINTY SAMPLING rather than RANDOM.

BOOTSTRAP-LV exhibits markedly superior performance compared to UNCERTAINTY SAMPLING. Particularly, BOOTSTRAP-LV is superior for 13 of the data sets (bold), and for 6 data sets the methods exhibit comparable performance, where savings were exhibited in 50% to 60% of the phases. UNCERTAINTY SAMPLING exhibits superior performance for one data set, Solar-flare, for which it produces better probability estimations (in the prior comparison for this data set BOOTSTRAP-LV was not considerably better than RANDOM).

Several factors contribute to the weak performance of UNCERTAINTY SAMPLING for CPE compared to BOOTSTRAP-LV. To understand them, recall the differences between UNCERTAINTY SAMPLING and BOOTSTRAP-LV: the effectiveness score each algorithm assigns to potential training examples and the mechanisms they employ to sample/select examples for labeling. Consider the latter first. Because it uses direct selection, UNCERTAINTY SAMPLING does not account for the potential relevance of a training example for improving the estimation of other examples in the space. It therefore is susceptible to selecting examples with little contribution to the average error across the example space. This may degrade its performance, particularly compared to random sampling. Second, its effectiveness score causes UNCERTAINTY SAMPLING to prefer examples whose CPE is close to 0.5. Thus examples whose true class probability is close to 0.5 and that are captured *correctly* by the model (hence their CPE is close to 0.5 as well) are more likely to be selected; yet they provide little or no new information for learning. Similarly, UNCERTAINTY SAMPLING is less likely to select examples whose CPEs are close to either 1 or 0, even when these estimations are erroneous. Note that because UNCERTAINTY SAMPLING was designed for classification, this is reasonable. A CPE that is on the "correct" side of the decision boundary is sufficient to make a correct classification, even though it may exhibit a large estimation error. Hence, this policy is likely to be productive for selecting examples to improve classification accuracy, but will deny information important for the learner to improve the model's CPEs.

*Table 2.* Improvement in number of training examples required to achieve a certain accuracy level and improvement in error for a given number of training examples using BOOTSTRAP-LV versus UNCERTAINTY SAMPLING.

| Data set | Phases with savings (%) | Top-20% relative saving (%) | Top-20% saving (#) | Avg. relative saving (%) | Avg. saving (#) | Top-20% relative error reduction (%) |
|---|---|---|---|---|---|---|
| Abalone | 50.00 | 61.09 | 801 | 17.63 | 102 | 14.11 |
| Adult | **69.23** | **35.03** | **284** | **9.56** | **69** | **11.13** |
| Breast cancer-w | 55.56 | 49.37 | 144 | 10.90 | 15 | 20.20 |
| Car | 62.50 | 50.46 | 68 | 9.95 | 6 | 36.30 |
| **Coding1** | **93.75** | **63.25** | **1027** | **31.77** | **686** | **6.74** |
| **Connect-4** | **89.47** | **85.52** | **3230** | **43.89** | **1958** | **54.02** |
| Contraceptive | 50.00 | 54.87 | 126 | 11.76 | 21 | 10.01 |
| **German** | **81.25** | **48.14** | **146** | **24.74** | **69** | **8.12** |
| **Hypothyroid** | **71.43** | **62.30** | **307** | **17.10** | **85** | **62.72** |
| **kr-vs-kp** | **94.74** | **57.71** | **144** | **33.90** | **90** | **60.43** |
| **Letter-a** | **85.00** | **44.34** | **771** | **15.50** | **395** | **21.29** |
| **Letter-vowel** | **100.00** | **81.27** | **14210** | **63.80** | **11463** | **44.97** |
| **Move1** | **100.00** | **62.89** | **247** | **39.96** | **194** | **16.29** |
| **ocr1** | **100.00** | **51.90** | **256** | **35.86** | **146** | **34.30** |
| **Optdigits** | **100.00** | **44.13** | **1359** | **26.08** | **570** | **34.91** |
| **Pendigits** | **95.00** | **60.85** | **1636** | **27.45** | **996** | **38.30** |
| **Sick-euthyroid** | **100.00** | **84.12** | **1692** | **59.13** | **1093** | **40.51** |
| Solar-flare | 0.00 | −2.98 | −17 | −16.66 | −69 | −1.64 |
| Weather | 56.25 | 35.06 | 351 | 6.32 | 3 | 1.98 |
| Yeast | 53.33 | 40.38 | 121 | 7.74 | 3 | 6.03 |

Note that when CPEs are extreme but on the "correct side" of the decision boundary, an effort to select examples to improve CPE may undermine an improvement in classification accuracy. This may be inferred from Friedman's analysis of classification error (Freidman, 1997). In particular, binary classification error is minimized if the class most likely to occur is predicted. The probability that due to erroneous CPE the predicted class $\hat{y}$ is not the most likely class, denoted $y_L$, is given by

$$P(\hat{y} \neq y_L) = I(f < 1/2) \int_{1/2}^{\infty} p(\hat{f})\, d\hat{f} + I(f \geq 1/2) \int_{-\infty}^{1/2} p(\hat{f})\, d\hat{f}$$

where $I(\mathrm{A}) = 1$ , if A is true, and $I(\mathrm{A}) = 0$ otherwise. Assuming that $p(\hat{f})$ is approximated with a standard normal distribution. This probability then is given by:

$$P(\hat{y} \neq y_L) = \Phi\left[\mathrm{sign}(f - 1/2)\frac{E\hat{f} - 1/2}{\sqrt{\mathrm{var}\,\hat{f}}}\right]$$

where $\Phi$ is the upper tail area of the standard normal distribution, and $E$ denotes a statistical expectation.[5] Given a certain estimation variance, when the true class probability $f$ and the expected probability estimation, $E\hat{f}$, are on the same "side" of the decision boundary, the farther $E\hat{f}$ is from 0.5, the more the probability of a classification error is reduced, because it is less probable for the estimated class probability to be on the "wrong" side of the decision boundary.

Therefore, for an active learning algorithm aiming to improve classification accuracy, it may not always be beneficial to improve CPEs. For instance, consider a true class probability of 0.6 and a mean estimation of 0.8. An attempt to alter the procedure to reduce the mean estimation to 0.6 increases the likelihood of an estimation that is below 0.5, particularly when the estimation variance is large, thus increasing the likelihood of a classification error.

### *4.6. The effect of weight sampling*

We argued earlier for using weight sampling to reduce generalization error. Particularly, we argued for its ability to account for an example's potential for reducing the error of other examples in the example space. Figure 10 shows for the Pendigits data set the error obtained with weight sampling (viz., using BOOTSTRAP-LV), BOOTSTRAP-LV using direct selection instead (with the same effectiveness score), and random sampling. For readability we present the first 10 samples. As can be seen in figure 10, in the initial and most critical sampling phases for active learning, weight sampling results in lower error compared to direct selection and to random sampling. This phenomenon can be seen for most of our data sets.

The superiority of BOOTSTRAP-LV over random sampling demonstrates that the weights assigned to examples in BOOTSTRAP-LV, and which underlie the sampling process, provide

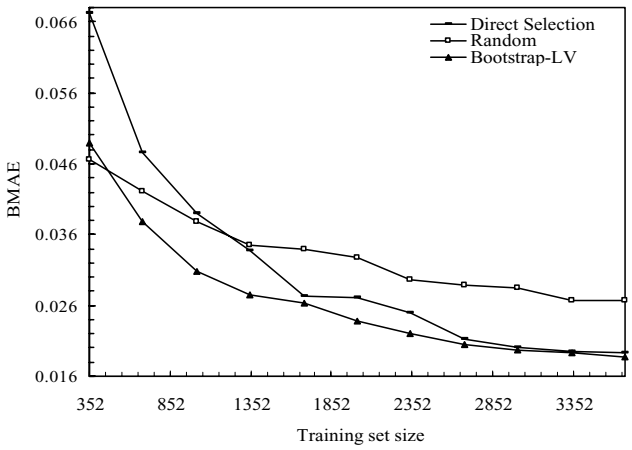


*Figure 10.* Learning curves for weight-sampling, direct selection with BOOTSTRAP-LV's effectiveness score, and RANDOM.

useful information for selecting more informative training examples. The models induced when these weights are ignored and examples are sampled at random (i.e., all weights are equal) are inferior to those induced when the assigned weights are incorporated to direct the sampling process.

Additionally, weight sampling also is important because direct selection often provides results inferior to BOOTSTRAP-LV. As discussed in Section 2, we argue that weight sampling is important in order to select examples more likely to affect other examples in the space, and to avoid selecting training examples that, although not well captured by the model (and hence their estimation may be improved), will not provide information about (many) other examples in the space—and therefore are not likely to reduce the generalization error significantly. These considerations may result in smaller error reduction for Direct Selection, as observed in the comparison to BOOTSTRAP-LV. Apparently, choosing examples based on the potential of reducing the error of a single example, as some methods do, is not sufficient. It is important to consider the effect of each training example on the general population of examples in the space.

***4.6.1. Improving uncertainty sampling for CPE.*** To demonstrate the effect weight sampling has on identifying informative examples, we propose an improvement to UNCERTAINTY SAMPLING by incorporating weights which reflect the UNCERTAINTY SAMPLING rationale (for its effectiveness score) and then to weight sample examples according to their weights. We will show how the performance of this algorithm improves CPE and compare it to BOOTSTRAP-LV.

Since UNCERTAINTY SAMPLING selects examples whose CPE is close to 0.5, we assign to each example a weight that reflects this distance. In particular, at each sampling phase $s$, the weight assigned to example $x_i$ is given by $W_s(x_i) = \frac{(0.5-|0.5-p_i|)}{R}$, where $R$ is a normalization factor such that $W$ is a distribution. The probability of an example being sampled increases the closer its CPE is to 0.5. The algorithm denoted WEIGHTED UNCERTAINTY SAMPLING (WUS), is described in figure 11 below.

Comparing the new WUS algorithm with BOOTSTRAP-LV for CPE we see that WUS is much more competitive with BOOTSTRAP-LV than UNCERTAINTY SAMPLING is. A summary of the results is presented in Table 3. BOOTSTRAP-LV outperforms WUS for 8 data sets (in

**Input**: an initial labeled set $L$, an unlabeled set $UL$, an inducer $I$,
a stopping criterion, and an integer $M$ specifying the number of actively selected examples in each phase.
1 While stopping criterion not met
  /* perform next phase: */
2  Apply inducer $I$ to $L$
3  For each example $\{x_i \mid x_i \in UL\}$ assign weight $W_s(x_i)=\frac{(0.5-|0.5-p_i|)}{R}$
4  Sample from the probability distribution $W_s$, a subset S of $M$ examples from $UL$ without replacement
5  Remove S from $UL$, label examples in S, and add them to $L$
6 end for
**Output:** estimator $E$ induced with $I$ from $L$

*Figure 11.* The WEIGHTED UNCERTAINTY SAMPLING algorithm.

*Table 3*. Improvement in examples needed and improvement in error using BOOTSTRAP-LV versus WEIGHTED UNCERTAINTY SAMPLING.

| Data set | Phases with savings (%) | Top-20% relative saving (%) | Top-20% saving (#) | Avg relative saving (%) | Avg saving (#) | Top-20% relative error reduction (%) |
|---|---|---|---|---|---|---|
| Abalone | 57.1 | 46.30 | 577 | 7.97 | 62 | 3.57 |
| Adult | 76 | 14.07 | 414 | 4.99 | 123 | 2.71 |
| Breast cancer-w | 44.44 | 18.75 | 44 | 0.10 | −9 | 6.86 |
| **Car** | **92.85** | **17.62** | **136** | **9.74** | **67** | **14.08** |
| **Coding1** | **87.5** | **28.33** | **671** | **16.55** | **379** | **2.77** |
| Connect-4 | 47.36 | 18.11 | 413 | 2.10 | 27 | 3.07 |
| Contraceptive | 33.33 | 14.15 | 58 | −2.51 | −5 | 3.19 |
| **German** | **68.75** | **43.01** | **133** | **17.24** | **43** | **6.66** |
| **Hypothyroid** | **100** | **81.83** | **1782** | **65.41** | **1260** | **62.08** |
| kr-vs-kp | 31.57 | 3.77 | 5 | −1.34 | −5 | 3.84 |
| *Letter-a* | *30* | *13.30* | *693* | *−7.23* | *−583* | *7.26* |
| Letter-vowel | 46.66 | 10.73 | 765 | 0.32 | −24 | 3.44 |
| **Move1** | **88.88** | **26.73** | **138** | **13.45** | **62** | **8.11** |
| ocr1 | 62.5 | 13.74 | 66 | 2.68 | 16 | 11.20 |
| Optdigits | 64.28 | 20.40 | 721 | 8.32 | 229 | 12.14 |
| **Pendigits** | **90** | **53.40** | **4064** | **36.39** | **2468** | **22.85** |
| **Sick-euthyroid** | **100** | **53.61** | **859** | **41.33** | **537** | **17.54** |
| *Solar-flare* | *0* | *2.46* | *−19* | *−16.79* | *−56* | *−9.11* |
| Weather | 52.63 | 17.80 | 328 | 0.35 | 45 | 1.50 |
| **Yeast** | **73.33** | **41.79** | **189** | **16.59** | **64** | **5.03** |

bold), BOOTSTRAP-LV and WUS are comparable for 10 data sets and WUS is superior in two (italicized). In comparison BOOTSTRAP-LV provides superior CPEs compared to UNCERTAINTY SAMPLING for 14 out of 20 data sets. For six data sets in which UNCERTAINTY SAMPLING is inferior to BOOTSTRAP-LV, WUS exhibits comparable performance to that of BOOTSTRAP-LV. Overall BOOTSTRAP-LV remains superior, yet the new WEIGHTED UNCERTAINTY SAMPLING algorithm exhibits improved performance compared to UNCERTAINTY SAMPLING.

Figure 12 shows CPE learning curves for BOOTSTRAP-LV, UNCERTAINTY SAMPLING and WUS for the Connect-4 data set. Whereas UNCERTAINTY SAMPLING is inferior to BOOTSTRAP-LV for the Connect-4 data set, WUS's performance is comparable to that of BOOTSTRAP-LV. We assert that this can be attributed primarily to WUS accounting for a broader set of considerations when selecting examples, particularly WUS's consideration of the potential error reduction effect an example may have on other examples in the space.

Figure 13 shows learning curves of the three algorithms for the sick-euthyroid data set, where similarly, WUS's performance is considerably better than that of UNCERTAINTY SAMPLING, but the CPE generalization error of BOOTSTRAP-LV still is better than that obtained

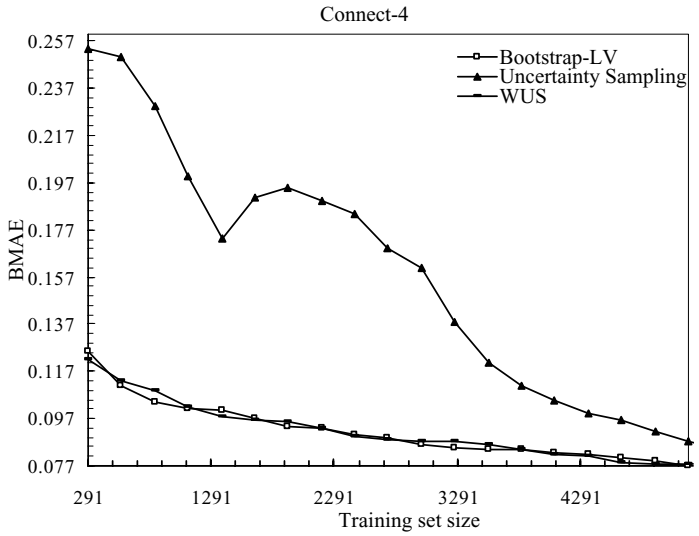


*Figure 12*. An example where WUS is superior to UNCERTAINTY SAMPLING and achieves performance comparable to that of BOOTSTRAP-LV.

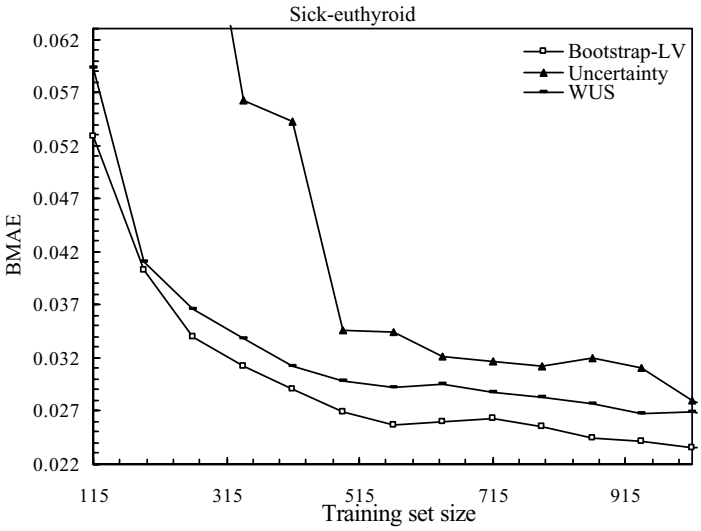


*Figure 13*. BOOTSTRAP-LV remains superior but WUS shows significant improvements compared to UNCERTAINTY SAMPLING.

with WUS. The improved performance of BOOTSTRAP-LV in 8 data sets demonstrates that the weights assigned by BOOTSTRAP-LV better support the sampling mechanism in identifying informative examples to improve CPE. As we discussed in section 4.2, weights assigned to examples in WUS may not always be adequate to improve CPEs. Particularly, the focus

on selecting examples whose CPE is closer to 0.5 and avoiding examples whose CPE is closer to either 0 or 1 sometimes hinders the reduction of CPE generalization error.

The above results suggest that given an informative effectiveness score, weight sampling indeed provides important additional information, improving the selection of informative training examples. Given the performance of the two algorithms, the effectiveness score computed in BOOTSTRAP-LV is superior to the score assigned to examples by WUS, yet the effectiveness score in WUS is informative. As we discussed earlier in the paper, by preferring examples whose CPE is close to 0.5 WUS identifies examples whose class is uncertain; however, such uncertainty apparently also implies some uncertainty regarding CPE and can benefit from gaining more relevant evidence. Yet BOOTSTRAP-LV produces better results because WUS may fail to identify all CPE uncertainties, particularly when these uncertainties do not imply class uncertainty. In addition, as we mentioned above, a CPE that is close to 0.5 does not necessarily imply CPE uncertainty when the true CPE is also close to 0.5 and is correctly estimated by the model.

Our results with WUS further suggest that algorithms for improving classification accuracy can capitalize on weight sampling. For example, WUS may also exhibit improved performance compared to UNCERTAINTY SAMPLING for classification accuracy. Similarly, other effectiveness scores proposed to identify examples to increase classification accuracy, such as entropy, and that do not incorporate additional measures to capture the effect of a training example on other examples in the space are likely to benefit from weight sampling.

## 5. Limitations

The advantages gained by BOOTSTRAP-LV come with computational cost. At each phase of the algorithm $k$ models are induced from bootstrap samples. If $n$ is the number of training examples, the cost of generating each bootstrap sample is $kO(n)$; for an arbitrary modeling scheme whose computational complexity of inducing a model from $n$ examples is $C(n)$, the added complexity from inducing these $k$ models is $kC(n)$. In order to compute the weights for weight sampling, the model is applied to estimate the class probability for all examples in $UL$. Let the average complexity of applying the model for a particular input example be $A$, which depends on the type of model. For each phase, BOOTSTRAP-LV subsequently samples $M$ examples from $UL$, a procedure whose generic complexity is $O(M \log M) + |UL|$, which constitutes the cost of sorting the list of selected random numbers and of scanning $UL$ for the corresponding examples. Therefore the computation cost at each phase is $k[O(n) + C(n)] + (A + 1) \cdot |UL| + O(M \log M)$, where $C(n)$ and $A$ are dependent on the modeling scheme used.

Given that the number of examples sampled in each phase, $M$, is relatively small, the dominant computational components are $C(n)$, the cost of generating a model (which must be done $k$ times), and $A$, the cost of applying a model, which must be done for all of $UL$. As mentioned earlier, for a very large unlabeled set, a sample from $|UL|$ can be used instead. In addition, because of the typical shape of the learning curve, beyond a certain training-set size the marginal error reduction is insignificant, whether active learning or random sampling is employed. Thus, intelligent selection of examples for learning is critical only in the early part of the curve (where $n$ is small). If the $n$ remains relatively small, multiple

model induction from bootstrap samples does not constitute a considerable computational toll.

Moreover, BOOTSTRAP-LV provides an appropriate solution whenever labeling costs are more important than computational costs, such as when the primary concern is to obtain accurate CPE or ranking with minimal costly labeling.

BOOTSTRAP-LV does not address computational concerns explicitly, as do Lewis and Catlett (1994). However, while UNCERTAINTY SAMPLING is simpler computationally, its performance is significantly inferior to that of BOOTSTRAP-LV and in the initial sampling phases is often inferior to random sampling as well. BOOTSTRAP-LV's performance also surpasses the performance of WEIGHTED UNCERTAINTY SAMPLING. Yet, since WEIGHTED UNCERTAINTY SAMPLING also incorporates a CPE uncertainty measure and is computationally simpler it should be considered for active learning of CPEs when computational concerns are particularly critical.

Lastly, BOOTSTRAP-LV relies on detecting variance in CPEs to infer what examples are useful for obtaining more accurate estimation. Its performance may be hampered, therefore, when a low-variance model such as logistic regression is used for learning.

## 6. Conclusions

BOOTSTRAP-LV was designed to use fewer labeled training data to produce accurate class probability estimates. The algorithm addresses two key components of active learning: an effectiveness score and a selection procedure, which complement each other to identify particularly informative examples for learning class probability estimates. BOOTSTRAP-LV is domain independent and is not restricted to a particular learning algorithm.

An empirical evaluation of the approach shows that it performs well, indeed using fewer training data. The evaluation encompasses a wide range of benchmark domains providing evidence for the general efficacy of the algorithm. In particular, the results show how the information provided by the effectiveness scores improves upon random sampling (i.e., when all weights are equal). They also show that BOOTSTRAP-LV outperforms an existing active learning method, UNCERTAINTY SAMPLING. We investigate the properties of the algorithms to explain these results. For example, we demonstrate how both the weights assigned to potential training examples and the weight sampling procedure combine to produce superior CPEs.

Lastly, we use the results of this investigation to propose yet another active learning algorithm, WEIGHTED UNCERTAINTY SAMPLING, which assigns effectiveness scores reflecting the rationale of UNCERTAINTY SAMPLING's effectiveness score, but which in addition, employs the scores to weight sample examples for training (as does BOOTSTRAP-LV). A comparison with BOOTSTRAP-LV reveals that BOOTSTRAP-LV still is superior for improving CPEs, demonstrating the value of BOOTSTRAP-LV's effectiveness score, but also demonstrates the advantages conferred by weight sampling. The improvement over direct selection suggests the application of weight sampling with other effectiveness scores proposed in the literature for the active learning of classifiers.

Making decisions in cost-sensitive environments often takes a decision-theoretic approach to evaluating alternatives, requiring the estimation of probabilities of events or

classes in order to assess alternative decisions. In such environments labeling costs often also must be taken into account. We have shown that active sampling can be effective for reducing the cost of labeling necessary to build accurate models for class-probability estimation and ranking.

## Acknowledgments

We thank Vijay Iyengar for helpful comments. We also thank IBM for a Faculty Partnership Award and The Penn State eBusiness Research Center for its support.

This work is sponsored in part by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-01-2-585. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory, or the U.S. Government.

## Notes

1. Classification accuracy has been criticized previously as a metric for machine learning *research* (Provost et al., 1998).
2. Roy and McCallum note the domain-specific limitation of this approach (Roy & McCallum, 2001).
3. Probability estimation trees are easy to build, fast computationally, robust across data sets, comprehensible to human experts, and produce surprisingly good probability-based rankings (Provost & Domingos, 2000; Perlich, Provost, & Simonoff, 2003).
4. Ensembles usually improve learning curves even with random selection.
5. Note that with respect to an active leaning algorithm the estimation procedure whose variance and expectation appear in the formulation above also incorporates the choice of training examples.

## References

Abe, N., & Mamitsuka, H. (1998). Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 1–9).

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

Argamon-Engelson, S., & Dagan, I. (1999). Committee-based sample selection for probabilistic classifiers. *Journal of Artificial Intelligence Research, 11*, 335–360.

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning, 36*, 105–142.

Blake, C. L., & Merz, C. J. (1998). UCI *Repository of Machine Learning Databases*. Irvine, CA: University of California, Department of Information and Computer Science [http://www.ics.uci.edu/∼mlearn/MLRepository.html].

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30:7*, 1145–1159.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 26:2*, 123–140.

Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning. In *Proceedings of the Ninth European Conference on Artificial Intelligence* (pp. 147–149).

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

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

Cohen, W., & Singer, Y. A. (1999). A simple, fast, and effective rule learner. In *Proceedings of the Sixteenth National Conference of the American Association of Artificial Intelligence* (pp. 335–342).

Efron, B., & Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Chapman and Hall.

Freund, Y., & Schapire, R. (1996). Experiments with a new boosting algorithm. In *Proceedings of the International Conference on Machine Learning* (pp. 148–156).

Friedman, J. H. (1997). On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Journal of Knowledge Discovery and Data Mining*, 55–77.

Iyengar, V. S., Apte, C., & Zhang, T. (2000). Active learning using adaptive resampling. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 92–98).

Lewis, D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 3–12).

Lewis, D., & Catlett, J. (1994). Heterogeneous uncertainty sampling. In *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 148–156).

MaCallum, A., & Nigam, K. (1998). Employing EM in pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 350–358).

Perlich, C., Provost, F., & Simonoff, J. S. (2003). Tree induction vs. logistic regression: A learning-curve analysis. *Journal of Machine Learning Research, 4:June*, 211–255.

Porter, B. W., & Kibler, D. F. (1986). Experimental goal regression: A method for learning problem-solving heuristics. *Machine Learning, 1:3*, 249–285.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing classifiers. In *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 445–453).

Provost, F., & Domingos, P. (2000). Well-trained PETs: Improving probability estimation trees. CeDER Working Paper #IS-00-04, Stern School of Business, NYU.

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufman.

Roy, N., & McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 441–448).

Seung, H. S., Opper, M., & Smopolinsky, H. (1992). Query by committee. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory* (pp. 287–294).

Simon, H. A., & Lea, G. (1974). Problem solving and rule induction: A unified view. In L. W. Gregg (Ed.), *Knowledge and Cognition*. Chap. 5. Potomac, MD: Erlbaum.

Turney, P. D. (2000).Types of cost in inductive concept learning. Workshop on Cost-Sensitive Learning at ICML-2000, Stanford University, California, 15–21.

Winston, P. H. (1975). Learning structural descriptions from examples. In P. H. Winston (Ed.), *The Psychology of Computer Vision*. New York: McGraw-Hill.

Zadrozny B., & Elkan, C. (2001). Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 204–212).

Received Nov. 8, 2001
Revised Sept. 18, 2002
Accepted Sept. 18, 2002
Final manuscript Sept. 18, 2002