

November 7, 2001

## Active Sampling for Class Probability Estimation and Ranking

**Maytal Saar-Tsechansky**

*Department of Information Systems  
Leonard N. Stern School of Business, New York University  
44 West Fourth Street  
New York, NY 10012, USA  
Tel: (212) 998-0812*

MTSECHAN@STERN.NYU.EDU

**Foster Provost**

*Department of Information Systems  
Leonard N. Stern School of Business, New York University  
44 West Fourth Street  
New York, NY 10012, USA  
Tel: (212) 998-0806*

FPROVOST@STERN.NYU.EDU

### 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 for an active learning approach and present an active learning method for estimating class probabilities and ranking. `BOOTSTRAP-LV` identifies particularly informative new data for learning based on the variance in probability estimates, and by accounting for a particular data item'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 class probability estimation accuracy and provide insights on the behavior of the algorithms. Finally, to further our understanding of the contributions made by the elements of `BOOTSTRAP-LV`, we experiment with a new active learning 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 learning algorithms for classification.

**Keyword:** Active learning, class probability estimation, cost-sensitive learning

## 1. Introduction

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

Active learning processes 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 of a fixed number 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 target marketing the estimated probability that a customer will respond to an offer is combined with the estimated profit (Zadrozny and Elkan, 2001) to evaluate various offer propositions. Other applications require ranking of cases, to improve consumer response rate to offer propositions, as well as to add flexibility for user processing<sup>1</sup>. 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. We therefore focus on learning class probability estimation (CPE) models.

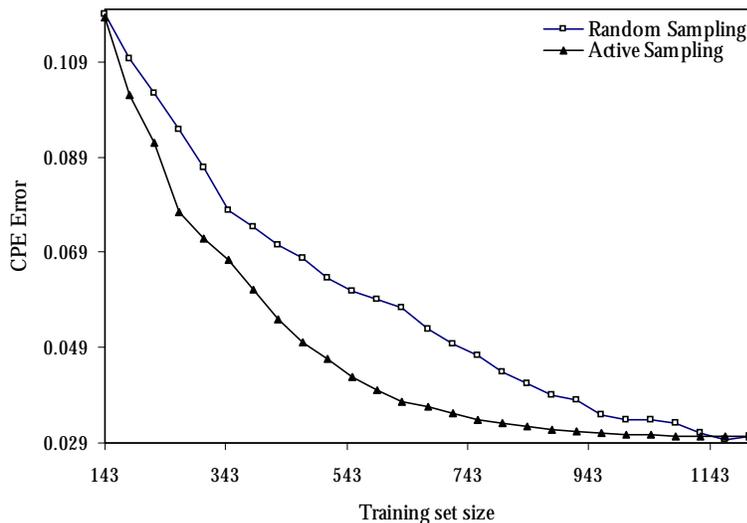


Figure 1: Learning curves for active learning vs. random sampling

<sup>1</sup> Classification accuracy has been criticized previously as a metric for machine learning *research* (Provost et al., 1998).

In this paper we consider active learning to produce accurate CPEs and class-based rankings from fewer learning 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 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 randomly selecting training data; the lower curve represents active learning. The two curves form a "banana" shape: very early on, the curves are comparable because a model is not yet available for active learning. The active learning curve soon accelerates, because of the careful choice of training data. Given enough data, random selection catches up.

We introduce a new active learning technique, BOOTSTRAP-LV, for learning CPEs. BOOTSTRAP-LV uses bootstrap samples (Efron and Tibshirani, 1993) of available training data to examine the variance in the probability estimates for not-yet-labeled data, and employs a weight sampling procedure to finally select particularly informative example for learning. We show empirically across a wide 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. A careful analysis of the algorithm's characteristics and performance reveals the contributions of its components. The analysis also leads to a new algorithm for active learning that is more competitive compared to other methods with respect to BOOTSTRAP-LV and has computational advantages. The latter 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 learning 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. Winston 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 can be 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  $UL$ , until some predefined condition is met (e.g., the labeling budget is exhausted). 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.

---

**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  $\{x_i | x_i \in UL\}$  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

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 learning 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. The example that is expected to reduce the generalization error the most should be added to the training set.

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 (Roy and McCallum, 2001)). We therefore resort to an indirect estimation of potential learning examples' informative value. Also, we consider the potential of each learning example to help improve the estimation of *other* examples in the space not only the performance of its own class probability, 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 its sources. A model's estimation  $\hat{f}(x|T)$  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  $\hat{f}(x|T)$  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.,  $|f(x) - \hat{f}(x|T)|$ . We may infer from the discrepancy whether additional information is needed to improve the model’s estimation. Note however, that in an inductive learning setting the class *probability* for an input,  $f(x)$ , is not known to us, even when an instance’s true class is known.

A common formulation of the estimation error (Friedman, 1997) decomposes the expected squared estimation error into the sum of two terms:  $E_T[(f(x) - \hat{f}(x|T))^2] = E_T[\hat{f}(x|T) - E_T\hat{f}(x|T)]^2 + [f(x) - E_T\hat{f}(x|T)]^2$ ,  $E(\cdot)$  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). Both the estimation error and the estimation bias refer to the actual probability function,  $f(x)$ , which as mentioned 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.

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 estimation variance of each potential learning example (the LV). If the LV is high, we infer that this example 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., generalization error) the most. We first describe the estimation of the estimation 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 and Tibshirani, 1993)  $B_j$ ,  $j = 1, \dots, k$  from  $L$ . We generate a set of models by applying the inducer  $I$  to each bootstrap sample  $B_j$ , resulting in  $k$  estimators  $E_j$ ,  $j = 1, \dots, k$ . 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.

**Algorithm** BOOTSTRAP-LV

---

**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  $B_j$ ,  $j=1,\dots,k$  from  $L$
- 4     Apply inducer  $I$  on each subsample  $B_j$  and induce estimator  $E_j$
- 5     For all examples  $\{x_i | x_i \in UL\}$  compute  $D_s(x_i) = \frac{\sum_{j=1}^k (p_j(x_i) - \bar{p}_i)^2}{R \bar{p}_{i,\min}}$
- 6     Sample from the probability distribution  $D_s$ , a subset  $S$  of  $M$  examples from  $UL$  without replacement
- 7     Remove  $S$  from  $UL$ , label examples in  $S$ , and add them to  $L$
- 8 end for

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

---

Figure 3: The BOOTSTRAP-LV Algorithm

The local variance provides an indication of the potential error reduction for each individual example in the example space if more relevant examples are provided to the learner. It does not however provide an indication of how much is learned from each learning example about other examples in the space. We stated earlier that our objective is to reduce the generalization error by training a model with fewer, particularly informative examples, and we emphasized the potential impact a learning example may have on reducing the estimation error of other examples in the example space. Consider an active learning algorithm where the effectiveness score reflects the potential contribution of a learning example to reducing its own error. Also assume examples are selected in order of their effectiveness score, such that the examples with the highest scores are selected for labeling first. We 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 an example to the training set. The latter, 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 et al., 1994, Lewis and Gale, 1994)). Nevertheless, random sampling is powerful because it allows the incorporation of information about the distribution of examples even when the 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 shares some common features with these examples. Consider a set of examples sharing a set of features. 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 associated with each other.

In order to reduce the average error across the example space  $\text{BOOTSTRAP-LV}$  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 local variance.

Specifically, the distribution from which examples are sampled is given by  $D_s(x_i) = \frac{\sum_{j=1}^k (p_j(x_i) - \bar{p}_i)^2}{R \sqrt{\bar{p}_{i,\min}}}$ , 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  $R = \sum_{i=1}^{\text{size}(UL)} \sum_{j=1}^k (p_j(x_i) - \bar{p}_i)^2 \sqrt{\bar{p}_{i,\min}}$ , 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 in the respected region. 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 et al. (Cohn *et al.*, 1996) propose an active learning approach for statistical learning models, generating queries (i.e., learning 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 learning example to the training set can be induced in order to estimate the expected changes in generalization error. Roy and McCallum (Roy and McCallum, 2001) propose this approach for building classifiers. At each phase they update the current model with each additional learning 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 allows efficient incremental updates of the model, such as the Naïve Bayes algorithm used to classify text documents in the paper (Roy and McCallum, 2001).

When an efficient closed-form computation of the error or incremental model updates are not possible, various active learning approaches compute alternative effectiveness scores. In particular, the QUERY BY COMMITTEE (QBC) algorithm (Seung *et al.*, 1992) was proposed to select learning examples actively for training a binary classifier. Examples are sampled at random, generating a “stream” of potential learning 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 (McCallum and Nigam, 1998) note that a disadvantage of the “stream-based” QBC approach lies in the decision of whether to label an example being “made on each document (i.e., example) individually, irrespective of the alternatives”. Alternatively, estimation uncertainty of all the unlabeled learning examples can be compared, 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 to 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 Mamimtsuka (Abe and Mamimtsuka, 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 and 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 that we return to below.

All these methods, as indicated by their effectiveness scores, are designed to increase classification accuracy, not to improve CPEs or rankings, which is our concern in this paper. In addition, differently from the approach we propose in this paper

these methods do not incorporate the effect of a learning example on other examples across the example space. Particularly, they disregard the potential effect of a training example on reducing the error of other examples in the example space. 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 Dagan and Engelson (Dagan and Engelson, 1999) as well as by McCallum and Nigam (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 may be for an arbitrary domain<sup>2</sup>.

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 and Shapire, 1996) on which Iyengar et al. (Iyengar et al., 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. However, 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 learning 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 and Nigam, 1998), the weight-sampling mechanism can be applied seamlessly for arbitrary domains.

In sum, `BOOTSTRAP-LV` employs an effectiveness score which identifies examples whose CPE varies highly. 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

---

<sup>2</sup> Roy and McCallum note the domain-specific limitation of this approach [Roy and McCallum, 2001].

examples for learning. We argue that weight sampling can help account for the informative value an example confers for 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 with simple probability estimation trees, and with 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 for classifiers, to provide insights on the operation of the algorithm and its advantage compared to existing approaches. Finally, experiments with a new active learning algorithm inspired by the empirical investigation provide further insights 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 *et al.*, 1998) and 3 used previously to evaluate rule-learning algorithms (Cohen and Singer, 1999). Data sets with more than two classes were mapped into two-class problems.

For these experiments we use tree induction to produce class probability estimates<sup>2</sup>. 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 and Kohavi, 1999; Provost *et al.*, 1998; Provost & Domingos, 2000; Perlich *et al.* 2001).

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, in these data sets only class membership is observed and the true class probabilities are unknown. As models are learned from more data, performance improves typically as a learning curve; `BOOTSTRAP-LV` aims to obtain comparable performance with fewer labeled data (recall figure 1). 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 learning 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 and Kohavi, 1999; Provost *et al.*, 1998; Provost & Domingos, 2000). We then calculate the mean abso-

---

<sup>2</sup> Probability estimation trees are easy to build, fast computationally, robust across data sets, comprehensible to human experts, and produces surprisingly good probability-based rankings [Perlich *et al.*, 2001]

lute 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 by `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 to 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; at each phase the same number of examples was added to  $L$ . Results are averaged over 10 random, three-way partitions of the data sets into an initial labeled set, an unlabeled set, and a test set against which the two 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. As shown in Figure 4, 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 learning examples. Note that for visibility the algorithms' performance with the initial labeled set (for which all algorithms perform identically) is 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 also across the entire learning 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 pertaining to the gain 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 gained 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 gain across all sampling phases, referred to as average gain, as well as the gain as a percentage of the number of examples needed by `RANDOM` (i.e., the percentage of examples gained if `BOOTSTRAP-LV` is used instead of `RANDOM`), referred to as average percentage gain. For instance, in the *Car* domain (Figure 4) the average gain is 155 examples and the average percentage gain 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. Therefore an average across all phases merely provides an indication of whether `BOOTSTRAP-LV` produces superior estimations. Therefore 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 we provide rather than the single best gain exhibited by `BOOTSTRAP-LV` in a certain sampling phase, the average gain of the top 20% sampling phases in which the largest gains were obtained. We call the latter the “top gain”. We also compute the top gain as a percentage of the examples needed by `RANDOM`, referred to as percentage top gain. For instance, in the Car domain the top gain is 281 examples or 35.4% of the examples needed by `RANDOM`. Lastly, we calculate the percentage of sampling phases in which `RANDOM` needs more examples to obtain the error level exhibited by `BOOTSTRAP-LV`. We refer to the latter as the percentage of phases with positive gain.

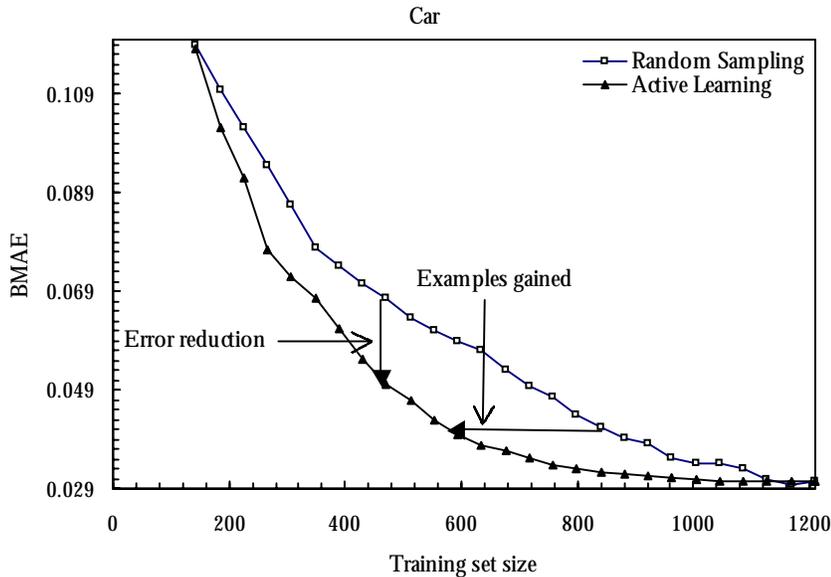


Figure 4: Learning behavior of `BOOTSTRAP-LV` and `RANDOM` for the Car data set

Finally, for each data set we also calculated the error reduction achieved by `BOOTSTRAP-LV` with respect to `RANDOM` for the same number of training examples. This dimension also is demonstrated in Figure 4. As above 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 error reduction. For the Car domain the top 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. As can be seen in Figure 4, in the initial phases the error exhibited by the model induced from `BOOTSTRAP-LV`’s selection of learning examples is reduced substantially faster than when examples are sampled randomly.

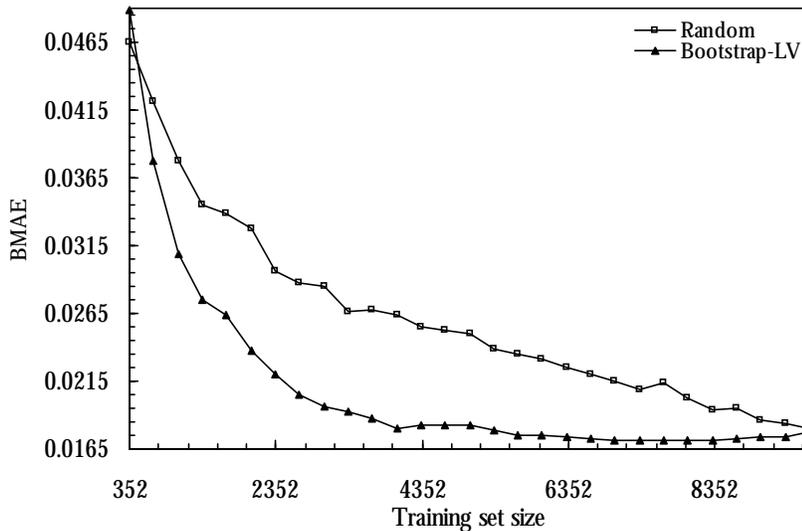


Figure 5: CPE learning curves for the Pendigits data set. `BOOTSTRAP-LV` accelerates error reduction considerable in the initial sampling phases.

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, as 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 positive gain. The third and fourth columns show the top percentage gain and the top gain (respectively). The fifth and sixth columns of Table 1 show the average percentage gain and the average gain across all sampling phases by applying `BOOTSTRAP-LV`. The seventh column presents the top error reduction.

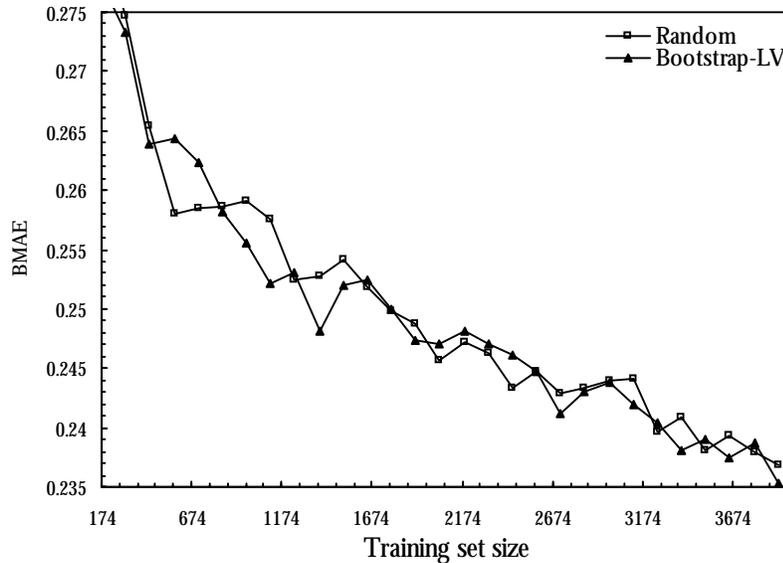


Figure 6: CPE learning curves for the Weather data set where Bootstrap-lv does not provide improvement in CPE

In summarizing these results, to be conservative we regard the two methods to be comparable if the percent of phases with positive gain is between  $50\% \pm 15\%$ . 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 examples gained by `BOOTSTRAP-LV` be at least 5% or higher (and symmetrically for `RANDOM` to be superior the average examples gained 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<sup>3</sup>. Particularly, in all but one data set the percentage of phases with positive gain is 75% or above. In 13 of those the top percentage gain was 30% or more, and in 9 data sets `BOOTSTRAP-LV` used less than 50% of the number of examples required for `RANDOM` to achieve comparable accuracy. For the Sick-euthyroid data set, for example, `BOOTSTRAP-LV` gradually improves until it requires 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 gain is higher.

<sup>3</sup> Appendix A shows the actual leaning curves for all data sets; the reader can verify that the conditions we apply are reasonable.

Table 1: Improvement in examples needed and improvement in error using `BOOTSTRAP-LV` versus `RANDOM`

Data set	Phases with positive gain (%)	Top gain (%)	Top gain (#)	Avg gain (%)	Avg gain (#)	Top error reduction (%)
<b>abalone</b>	<b>92.5</b>	<b>76.9</b>	<b>1152</b>	<b>34.9</b>	<b>574</b>	<b>10.1</b>
<b>adult</b>	<b>96</b>	<b>30.2</b>	<b>585</b>	<b>17.8</b>	<b>302</b>	<b>6.6</b>
<b>breast cancer-w</b>	<b>100</b>	<b>51.6</b>	<b>110</b>	<b>23.8</b>	<b>44</b>	<b>9.3</b>
<b>car</b>	<b>89.6</b>	<b>35.4</b>	<b>281</b>	<b>23.3</b>	<b>155</b>	<b>31.3</b>
<b>coding1</b>	<b>80</b>	<b>47.1</b>	<b>475</b>	<b>16.2</b>	<b>228</b>	<b>2.5</b>
<b>connect-4</b>	<b>100</b>	<b>75.4</b>	<b>1939</b>	<b>45.5</b>	<b>984</b>	<b>9.5</b>
<b>contraceptive</b>	<b>93.7</b>	<b>42.3</b>	<b>129</b>	<b>18.4</b>	<b>55</b>	<b>5.7</b>
german*	57.1	46.5	113	5.8	7	5.9
<b>hypothyroid</b>	<b>100</b>	<b>69.0</b>	<b>1233</b>	<b>64.6</b>	<b>705</b>	<b>41.1</b>
<b>kr-vs-kp</b>	<b>100</b>	<b>27.1</b>	<b>57</b>	<b>18.1</b>	<b>37</b>	<b>25.5</b>
<b>letter-a**</b>	<b>72.4</b>	<b>24.8</b>	<b>529</b>	<b>14.5</b>	<b>229</b>	<b>10.4</b>
letter-vowel <sup>+</sup>	50	12.8	429	2.1	121	3.4
move1	65	68.4	75	17.2	23	3.9
<b>ocr1</b>	<b>93.7</b>	<b>42.9</b>	<b>168</b>	<b>24.5</b>	<b>83</b>	<b>21.7</b>
<b>optdigits</b>	<b>94.4</b>	<b>50.0</b>	<b>762</b>	<b>24.5</b>	<b>412</b>	<b>32.6</b>
<b>pendigits</b>	<b>100</b>	<b>68.6</b>	<b>5352</b>	<b>61.0</b>	<b>3773</b>	<b>29.9</b>
<b>sick-euthyroid</b>	<b>93.1</b>	<b>70.2</b>	<b>924</b>	<b>45.2</b>	<b>600</b>	<b>26.2</b>
solar-flare	64.2	41.5	58	13.5	25	6.3
weather	41.6	35.9	438	-10.4	-46	1.7
<b>yeast</b>	<b>75</b>	<b>58.7</b>	<b>159</b>	<b>23.6</b>	<b>79</b>	<b>4.9</b>

\* German credit database

\*\* letter-recognition, letter a

+ letter-recognition, vowels

The measures pertaining to the number of examples gained and the error gain complement each other and can provide interesting insight. For instance, the number of examples gained 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 error reduction for `Connect-4` is less than 10%, Table 1 shows that it requires `RANDOM` 984 additional examples on average to obtain the same improvement.

For a single data set (`Weather`) `BOOTSTRAP-LV` exhibited a negative average gain. However, the percentage of phases with positive gain, showing that `BOOTSTRAP-LV` uses fewer examples in 41% of phases examined, and Figure 5, 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 at all 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 learning examples are equally informative regardless of what or how many examples have been already used for training. An intelligent selection of learning examples, therefore, is not likely to improve learning, and will produce results comparable to those obtained with random selection.

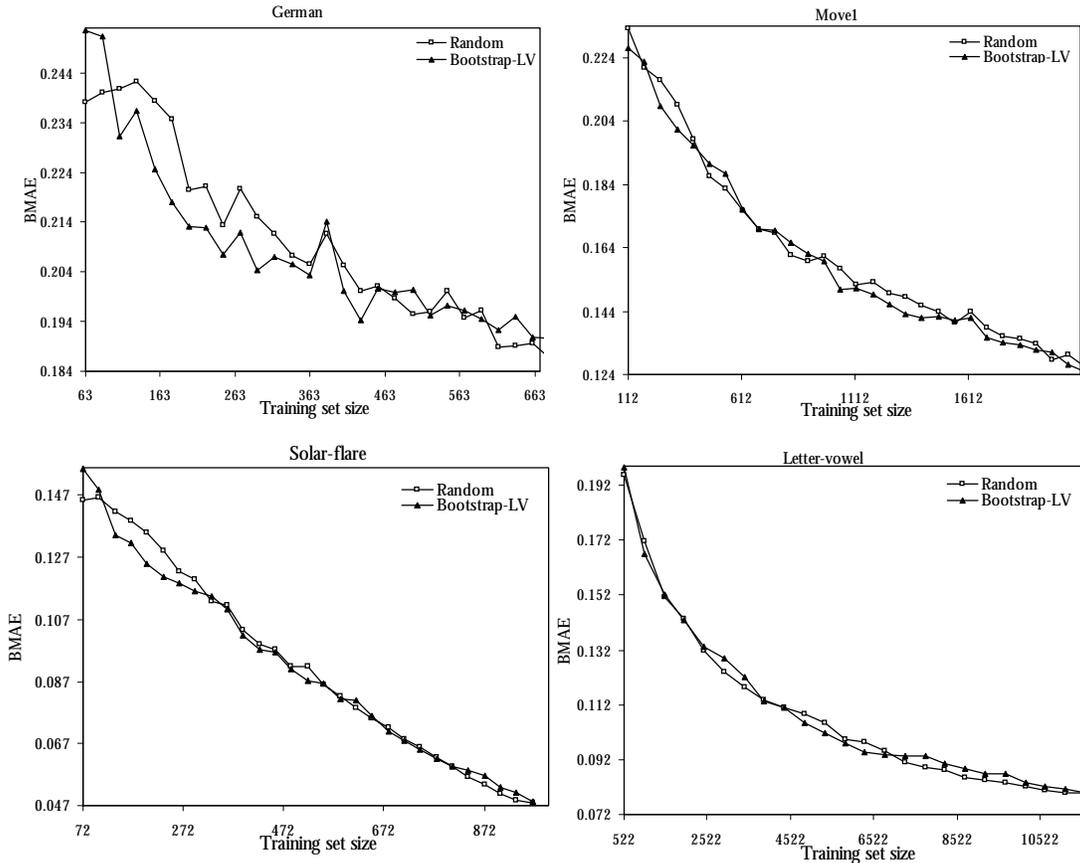


Figure 7: Learning curves for data sets where BOOTSTRAP-LV and RANDOM show comparable performance

### 4.3 Experiments with Bagged PETs Model

In order to verify that BOOTSTRAP-LV is effective not solely with PETs, we also experimented with a differently CPE learner. Bagged-PETs is an ensemble of bagged (Brieman, 1996) trees, where each tree is induced from a differently bootstrap (Efron

and Tibshirani, 1993) sample. The trees are used to estimate the class probability of an instance by averaging the CPEs of the PETs in the ensemble. Bagged-PETs are not comprehensible models, but have been shown generally to produce superior CPEs compared to single PETs (Bauer and Kohavi, 1999; Provost et al., 1998; Provost & Domingos, 2000).

`BOOTSTRAP-LV`'s performance for the bagged-PETs model concurs with the results obtained for individual PETs. Particularly, for 15 of the data sets `BOOTSTRAP-LV` exhibited phases-gained of more than 65% (in 13 of those phases-gained is more than 75%). The average of the top-20% example gain was 25% or higher in 11 of those data sets. Only in two data sets is phases-gained less than 40%.

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

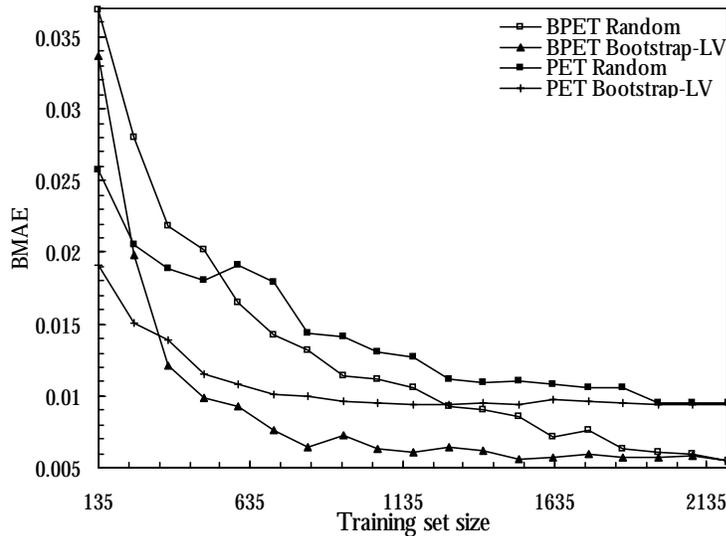


Figure 8: CPE learning curves for the Hypothyroid data set showing performance for `BOOTSTRAP-LV` and `RANDOM` with bagged PETs and single PETs.

#### 4.4 Other Evaluation Criteria

We also evaluated `BOOTSTRAP-LV` using alternative performance measures: the 0/1 mean squared error measure used by Bauer and Kohavi (1999), as well as the area under the 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 and Pendigits and Hypothyroid data sets, whose learning curves using BMAE were presented earlier. As AUC and

MSE also measure aspects of CPE accuracy they provide further evidence to the efficacy of the BOOTSTRAP-LV algorithm.

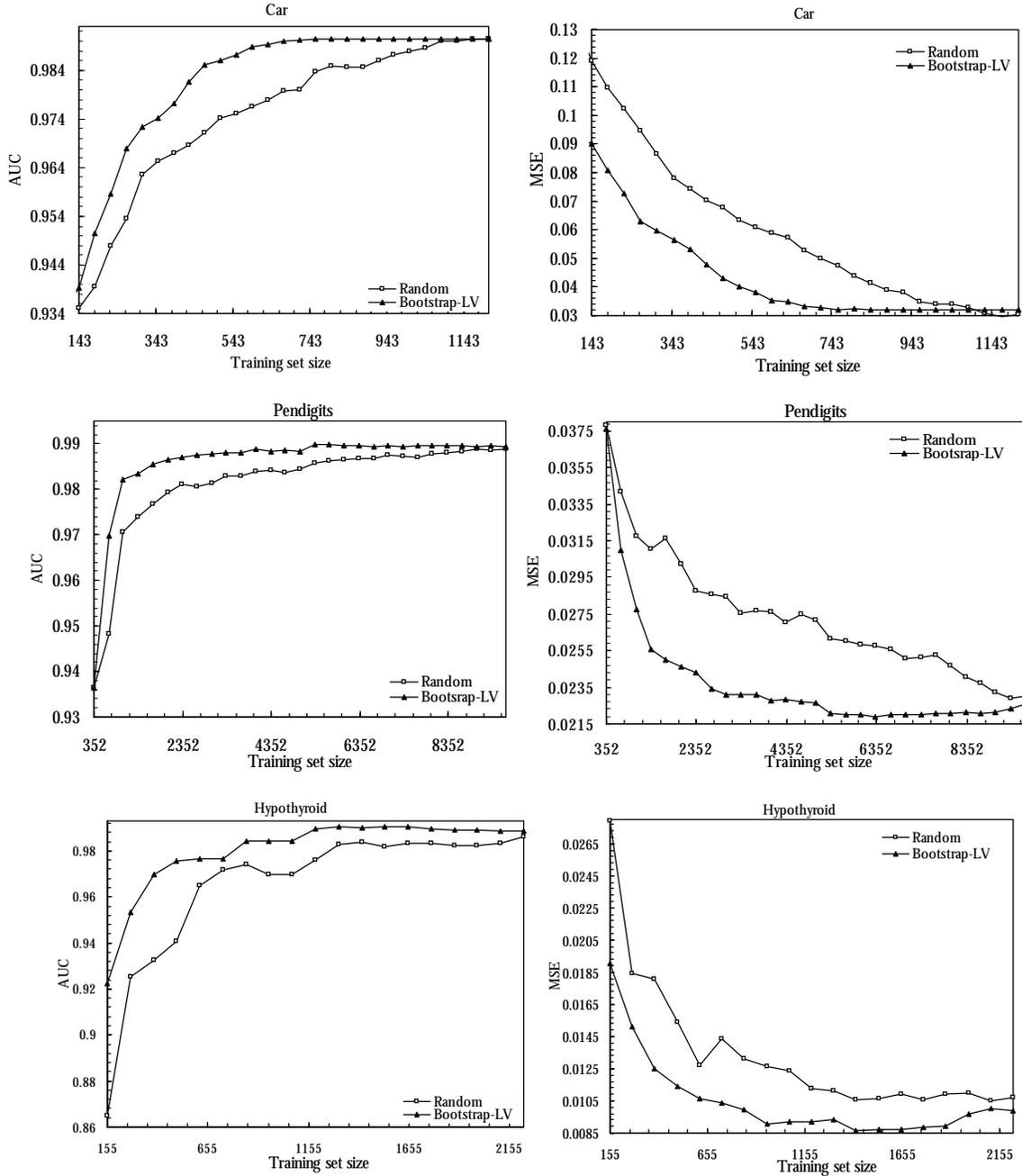


Figure 9: Learning curves for Area Under the ROC Curve and 0/1 MSE, comparing BOOTSTRAP-LV and RANDOM

#### 4.5 Comparisons with Uncertainty Sampling

We now compare `BOOTSTRAP-LV` with an active learning algorithm previously shown to improve classification accuracy. Since improved classification accuracy may also result from improved class probability estimation error, it is important to evaluate whether `BOOTSTRAP-LV` provides additional improvements. This comparison also provides interesting insights into the properties of these algorithms.

For the comparison we selected the well-known `UNCERTAINTY SAMPLING` algorithm (Lewis and Gale, 1994), proposed for 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 and Mamitsuka, 1998) (Iyengar *et. al*). Comparing these to active learning for single classifiers (with active or random selection) confounds the effects of active learning and producing ensembles<sup>4</sup>. `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 phases with positive gain for `BOOTSTRAP-LV` is between 50% and 60%. `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` wasn't 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 learning 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 learning 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

---

<sup>4</sup> Ensembles usually improve learning curves even with random selection

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 to the model. 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 examples needed and improvement in error using BOOTSTRAP-LV versus UNCERTAINTY SAMPLING

Data set	Phases with positive gain (%)	Top per gain (%)	Top gain (#)	Avg per gain (%)	Avg gain (#)	Top error reduction (%)
abalone	50.00	61.09	801	17.63	102	14.11
adult	<b>69.23</b>	<b>35.03</b>	<b>284</b>	<b>9.56</b>	<b>69</b>	<b>11.13</b>
breast cancer-w	55.56	49.37	144	10.90	15	20.20
car	62.50	50.46	68	9.95	6	36.30
<b>coding1</b>	<b>93.75</b>	<b>63.25</b>	<b>1027</b>	<b>31.77</b>	<b>686</b>	<b>6.74</b>
<b>connect-4</b>	<b>89.47</b>	<b>85.52</b>	<b>3230</b>	<b>43.89</b>	<b>1958</b>	<b>54.02</b>
contraceptive	50.00	54.87	126	11.76	21	10.01
<b>german</b>	<b>81.25</b>	<b>48.14</b>	<b>146</b>	<b>24.74</b>	<b>69</b>	<b>8.12</b>
<b>hypothyroid</b>	<b>71.43</b>	<b>62.30</b>	<b>307</b>	<b>17.10</b>	<b>85</b>	<b>62.72</b>
<b>kr-vs-kp</b>	<b>94.74</b>	<b>57.71</b>	<b>144</b>	<b>33.90</b>	<b>90</b>	<b>60.43</b>
<b>letter-a</b>	<b>85.00</b>	<b>44.34</b>	<b>771</b>	<b>15.50</b>	<b>395</b>	<b>21.29</b>
<b>letter-vowel</b>	<b>100.00</b>	<b>81.27</b>	<b>14210</b>	<b>63.80</b>	<b>11463</b>	<b>44.97</b>
<b>move1</b>	<b>100.00</b>	<b>62.89</b>	<b>247</b>	<b>39.96</b>	<b>194</b>	<b>16.29</b>
<b>ocr1</b>	<b>100.00</b>	<b>51.90</b>	<b>256</b>	<b>35.86</b>	<b>146</b>	<b>34.30</b>
<b>optdigits</b>	<b>100.00</b>	<b>44.13</b>	<b>1359</b>	<b>26.08</b>	<b>570</b>	<b>34.91</b>
<b>pendigits</b>	<b>95.00</b>	<b>60.85</b>	<b>1636</b>	<b>27.45</b>	<b>996</b>	<b>38.30</b>
<b>sick-euthyroid</b>	<b>100.00</b>	<b>84.12</b>	<b>1692</b>	<b>59.13</b>	<b>1093</b>	<b>40.51</b>
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  $I(A) = 1$ , if  $A$  is true, and  $I(A) = 0$  otherwise.

Assume also that  $p(\hat{f})$  is approximated with a standard normal distribution. This probability then is given by:

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

where  $\Phi$  is the upper tail area of the standard normal distribution, and  $E$  denotes a statistical expectation<sup>5</sup>. 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 be always beneficial to improve CPEs. For instance, consider a true class probability of 0.6 and a mean estimation for the estimation procedure 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 the role of weight sampling in generalization error reduction. 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 8 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 8, 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 is present for most data sets we experimented with. The superiority of `BOOTSTRAP-LV` over random sampling demonstrates first that the weights assigned to examples in `BOOTSTRAP-LV`, and which underlie the sampling process, provide useful information for selecting more informative training examples. The models induced when these weights are ignored and examples are sampled at

---

<sup>5</sup> Note that with respect to an active learning algorithm the estimation procedure whose variance and expectation appear in the formulation above also incorporates the choice of learning examples.

random (i.e., all weights are equal) are inferior to those induced when the assigned weights are incorporated to direct the sampling process.

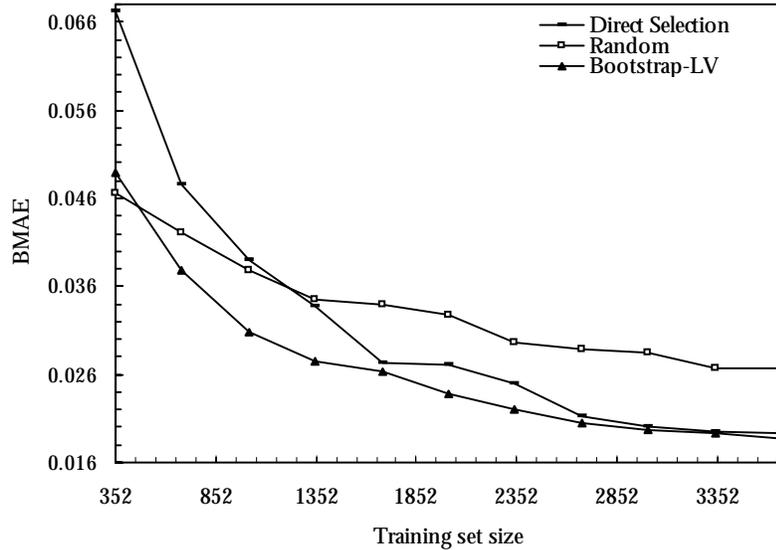


Figure 8: Learning curves for Weight-sampling, Direct Selection with `BOOTSTRAP-LV`'s effectiveness score, and `RANDOM`

Second, 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 worse error reduction for Direct Selection as observed for Direct Selection compared to `BOOTSTRAP-LV`. Apparently, making decisions 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. This we argue is enabled with weight sampling.

#### 4.6.1 Improving the Uncertainty Sampling Algorithm 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 9 below.

---

**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 9: `WEIGHTED UNCERTAINTY SAMPLING` Algorithm

Comparing the new `WUS` algorithm with `BOOTSTRAP-LV` for CPE we see that `WUS` is much more competitive with `BOOTSTRAP-LV` than `UNCERTAINTY SAMPLING`. A summary of the results is presented in Table 3. `BOOTSTRAP-LV` outperforms `WUS` for 8 data sets (in 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 10 shows CPE learning curves for the `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`. This can be primarily attributed 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,

whereas UNCERTAINTY SAMPLING merely considers the potential to improve the examined example’s classification accuracy.

Figure 11 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 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 discuss in section 4.2, weights assigned to examples in UWS 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 is detrimental to reducing CPE generalization error.

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

Data set	Phases with positive gain (%)	Top per gain (%)	Top gain (#)	Avg per gain (%)	Avg gain (#)	Top 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
<b>car</b>	<b>92.85</b>	<b>17.62</b>	<b>136</b>	<b>9.74</b>	<b>67</b>	<b>14.08</b>
<b>coding1</b>	<b>87.5</b>	<b>28.33</b>	<b>671</b>	<b>16.55</b>	<b>379</b>	<b>2.77</b>
connect-4	47.36	18.11	413	2.10	27	3.07
contraceptive	33.33	14.15	58	-2.51	-5	3.19
<b>german</b>	<b>68.75</b>	<b>43.01</b>	<b>133</b>	<b>17.24</b>	<b>43</b>	<b>6.66</b>
<b>hypothyroid</b>	<b>100</b>	<b>81.83</b>	<b>1782</b>	<b>65.41</b>	<b>1260</b>	<b>62.08</b>
kr-vs-kp	31.57	3.77	5	-1.34	-5	3.84
<i>letter-a</i>	<i>30</i>	<i>13.30</i>	<i>693</i>	<i>-7.23</i>	<i>-583</i>	<i>7.26</i>
letter-vowel	46.66	10.73	765	0.32	-24	3.44
<b>move1</b>	<b>88.88</b>	<b>26.73</b>	<b>138</b>	<b>13.45</b>	<b>62</b>	<b>8.11</b>
ocr1	62.5	13.74	66	2.68	16	11.20
optdigits	64.28	20.40	721	8.32	229	12.14
<b>pendigits</b>	<b>90</b>	<b>53.40</b>	<b>4064</b>	<b>36.39</b>	<b>2468</b>	<b>22.85</b>
<b>sick-euthyroid</b>	<b>100</b>	<b>53.61</b>	<b>859</b>	<b>41.33</b>	<b>537</b>	<b>17.54</b>
<i>solar-flare</i>	<i>0</i>	<i>2.46</i>	<i>-19</i>	<i>-16.79</i>	<i>-56</i>	<i>-9.11</i>
Weather	52.63	17.80	328	0.35	45	1.50
<b>yeast</b>	<b>73.33</b>	<b>41.79</b>	<b>189</b>	<b>16.59</b>	<b>64</b>	<b>5.03</b>

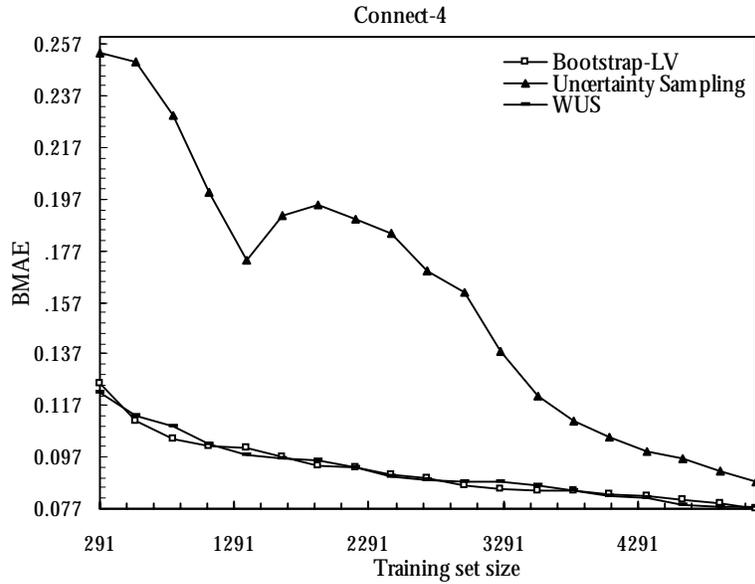
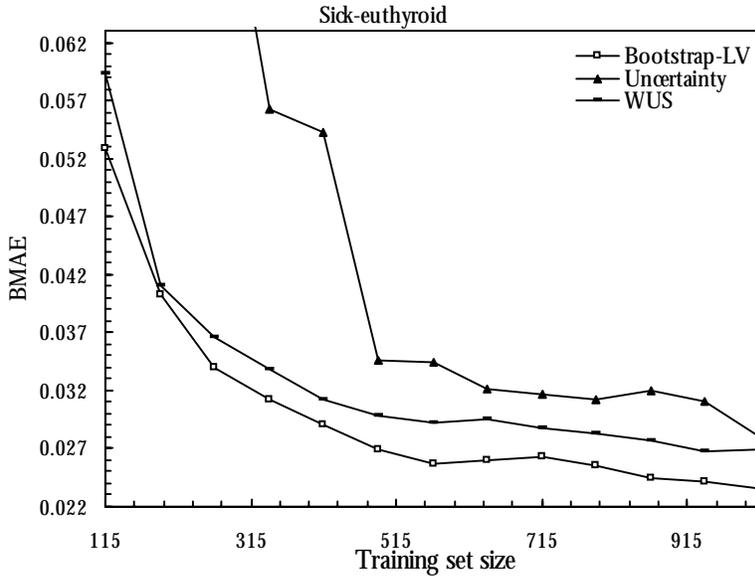


Figure 10: An example where WUS is superior to UNCERTAINTY SAMPLING and achieves performance comparable to that of BOOTSTRAP-LV



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 discuss earlier in the paper, by preferring examples whose CPE is close to 0.5 `WUS` identifies examples whose class is uncertain; however, such uncertainty also implies uncertainty regarding CPE and can benefit from gaining more relevant evidence. Yet `BOOTSTRAP-LV` produces better results because `WUC` may fail to identify all CPE uncertainties, particularly when these uncertainties do not imply class uncertainty. In addition, as we mentioned above, CPE that is close to 0.5 does not necessarily imply class (or 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, it is likely that `UWS` will 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 learning example on other examples in the space are likely to benefit from weight sampling.

## 6 Limitations

Although `BOOTSTRAP-LV`'s performance surpasses the performance of Weighted `UNCERTAINTY SAMPLING`, the latter also identify some CPE uncertainty and is computationally simpler. Hence `WUS` may be considered for active learning of CPE when computational concerns are particularly critical.

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

`BOOTSTRAP-LV` also does not address computational concerns, as do Lewis and Catlett (Lewis and Catlett, 1994). Indeed it requires the induction of multiple models from a set of bootstrap samples. However, 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 only critical in the early part of the curve. Therefore, if the number of training examples remains relatively small multiple model inductions from bootstrap samples do not constitute a considerable computational toll. `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.

## 7 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 remarkably well. The evaluation encompasses a wide range of benchmark domains providing comprehensive evidence for the efficacy of the algorithm. We show how the information provided by the effectiveness scores produces better results than can be obtained with random sampling (i.e., when all weights are equal). We also show that BOOTSTRAP-LV outperforms an existing active learning method, UNCERTAINTY SAMPLING. We investigate the properties of the algorithms to explain our empirical results. In particular, the experimental results demonstrate how both the weights assigned to potential training examples and the weight sampling procedure combine to produce superior CPEs. We also examine the properties of the UNCERTAINTY SAMPLING algorithm compared to those of BOOTSTRAP-LV to explain the comparison in performance for estimating class probabilities.

Lastly, we use the results of this investigation to propose 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. 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. Our empirical analysis 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 resorts to decision-theoretic approaches for evaluating alternatives, requiring the estimation of probabilities of events or classes to score alternative outcomes. The cost-sensitive nature of such environments can greatly benefit from active learning of class probability estimations and rankings of alternatives. BOOTSTRAP-LV was designed to address this need. The paper provides a comprehensive study of the performance of the BOOTSTRAP-LV algorithm with respect to several alternative approaches and highlights the properties responsible for the observed behavior.

## Acknowledgments

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

## References

- Abe, N. and 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. (1998). Queries and concept learning. *Machine Learning* 2:319-342, 1988.
- Bauer, E., Kohavi, R. (1999). An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning* 36, 105-142 (1999).
- Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science, 1998 [<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, 1997.
- L. Breiman. (1996). Bagging predictors. *Machine Learning* 26(2), 123--140, 1996
- Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning. In *Proceedings of the Ninth European Conference on Artificial Intelligence*, 147-149, 1990.
- Cohn, D., Atlas, L. and Ladner, R. (1994). Improved generalization with active learning. *Machine Learning* 15:201-221, 1994.
- Cohn, D., Ghahramani, Z., and Jordan M. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129-145, 1996.
- Cohen, W. W. and Singer, Y. (1999). A simple, fast, and effective rule learner. In proceedings of the National Conference of the *American Association of Artificial Intelligence*, 335-342, 1999.
- Efron, B. and Tibshirani, R. (1993). *An introduction to the Bootstrap*, Chapman and Hall, 1997.
- Freund, Y. and Shcapiro R. (1996). Experiments with a new boosting algorithm. In *Proceedings of the International Conference on Machine Learning* 148-156, 1996.
- Iyengar, V. S., Apte, C., and Zhang T. (2000). Active Learning using Adaptive Resampling. In *Sixth ACM Special Interest Group on Knowledge Discovery in Databases International Conference on Knowledge Discovery and Data Mining* 92-98.

- Lewis, D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Association for Computing Machinery-Special Interest Group on Information Retrieval, 1994*, 3-12.
- Lewis, D. D., and Catlett, J. (1994). Heterogeneous Uncertainty Sampling. In *Proceedings of the Eleventh International Conference on Machine Learning* 148-156, 1994.
- [Perlich, C.; Provost F.; and Seminoff, J. S. (2001). Tree induction vs. logistic regression: A learning-curve analysis. *CeDER Working Paper #IS-01-02, Stern School of Business, NYU*.
- Provost, F.; Fawcett, T.; and Kohavi, R. (1998). The case against accuracy estimation for comparing classifiers. In *Proceedings of the International Conference on Machine Learning* 445-453, 1998.
- Provost, F. and 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. Morgan Kaufman, San Mateo, California, 1993.
- H. S. Seung, M. Opper, and H. Smopolinsky. (1992). Query by committee. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, 287-294, 1992.
- Simon H. A. and Lea G., (1974). Problem solving and rule induction: A unified view. In L.W. Gregg (ed.), *Knowledge and Cognition*. Chap. 5. Potomac, MD: Erlbaum, 1974.
- Turney, P.D. (2000). Types of cost in inductive concept learning, Workshop on Cost-Sensitive Learning at ICML-2000, Stanford University, California, 15-21.
- Valiant L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27:1134-1142, 1984.
- Winston, P. H. (1975). Learning structural descriptions from examples. In “*The Psychology of Computer Vision*”, P. H. Winston (ed.), McGraw-Hill, New York, 1975.
- Zadrozny B. and Elkan C. (2001). Learning and making decisions when costs and probabilities are both unknown. In the proceedings of *The Seventh ACM Special Interest Group on Knowledge Discovery and Data Mining International Conference on Knowledge Discovery and Data Mining* 204-212, 2001.