VARIANCE-BASED ACTIVE LEARNING

Maytal Saar-Tsechansky
and
Foster Provost

IS-00-05

Center for Digital Economy Research
Stern School of Business
Working Paper IS-00-05

Variance-based Active Learning

Maytal Saar-Tsechansky
Department of Information Systems
Leonard N. Stern School of Business
New York University

Foster Provost
Department of Information Systems
Leonard N. Stern School of Business
New York University

Abstract

For many supervised learning tasks, the cost of acquiring training data is dominated by the cost of class labeling. In this work, we explore active learning for class probability estimation (CPE). Active learning acquires data incrementally, using the model learned so far to help identify especially useful additional data for labeling. We present a new method for active learning, Bootstrap-LV, which chooses new data based on the variance in probability estimates from bootstrap samples. We then show empirically that the method reduces the number of data items that must be labeled, across a wide variety of data sets. We also compare Bootstrap-LV with Uncertainty Sampling, an existing active-learning method for maximizing classification accuracy, and show not only that Bootstrap-LV dominates for CPE but also that it is quite competitive even for accuracy maximization.

1 Introduction

In order to undertake supervised learning for classification problems, it is necessary to obtain data with class labels. Procuring these labels can be costly. Experts may need to be consulted, users may need to provide feedback, or a system may need to make suboptimal decisions in order to label data. For example, consider an automated commerce engine that models customer response to offers and uses these models to target future offers. Such a system is faced continually with the choice of using the model learned so far to try to maximize revenue, versus making possibly suboptimal offers in order to label more training data. If possible, the system should focus on those data points that will accelerate learning the most.

For this paper, we consider supervised learning for class probability estimation. Class probability estimates (CPEs) can be combined with decision-making costs/benefits to minimize expected cost (maximize expected benefit). Also, we are interested primarily in comprehensible models, so we use decision trees to produce class probability estimates [Smyth *et al.* 1995; Bauer and Kohavi 1998; Provost *et al.* 1998]. However, the method we introduce applies to any technique for learning CPEs.

Active learning incrementally acquires training data, using the model learned "so far" to select subsequent examples. In the case of expensive labeling, active learning methods can be used to reduce the number of instances that must be labeled, in order to achieve a particular level of accuracy.

Figure 1 shows the ideal behavior of an active learner. The horizontal axis represents the number of training data, and the vertical axis represents the error rate of the model learned. Each *learning curve* shows how error rate decreases as more training data are used. The upper curve uses random sampling; the lower curve uses active learning. The two curves form a "banana" shape: very early on, the curves are comparable because no model is available yet for active learning. However, very quickly the active learning curve accelerates, because it chooses training data carefully. Given enough data, random sampling catches up.

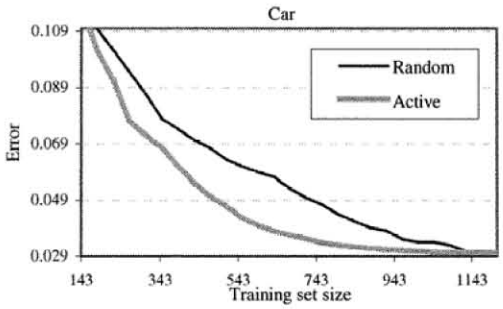

Figure 1: Learning curves for active learning and random sampling

We introduce a new active-learning technique, inspired by prior work. Bootstrap-LV uses bootstrap samples of the existing training data to examine the variance in the probability estimates for not-yet-labeled data. We show empirically that across a wide range of data sets Bootstrap-LV decreases the number of labeled instances needed to achieve accurate probability estimates, or alternatively increases the accuracy of the probability estimates for a fixed number of training instances. We also show that Bootstrap-LV is surprisingly effective even for accuracy maximization.

2 Active Learning: Prior Work

The 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 searched. 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, Valiant 1984]. The term *active learning* was coined later to describe induction where the algorithm assumes control over the selection from a set of potential training examples [Cohn *et al.* 1994]. A generic algorithm for active learning is shown in Figure 2.

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.
While stopping criteria not met
  /* perform next phase: */
    Apply inducer I to L
    For each example $\{x_i \mid x_i \in UL\}$ compute ES_i
    Select/Sample a subset S of size M from UL based on ES_i
    Remove S from UL, label S, and add S to L
Output: estimator E induced with I from the final labeled set L

Figure 2: Generic Active Learning Algorithm

A learner first is applied to an initial (usually small) 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 given an effectiveness score ES_i based on its contribution to an objective function, reflecting the estimated magnitude of its contribution to subsequent learning (or simply whether it will or will not contribute). Examples are selected either directly by selecting the top M examples in the ranking, or via a weighted sample, where the probability of an example to be sampled is proportional to ES_i. Usually, multiple examples, rather than a single example, are selected at each phase due to computational constraints. Once examples are selected, their labels are obtained (e.g., via a query to an expert) before being added to L on which the learner is applied next.

Cohn et al [Cohn *et al.* 1994] determine ES_i based on identifying what they called the "region of uncertainty," defined such that concepts from the current version space are inconsistent with respect to examples in the region. The region of uncertainty is redetermined at each phase and subsequent examples are selected from this region. The main practical problem with this approach is that the estimation of the uncertainty region becomes increasingly difficult, as the concept becomes more complex. In addition, for complex concepts the region of uncertainty may initially span the entire domain before the concept is well understood, rendering the selection process ineffective. A closely related approach is Query By Committee [Seung *et al.*1992] classifiers are *sampled* from the version space, and the examples on which they disagree are considered for labeling.

Practically, our technique is inspired most by the work of Lewis and Gale [Lewis and Gale 1994]. In their Uncertainty Sampling, ES_i is based on the estimated probability of binary class membership. Specifically, a probabilistic classifier is employed, and examples whose probabilities of class membership are closest to 0.5 are considered for labeling. A closely related technique [Iyegar *et al.* 2000] considers adaptive resampling to help compute ES_i: examples estimated to be *misclassified* in the next phase are assigned higher probability to be sampled at each phase.

Our approach also uses the generic algorithm shown in Figure 2, but instead of looking for examples whose classification is likely to be erroneous or uncertain, we look for examples whose CPEs are uncertain.

Theoretically, our technique is inspired most by the approach presented by Cohn et al. [Cohn *et al.* 1996] for statistical learning models. At each phase the learner computes the expectation of the model's variance over the example space resulting from adding each candidate example to the training set. However, this approach requires knowledge of the underlying domain, as well as the computation in closed form of the learner's variance, a constraint that renders this method impracticable for arbitrary models.

With our approach ES_i is based on the variance of the current model on each specific example, $x_i \in UL$. To contrast with prior work we call this *local* variance, or *LV*.

3 Our Approach

We now describe our approach, which actively samples examples from UL to learn *class probability estimates* (CPE) from fewer examples. The description we provide here pertains to binary class problems where the set of class labels is $C = \{0,1\}$. As the discussion above indicates, we wish to add to L examples that are likely to improve the available evidence pertaining to poorly understood subspaces of the domain.

Ideally, the most direct indication of the *quality* of the current class probability estimation for example x_i is the discrepancy between the estimated probability and its true probability. However, the true class probability for an instance is not known, nor is its actual class. Therefore we use the local variance to estimate this quality. If the estimated LV is high compared to that of other examples, we infer that this example is "difficult" for the learner to estimate given the available data, and increase the probability that it will be

sampled next. Otherwise, if the LV is low, we interpret it as an indication that either the class probability is well learned or, on the contrary, that it will be extremely difficult to improve. We therefore decrease probability of these examples being added to L.

Given that a closed-form computation/estimation of this local variance may not (easily) be obtained, we estimate it empirically. We generate a set of k *bootstrap* subsamples [Efron and Tibshirani, 1993] B_j, $j=1,...,k$ from L, and apply the inducer I on each subsample to generate k estimators E_j $j=1,...,k$ respectively. For each example in UL we estimate the variance in CPEs given by the estimators $\{E_j\}$, $j=1,...,k$. Each example in UL is assigned a weight, which determines its probability of being sampled, and which is proportional to the variance of the CPEs. More specifically, the distribution from which examples are sampled is given by $D_s(x_i)=\frac{\left\{\sum_{j=1}^{k}\left[\left(p_j(x_i)-\overline{p}_i\right)^2\right]\right\}/\overline{p}_i}{R}$ where $p_j(x_i)$ denotes the estimated probability an estimator E_j assigns to the event that example x_i belongs to class 0, $\overline{p}_i$ is the average $\frac{\sum_{j=1}^{k} p_j(x_i)}{k}$, and R is a normalizing factor $R=\sum_{i=1}^{size(UL)}\left\{\sum_{j=1}^{k}\left[\left(p_j(x_i)-\overline{p}_i\right)^2\right]\right\}/\overline{p}_i$ so that D_s is a distribution. This is the Bootstrap-LV algorithm, shown in Figure 3.

Algorithm Bootstrap-LV

1 **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, $j=1,...,k$ from L

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

5 For all examples $\{x_i \mid x_i \in UL\}$ compute $D_s(x_i)=\frac{\left\{\sum_{j=1}^{k}\left[\left(p_j(x_i)-\overline{p}_i\right)^2\right]\right\}/\overline{p}_i}{R}$ (R is a normalizing factor so that D_s is a distribution)

6 Sample a subset S of M examples from UL without replacement with weights from the probability distribution D_s

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

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

Figure 3: The Bootstrap-LV Algorithm

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 oversampling from the respected region. Therefore we also divide the estimated variance by the average value of the minority-class probability estimates. We estimate this value once from the initial random sample.

4 Experiments and Evaluation

To evaluate its performance, we applied Bootstrap-LV to 20 data sets from the UCI repository. Data sets with more than two classes were mapped into two-class problems. As mentioned above, because we are interested in comprehensible models, for our experiments the underlying probability estimator is a probability estimation tree (PET)--a C4.5 decision tree [Quinlan, 1993] for which the Laplace correction is applied at the leaves [Bauer and Kohavi, 1998; Provost et al., 1998]. The Laplace correction has been shown to improve significantly the CPEs produced by decision trees [Bauer and Kohavi, 1998].

If the true underlying class probability distribution were known, an evaluation of an estimator's performance could be based on a measure of the actual error in probability estimation. Since the true probabilities of class membership are not known, we compare the probabilities assigned by the models induced at each phase with those assigned by a "best" estimator, **B**, as surrogates to the true probabilities. **B** is induced from the entire set of examples ($UL \cup L$), using bagged-PETs, which have been shown to produce superior probability estimates compared to individual PETs [Bauer and Kohavi 1998]. We compute the mean absolute error (MAE) for an estimator E with respect to **B**'s estimation, denoted by *BMAE*, given by $BMAE=\frac{\sum_{i=1}^{N}\left|p_B(x_i)-p_E(x_i)\right|}{N}$, where $p_B(x_i)$ is the estimated probability given by **B**, $p_E(x_i)$ is the probability estimated by E, and N is the number of examples examined.

We compare the performance of Bootstrap-LV against a method denoted by *Random*, where estimators are induced with the same inducer and training set-size, but for which examples are sampled at random. We show the comparison for 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 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 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*. Figure 1 above shows results for the *Car* data set (where *Active* refers to *Bootstrap-LV*). As shown in Figure 1, the error of the estimator induced with our approach 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 with fewer examples.

For some data sets Bootstrap-LV exhibited even more dramatic results. For instance, Figure 4 shows results for the

Pendigits data set. Bootstrap-LV achieved its almost minimal level of error at 5000 examples. It required Random more than 9300 examples to obtain this error level.

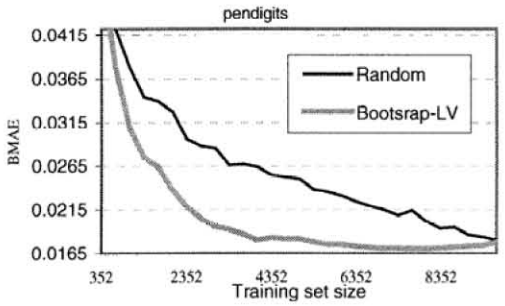

Figure 4: learning curves for CPE

For 5 data sets, however, our approach did not succeed in accelerating learning much or at all, as can be shown for the W data set in Figure 5. Note, however, that neither curve consistently resides above the other and the two methods exhibit comparable performance.

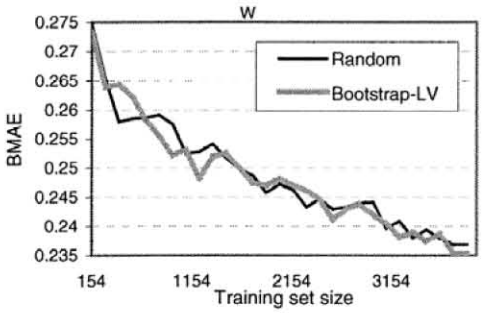

Figure 5: Learning curves for the w data set

Table 1 presents a summary of our results for all the data sets. The primary motivation for applying active sampling techniques is to allow learning with fewer examples. Table 1 provides a set of measures pertaining to the number of examples gained via Bootstrap-LV with respect to Random. The second column shows the percent of phases in which Bootstrap-LV produced the same level of accuracy with fewer examples compared to Random. The third and fourth columns show the percentage and number of examples *gained* by applying Bootstrap-LV, respectively. The gain is calculated as the difference between the number of examples used by Random and that used by Bootstrap-LV to obtain the same accuracy. The percentage is calculated from the number of examples used by Random. Because of the natural banana shape even for the ideal case, the performance of estimators induced from any two samples cannot be considerably different at the final phases, thus the averages as well as the percentage of positive gain merely provide an indication of whether our approach provides superior estimations. It is important, also to observe 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, the average of the top 20% gains. Columns 5 and 6 of Table 1 show the average percent and average number of examples gained for the top 20% gains respectively. It is important that these figures be viewed in tandem with column 2 (pos gain), to ensure that there is in fact a banana shape to the graph.

	Examples					Error (%)	
Data set	Pos gain (%)	% Avg gained	Avg # gained	top 20% (%)	# gained top 20%	Avg top 20%	Avg top 20% (% from maximal gain)
abalone	92.5	34.9	574	76.9	1152	10.1	64.0
adult	96	17.8	302	30.2	585	6.6	25.0
bc-wisc	100	23.8	44	51.6	110	9.3	41.0
car	89.6	23.3	155	35.4	281	31.3	53.3
coding1	80	16.2	228	47.1	475	2.5	28.9
connect-4	100	45.5	984	75.4	1939	9.5	27.5
contraceptive	93.7	18.4	55	42.3	129	5.7	31.3
german	57.1	5.8	7	46.5	113	5.9	31.0
hypothyroid	100	64.6	705	69.0	1233	41.1	72.4
kr-vs-kp	100	18.1	37	27.1	57	25.5	30.8
Letter-a	72.4	14.5	229	24.8	529	10.4	26.0
Letter-vowel	50	2.1	121	12.8	429	3.4	18.0
move1	65	17.2	23	68.4	75	3.9	12.8
ocr1	93.7	24.5	83	42.9	168	21.7	65.0
optdigits	94.4	24.5	412	50.0	762	32.6	47.8
pendigits	100	61.0	3773	68.6	5352	29.9	75.6
sick-euthyroid	93.1	45.2	600	70.2	924	26.2	58.5
solar-flare	64.2	13.5	25	41.5	58	6.3	9.9
w	41.6	-10.4	-46	35.9	438	1.7	20.1
yeast	75	23.6	79	58.7	159	4.9	30.8

Table 1: Examples and error gain measures for CPE

Table 1 also includes summary results pertaining to the error rates achieved by both methods for the same number of examples. Column seven presents the average percent gain of the top 20% error reduction. For some data sets the generalization error for the initial training sets was small and was not considerably reduced even when the entire data was used for training (e.g., for connect-4, only 34% error reduction was obtained, from 11.7 to 7.7). We therefore also provide in the last column, the top 20% error gain as a percentage of the reduction required to obtain the minimal error (the latter is referred to in the table as *maximal gain*). In the Adult data set, for instance, Bootstrap-LV exhibited only 6.6% error gain (for the top 20%), but this improvement constitutes 25% of the possible improvement were the entire data set used for training.

Since not all plots can be presented due to space constraints, we aimed at expressing in the table various performance measures that would provide a comprehensive perspective. The criterion that we apply to assess Bootstrap-LV's success over Random is the combination of the following: the minimal positive gain should be above 60%, both the average examples and error gains are positive, and the top 20% error from maximal gain is 25% or higher. If the positive gain is between 40% and 60% we consider both methods to be comparable, and when it is below 40% we consider Bootstrap-LV to be inferior. 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 the positive gain is 75% and above. In 13 of those, more than 30% of the examples were gained (for the top 20%), and in 9 data sets our method used less than 50% of the

number of examples required for Random to achieve the same level of accuracy. For the Sick-euthyroid data set, for instance, Bootstrap-LV gradually improves until it requires fewer than 30% of the examples required by Random to obtain the same level of accuracy. As we mentioned earlier, these results pertain to the top 20% improvement, thus, the maximal gain is indeed higher. For a single data set (W) Bootstrap-LV exhibited a negative average examples gain. However, both the percent of positive gain, showing that Bootstrap-LV has exhibited examples gain in 41% of phases examined, and Figure 5, indicate that the two methods indeed exhibit comparable performance for this data set.

The measures pertaining to number of examples gained and error gain complement each other and may provide interesting insights. For instance, the number of examples gained can help evaluate the "difficulty" in error reduction in terms of the number of examples required by Random to obtain such reduction. For example, although the average top 20% error gain for Connect-4 was less than 10%, Table 1 shows that it required Random 984 additional examples on average to obtain the same improvement.

A single data set, Letter-vowel, exhibited a negative average error gain. However, exactly 50% of the phases have shown positive examples gain, indicating that Random indeed does not exhibit superior performance overall and that both methods exhibited similar performance.

We also assessed both methods with two alternatives to BMAE: the mean squared error measure proposed by Bauer and Kohavi [1998], as well as the area under the ROC curve [Bradley 1997]. The results for these measures agree with those obtained with BMAE. For example, Bootstrap-LV generally leads to fatter ROC curves with fewer examples.

Tree-based models offer a comprehensible structure that is important in many decision-making contexts. However, they do not provide the best probability estimates. In order to assess Bootstrap-LV's performance on a better CPE learner, we experimented with bagged-PETs, which are not comprehensible models, but have been shown to produce markedly superior CPEs [Bauer and Kohavi 1998].

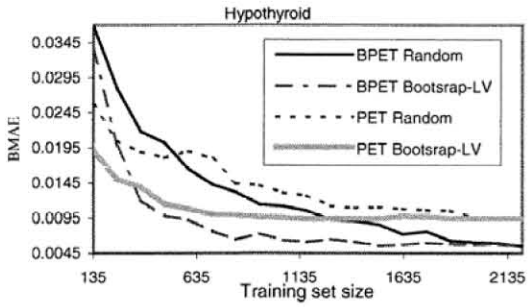

Figure 6: BMAE learning curves for the Hypothyroid data set

The results for the the bagged-PETs model also agreed with those obtained for individual PETs. Particularly, for 15 of the data sets Bootstrap-LV exhibited a positive-example gain of more than 65% (in 13 of those the positive-example gain is more than 75%). The average top example-gain was 25% or higher in 11 of those data sets. Only in two data sets was the positive-example gain less than 50%. Figure 6 shows a comparison between Bootstrap-LV and Random for individual and bagged-PETs. As expected, 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 required for Random.

5 Comparison with Uncertainty Sampling

As described above, Uncertainty Sampling [Lewis and Gale, 1994] was proposed for binary text classification. However, it too samples examples that are not well understood by the model. Since it was shown to improve a model's classification accuracy, it is bound to improve the model's CPE as well. It is therefore interesting to compare the improvements exhibited by Bootstrap-LV against Uncertainty sampling. We present a summary of the comparison results in Table 2.

	Examples					Error (%)	
Data set	Pos gain (%)	% Avg gained	Avg # gained	top 20% (%)	# gained top 20%	Avg top 20%	Avg top 20% (% from maximal gain)
abalone	50.00	17.63	102	61.09	801	14.11	57.57
adult	**69.23**	**9.56**	**69**	**35.03**	**284**	**11.13**	**27.18**
bc-wisc	55.56	10.90	15	49.37	144	20.20	43.91
car	**62.50**	**9.95**	**6**	**50.46**	**68**	**36.30**	**43.26**
coding1	**93.75**	**31.77**	**686**	**63.25**	**1027**	**6.74**	**49.26**
connect-4	**89.47**	**43.89**	**1958**	**85.52**	**3230**	**54.02**	**82.91**
contraceptive	50.00	11.76	21	54.87	126	10.01	29.13
german	**81.25**	**24.74**	**69**	**48.14**	**146**	**8.12**	**37.63**
hypothyroid	**71.43**	**17.10**	**85**	**62.30**	**307**	**62.72**	**77.74**
kr-vs-kp	**94.74**	**33.90**	**90**	**57.71**	**144**	**60.43**	**64.07**
Letter-a	**85.00**	**15.50**	**395**	**44.34**	**771**	**21.29**	**30.65**
Letter-vowel	**100.00**	**63.80**	**11463**	**81.27**	**14210**	**44.97**	**43.41**
move1	**100.00**	**39.96**	**194**	**62.89**	**247**	**16.29**	**36.26**
ocr1	**100.00**	**35.86**	**146**	**51.90**	**256**	**34.30**	**61.75**
optdigits	**100.00**	**26.08**	**570**	**44.13**	**1359**	**34.91**	**58.16**
pendigits	**95.00**	**27.45**	**996**	**60.85**	**1636**	**38.30**	**58.03**
sick-euthyroid	**100.00**	**59.13**	**1093**	**84.12**	**1692**	**40.51**	**64.49**
solar-flare	0.00	-16.66	-69	-2.98	-17	-1.64	-6.54
w	56.25	6.32	3	35.06	351	1.98	24.74
yeast	53.33	7.74	3	40.38	121	6.03	28.88

Table 2: Summary results CPE of Bootstrap-LV against Uncertainty sampling

Bootstrap-LV exhibited superior performance in 13 of the data sets. In 6 data sets both methods exhibited comparable performance, where the positive examples gain for Bootstrap-LV was between 50% and 60%.

Uncertainty Sampling exhibited superior performance in one data set, solar-flare, for which it consistently produced better probability estimations. In 9 out of the 14 data sets in which Bootstrap-LV was superior, the average top error reduction was more than 30%. These results demonstrate that Bootstrap-LV has a solid advantage when compared to Uncertainty Sampling for class probability estimation. It is important to emphasize once again that indeed Uncertainty Sampling was not designed to improve class probability estimation, but rather to improve classification accuracy.

We also compared the performance of Uncertainty Sampling against Bootstrap-LV for improving classification accuracy. Since Bootstrap-LV was found to improve CPEs, a similar effect may be obtained for classification accuracy, but not necessarily: Bootstrap-LV may select examples to

improve class probability estimation even when the estimated decision boundary required for classification is already well understood, thereby "wasting" examples that do not improve classification accuracy.

Our results for classification accuracy show that in 11 data sets Bootstrap-LV exhibited superior performance. Uncertainty Sampling was superior in 7 data sets and both methods exhibited comparable performance for the remaining two. These results indicate that although Bootstrap-LV is not generally superior to Uncertainty Sampling for classification tasks, it should be considered a viable alternative--it often yields much better performance. Interestingly, in most cases where Bootstrap-LV does not dominate, it performs better in the initial phases, whereas Uncertainty sampling surpasses Bootstrap-LV in later phases. This phenomenon is demonstrated in Figure 7 for the Breast-Cancer data set. Recall that Uncertainty Sampling uses the CPEs to determine the potential contribution of an example for learning. Therefore, its performance will be sensitive to the accuracy of the CPEs. Poor CPEs produced in the initial phases undermine the data selections by Uncertainty Sampling. On the other hand, in later phases, more accurate probability estimations allow the selection process to focus in on the decision boundary. Bootstrap-LV, on the contrary, focuses early on improving the CPEs, and therefore performs well even very early on the learning curve, however, later on it indeed "wastes" examples to improve CPE as described above.

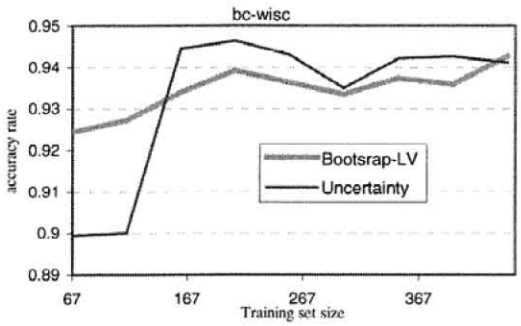

Figure 7: Classification accuracy rate

In light of this typical behavior, a better strategy for actively improving classification accuracy may be a hybrid approach, where Bootstrap-LV is applied in initial phases and Uncertainty Sampling in later ones. When to switch is still an open question.

6 Conclusion

We introduced a new technique for active learning. Bootstrap-LV was designed to use fewer labeled training data to produce better class-probability estimates from fewer labeled data. We showed empirically that it does this remarkably well, and performs better than prior active learning methods. We also showed that Bootstrap-LV is competitive with prior methods even for accuracy maximization. These last results suggest a hybrid strategy that may be even more effective than either technique alone.

References

[Angluin 1988] Angluin, D. Queries and concept learning. *Machine Learning*, 2:319-342, 1988.

[Bauer and Kohavi, 2998] Bauer, E., Kohavi, R. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, 36, 105-142 (1998).

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

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

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

[Efron and Tibshirani, 1993] Efron, B. and Tibshirani, R. *An introduction to the Bootstrap*, Chapman and Hall, 1997.

[Iyegar *et al.* 2000] Iyegar, V. S., Apte, C., and Zhang T. Active Learning using Adaptive Resampling. *In SIGKDD-2000.* Pages 92-98, Aug 2000.

[Lewis and Gale 1994] Lewis, D. D. and Gale, W. A. A sequential algorithm for training text classifiers. In *ACM-SIGIR-94*, pages 3-12, Springer-Verlag.

[Seung *et al.* 1992] H. S. Seung, M. Opper, and H. Smopolinsky. Query by committee. *In Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 287-294, 1992.

[Provost et al 1998] Provost, F.; Fawcett, T.; and Kohavi, R. The case against accuracy estimation for comparing classifiers. In *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998

[Quinlan 1993] Quinlan, J. R.. *C4.5: Programs for machine learning*. Morgan Kaufman, San Mateo, California, 1993.

[Simon and Lea 1974] Herbert A. Simon and Glenn Lea, Problem solving and rule induction: A unified view. In L.W. Gregg (Ed.), Knowledge and cognition (Chap. 5). Potomac, MD: Erlbaum, 1974.

[Smyth *et al.* 1995] Smyth, P.; Gray, A and Fayyad U. M. Retrofitting Decision Tree Classifiers using density estimation. *In Proceedings of the 12th International Conference on Machine Learning*. Pages 506-514, 1995.

[Valiant 1984] Valiant L. G. A theory of the learnable. *Communications of the ACM*, 27:1134-1142, 1984.

[Winston 1975]. Winston, P. H. Learning structural descriptions from examples. In *"The Psychology of Computer Vision"*, edited by Patrick H. Winston, McGraw-Hill Book Company, New York, 1975.