2015 IEEE International Conference on Data Mining

Accurate Estimation of Generalization Performance for Active Learning

Aubrey Gress, Ian Davidson
University of California, Davis
adgress@ucdavis.edu, davidson@cs.ucdavis.edu

***Abstract*—Active learning is a crucial method in settings where a human labeling of instances is challenging to obtain. The typical active learning loop builds a model from a few labeled instances, chooses informative unlabeled instances, asks an Oracle (i.e. a human) to label them and then rebuilds the model. Active learning is widely used with much research attention focused on determining which instances to ask the human to label. However, an understudied problem is estimating the accuracy of the learner when instances are added actively. This is a problem because regular cross validation methods may not work well due to the bias in selecting instances to label. We show that existing methods to address the issue of estimating performance are not suitable for practitioners since the scaling coefficients can have high variance, the estimators can produce nonsensical results and the estimates are empirically inaccurate in the classification setting. We propose a new general active learning method which more accurately estimates generalization performance through a sampling step and a new weighted cross validation estimator. Our method can be used with a variety of query strategies and learners. We empirically illustrate the benefits of our method to the practitioner by showing it is more accurate than the standard weighted cross validation estimator and, when used as part of a termination criterion, obtains more accurate estimates of generalization error while having comparable generalization performance.**

I. Introduction

Active learning methods are popular in domains where labeling is expensive or time consuming. The standard active learning setting is to start with zero or a few labeled points and repeatedly build a model, select unlabeled instances for an Oracle to label which maximally improve accuracy. The aim here is for the learner to achieve a high accuracy while minimizing human effort by querying as few instances as possible. Though much work has focused on active learning query schemes (i.e. which instances to choose to be labeled) [1] very little work has explored measuring the performance of the trained model. This is a non-trivial problem as traditional cross validation methods are not suitable for estimating the generalization performance of actively trained learners since they assume unbiased samples. However, accurate performance estimates are critical to practitioners for two reasons. Firstly, after adding more instances the performance improvement is used to determine if accuracy has converged or if further rounds of active learning are required. Secondly, it is important to have a confident estimate of performance to determine if a model is even worth deploying [2].

Many academics studies on active learning are flawed from the practitioner's perspective due to two issues. Firstly, they typically report active learning performance using an independent hold out set, but in practice this is unreasonable since any difficult to obtain labeled data would be used for training rather than evaluation. In practice, performance estimates need to be made using methods such as cross validation, which become innaccurate for biased training sets [3]. Secondly, the work that addresses this by constructing modified estimators for generalization performance compare the mean estimated performance with the mean generalization performance [3]. This involves averaging performance over many experiments which is possible since the human expert (Oracle) is simulated. However, in practice, the accuracy of the mean estimate is not useful because practitioners only have "one shot" at estimating generalization performance due to the human labeling process being expensive.

Measuring the performance of active learning for practitioners is an understudied problem with the main method being a weighting scheme which has been discovered under different settings [3]–[6]. This method requires an unbiased distribution $p(x)$ and a biased distribution $q(x)$ over the unlabeled instances and, in the cross validation estimate, weights the error for the i^{th} instance as $\frac{p(x_i)}{q(x_i)}$. Though these methods eliminate the bias of the generalization performance estimate they have three significant limitations for practitioners. Firstly, while the estimator removes bias, we have empirically found these estimates to be high variance. In academic settings where hundreds or thousands of experimental results are averaged this does not show up as an issue. However, in *practice* this is a significant issue since the practitioner only performs the active learning "experiment" once. Rather than reporting the average of many estimates we report the average `absolute error` (i.e. abs(PerformanceEstimate - TruePerformance)) and show that the absolute error of these methods can be very high. Secondly, in the context of classification, this weighting scheme can yield nonsensical results such as accuracies greater than 1. Finally, methods that apply this weighting scheme to active learning use algorithm specific sampling distributions which do not generalize to other active learning methods [4], [5].

We propose a novel multi-step approach to calculating accurate estimates for active learning. The method can be used with many active learning query schemes and many possible learners since it first constructs a probability distribution over the instances based on an active query

1550-4786/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDM.2015.137

scoring strategy and samples instances to query according to this distribution. We then apply this to a new weighted cross validation estimator which adjusts for not choosing instances uniformly at random. The two are intertwined and are guaranteed to give an estimate between 0 and 1.

Our contributions are:

- We illustrate three important, previously unaddressed issues with the previously proposed weighted cross validation estimator (CVW) [3]–[6].
- We empirically show CVW can be very inaccurate for SVMs, naive Bayes and logistic regression (see Figures 2-6 cyan lines).
- We present a new weighted cross validation estimator that produces more accurate estimate **for each trial** for SVMs, naive Bayes and logistic regression (see Figures 2-6).
- We show the benefits of our method when used in the active learning setting by illustrating that for more conservative termination criteria the absolute error in the estimate typically gets smaller (see Table IIa).
- We show that our method achieves similar performance on a hold out test set *while* obtaining more accurate estimates of error c.f. Table III.

We begin our paper by describing related work next and then describe previous key results in section III. We then describe our method in section IV and our experimental results for three learning algorithms (SVM, logistic regression and naive Bayes) in section V. Finally we discuss our results and conclude.

II. Related Work

The core contribution of our work is a more accurate method for estimating learner performance in the active learning setting. Three areas have touched upon this area and we briefly survey them along with how our work differs.

Weighted Cross Validation. Cross validation methods estimate the expected generalization error of a learning algorithm on a training set by splitting the labeled training data into many train-test splits, repeatedly training the learner on the train split and testing on the test split [7]. The performance is then averaged across the test splits and used as an estimate for the generalization error. An important assumption in cross validation is that the labeled training data has been sampled according to the true distribution of the data [3].

Because active learning methods generate a biased training set, cross validation estimates may be inaccurate. Many authors have noted this phenomenon in the settings of active learning, coviarate shift and sample selection bias [3]–[6], [8]. Most of these works suggest using a weighted cross validation estimator $\hat{\theta}_{CVW}$ (described in section III) [3]–[6]. As explained in section III, it is easy to prove (assuming minor regularity conditions) this estimator is "unbiased" in the sense that $E[\hat{\theta}] = E[\hat{\theta}_{CVW}]$ where $\hat{\theta}$ is the standard unweighted cross validation estimator and the expectation is with respect to the given labeled training set. While these works *suggest* the use of $\hat{\theta}_{CVW}$, only [3] experiments with the accuracy of this estimator, and in their experiments they only compare the **mean** cross validation estimate to the **mean** test error. That is, rather than measure the accuracy of the estimates for each experiment, they compute the accuracy of the mean estimate. Because the estimator corrects for the sampling bias, it's expected that the *mean* predicted and actual errors would be similar. However, in practice practitioners only run an experiment once, so what is important to them is the error between the predicted and actual errors over a *single experiment*. In section V we show $\hat{\theta}_{CVW}$ produces very inaccurate estimates when you compare them to the actual performance *on a single experiment*.

Unbiased Validation Sets. A related line of work is [9], but instead of weighting cross validation estimates they actively sample a biased training set and an unbiased validation set. For cross validation they only use the losses on the unbiased validation set. While their work focuses on model selection, ours focuses on the accuracy of generalization error estimates. They actively construct two data sets while our method uses queried data for both training and validation which critically minimizes the effort of the human Oracle. We see two additional issues with this method:

- They use potentially inaccurate pseudolabels based on previously trained models for selecting the best candidate point to query. Furthermore, we are aware of no theoretical justification for the use of pseodolabels in approximating the posteriors of unlabeled instances.
- They use an expensive and complicated leave-2-out cross validation procedure to compute the expected improvement in performance.

Our method doesn't rely on pseudolabels, can be much faster because it can be used with any active scoring method and is simple to implement.

Active Learning of the Validation Set. Finally, [10] takes a different approach to generating more accurate generalization performance estimates by fixing the model generated by the learning algorithm and actively querying unlabeled instances to generate a validation set used *only* for estimating generalization performance. While there may be settings where this approach is valuable (e.g. the original labeled training data is unavailable and the user is only interested in the generalization performance of the existing model), the much more common setting is the labeled training data is available and the user wants to acquire more labeled training data to improve generalization performance. Estimating generalization performance using cross validation makes much more sense in this setting because labeled data can be used both for the training and assessing generalization performance.

III. A Short Primer on Key Results

In this section we review past results that show regular cross validation leads to inaccurate results when the train-

TABLE I. VARIABLES AND NOTATION USED WITHIN THE PAPER.

D	labeled training set $\{(x_1, y_1), ..., (x_n, y_n)\}$ where each (x_i, y_i) is drawn from a different distribution $q_i(x)$ and the draws are conditionally independent given the learner trained at iteration $i-1$.
$U = \{x_1, ..., x_u\}$	Unlabeled training data
u	The size of the unlabeled training data set. For ease of exposition we assume this remains constant. Additionally, we assume that $u > n$ (i.e. we have more unlabeled data than labeled data).
$L(f_D(x), y)$	loss on (x, y) with function f_D trained on data set D.
$D - i$	Labeled training data without instance i.
$L(f_{D-i}(x_i), y_i)$	loss on instance i when training without instance i
L_i	shorthand notation for $L(f_{D-i}(x_i), y_i)$
$p_i(x)$	Unbiased distribution over unlabeled instances at iteration i
$q_i(x)$	Biased distribution over unlabeled instances at iteration i
$\hat{\theta}$	Standard, unweighted cross validation estimator.
$\hat{\theta}_{CVW}$	Previously proposed weighted cross validation estimator.
$\hat{\theta}_{CVN}$	Our proposed weighted cross validation estimator.

ing data is selected with bias. We review the weighted cross validation estimator $\hat{\theta}_{CVW}$ and discuss why it corrects for sampling distribution bias. Finally, we expose three issues with $\hat{\theta}_{CVW}$ which lead to inaccurate estimates and motivate our new estimator. Table I summarizes variables and notation used within the remainder of the paper.

For ease of exposition we focus on leave-one-out cross validation (LOOCV) [7], but the results and issues extend to other forms of cross validation such as k-fold and repeated cross validation [7].

A. Cross Validation Bias in Active Learning

LOOCV is a popular estimator for generalization error but becomes more biased if the training data are not sampled according to the true distribution of the data. This bias in the cross validation estimate can be eliminated by scaling error/accuracy calculations by a factor of $\frac{p_i(x_i)}{q_i(x_i)}$, where $p_i(x_i)$ and $q_i(x_i)$ are the unbiased and biased probability densities for the instance queried at iteration i. This leads to the following weighted leave-one-out cross validation estimator

$$\hat{\theta}_{CVW} = \frac{1}{n} \sum_{(x_i, y_i) \in D} L_i \frac{p_i(x_i)}{q_i(x_i)} \tag{1}$$

This estimator has been proposed by many such as [3]–[6] and the proof that this estimator eliminates the bias due to sampling bias can be found in these papers.

Without any information of the true distribution of the data, $p_i(x_i)$ can be approximated by $\frac{1}{u}$ where u is the size of the set of unlabeled data.

B. Issues with Weighted Cross Validation

An issue that has not been investigated is the variance of $\hat{\theta}_{CVW}$. An analysis of this is challenging without making restrictive assumptions of the true generalization error, the learning algorithm or the loss function. Instead, we present three issues with $\hat{\theta}_{CVW}$ that can make it an inaccurate estimator.

Weighted cross validation attempts to correct for sampling bias by weighting the loss of individual instances. Part of the reason the weighting scheme reduces the bias is because $E_{x \backsim q_i}[\frac{p_i(x)}{n q_i(x)}] = \frac{1}{n}$ where $\frac{1}{n}$ is the weight used in regular cross validation. Focusing only on this term, we can show it can be very high variance when q_i is very distant from the uniform distribution.

Let p_i be the discrete uniform distribution over the u instances in the unlabeled data set. It is straightforward to show the following lower bound on the variance of $\frac{p_i(x)}{q_i(x)}$ with respect to q_i:

$$\mathrm{Var}[\frac{p_i(x)}{q_i(x)}] \geq V_2(p_i||q_i) - \frac{2}{u} \tag{2}$$

Where V_2 is the Rényi Divergence of order α [11]. Thus, when q_i is far from the uniform distribution the variance of this term will be large.

The second and third issues with $\hat{\theta}_{CVW}$ are due to the fact that the terms $\frac{p_i(x_i)}{n q_i(x_i)}$ do not necessarily sum to 1. There are two consequences of this:

1) Even if every prediction in computing $\hat{\theta}_{CVW}$ is correct, it's possible have an estimated accuracy less than 1.
2) It is possible to have an estimated accuracy greater than 1.

If high density instances are selected, then (1) may occur. For example, suppose each of the n labeled instance were selected with probability $\frac{u-1}{u}$ (for simplicity we assume u doesn't change between active iterations, but the issue remains if it does). If every instance is correctly predicted when computing $\hat{\theta}_{CVW}$'s estimate then the best accuracy attainable is $\frac{nu}{nu(u-1)} = \frac{1}{u-1} < 1$. This problem is likely to occur in practice if the sampling distribution is low entropy because it is due to instances with high probability density being sampled.

Alternatively, if low density instances are sampled then (2) may occur. Suppose each of the n sampled labeled instances were sampled with probability density $\frac{1}{u^2}$. If every instance is correctly predicted then the best accuracy attainable is $\frac{nu^2}{nu} = u > 1$. This phenomenon can occur even if only a *single* instance with low probability density is selected. For example, suppose we have n labeled instances, one of which was sampled with probability density $\frac{1}{u^2}$. Even if every other instance is labeled incorrectly during cross validation, it is still possible to have an estimated accuracy of $\frac{u^2}{un} = \frac{u}{n} > 1$. Thus, while this behavior requires sampling an instance with low probability density, even a single sample can lead to nonsensical accuracy estimates.

All of these problems can cause $\hat{\theta}_{CVW}$ to make inaccurate estimates of generalization performance. In the

Input:
D: Labeled data
U: Unlabeled data
$\mathcal{F}$: Learning algorithm
$\mathcal{E}$: Active learning query method
Output:
f: Trained model
$\hat{\theta}_{CVN}$: Generalization error estimate

1: Let q be the list of probability densities of the instances in D.
2: $q(x_j) \leftarrow \frac{1}{|U|}$ for all $x_j \in D$
3: $f \leftarrow$ Trained model using D and $\mathcal{F}$
4: **repeat**
5: $S \leftarrow$ Scores generated by $\mathcal{E}$ using f and U
6: Normalize S to be a probability distribution
7: $x_i \leftarrow$ Sample S
8: $q(x_i) \leftarrow S(x_i)$
9: Query label y_i of x_i
10: $D \leftarrow D \cup \{x_i, y_i\}$
11: $U \leftarrow U - \{x_i\}$
12: $\hat{\theta}_{CVN} \leftarrow$ Estimate generalization performance using equation 5
13: $f \leftarrow$ Trained model using D and $\mathcal{F}$
14: **until** Termination criterion met

Fig. 1. Our Method

next section we propose a new weighted cross validation estimator that addresses these issues.

IV. Our Method

Our active learning method is summarized in algorithm IV, but the 4 main steps are:

1) Generate a sampling distribution over the unlabeled instances using the previously trained model.
2) Sample unlabeled instances for labeling according to the distribution.
3) Retrain the model with the new labeled data.
4) Use the probability densities of the sampled instances for our new weighted cross validation estimator to estimate the generalization performance.

This process is repeated until a termination criterion is met. The key difference between our method and previous work is how we perform steps 1 and 4. In the previous section we noted several issues with $\hat{\theta}_{CVW}$ so we first describe a new weighted cross validation estimator that addresses these issues. Next, we describe how we construct the sampling distribution in step 1. A limitation of previous work in this area is that the sampling distributions are constructed in an algorithm specific fashion [3]–[6]. Our method is more general because it allows for the construction of sampling distributions using many different active learning query schemes as a base.

It is important to note that while many active learning query schemes make queries deterministically [1], our weighted cross validation estimator requires the construction of a sampling distribution and for queries to be made on instances sampled from this distribution. Also, while algorithm IV samples a single instance each time f is trained, the method can be changed to sample multiple instances each iteration by simply repeating lines 6-11 and removing $S(x_i)$ from S.

Weighting Strategy. Rather than use the traditional weighting term $\frac{p_i(x)}{q_i(x)}$ we propose the use of the following weighting scheme:

$$\frac{1}{q_i(x_i)\sum_{(x_i,y_i)\in D}\frac{1}{q_j(x_j)}} = \frac{\frac{1}{q_i(x_i)}}{\sum_{(x_i,y_i)\in D}\frac{1}{q_j(x_j)}} \quad (3)$$

Where we let $p_i(x)$ be the uniform distribution over the labeled data. While both terms give higher weights to sampled instances with lower probability densities, our term has the useful property that the weights always sum to 1:

$$\sum_{(x_i,y_i)\in D}\frac{\frac{1}{q_i(x_i)}}{\sum_{(x_j,y_j)\in D}\frac{1}{q_j(x_j)}} = \frac{\sum_{(x_i,y_i)\in D}\frac{1}{q_i(x_i)}}{\sum_{(x_j,y_j)\in D}\frac{1}{q_i(x_i)}} = 1 \quad (4)$$

This is an important difference because the weighting terms used by $\hat{\theta}_{CVW}$ which only sum to 1 in *expectation*. The effect is that using this weighting scheme will cause estimates to always be in the range $[0, 1]$.

Thus, we propose the following normalized weighted cross validation estimator (CVN):

$$\hat{\theta}_{CVN} = \sum_{(x_i,y_i)\in D} L_i \frac{1}{q_i(x_i)\sum_{(x_j,y_j)\in D}\frac{1}{q_j(x_j)}} \quad (5)$$

As with $\hat{\theta}_{CVW}$, it doesn't seem possible to derive a closed form expression for the variance of $\hat{\theta}_{CVN}$ without making unrealistic assumptions. We leave these issues for future work.

Sampling Distribution Construction. Our method assumes the existence of a sampling distribution over the unlabeled data. In most work on weighted cross validation for active learning the sampling distribution is constructed in an algorithm specific manner such as in [4] and [5]. We propose a more general approach. Most active learning querying strategies assign some sort of score to each unlabeled instances. This is true for popular querying strategies such as entropy sampling, query by committee and expected model change [1]. Thus, we propose taking these scores and normalizing them to produce a probability distribution.

V. Experiments

We ask the following questions:

1) How accurate are the generalization error estimates made by our method relative to other

methods for logistic regression, naive Bayes and SVM classifiers (see Figures 2-6)?
2) How accurate are these generalization error estimates when incorporating a termination criterion (Table II)?
3) Does our method produce models comparable in generalization accuracy to regular cross validation using standard, deterministic active learning query strategies (Table III)?

While there are many active learning query schemes [1], we chose to focus most of our experiments on entropy sampling (for logistic regression and naive Bayes) and distance from the hyperplane (for SVM) due to their popularity, generality and simplicity. These are the query strategies used in our experiments unless otherwise stated. For these methods, queries are made with respect to the model chosen in the previous iteration by cross validation. However, to show that our method works well for other query strategies as well we included experiments using the modified query by committee (QBC) proposed in [9] (Figures 5 and 6). As in [9], the committee is all the models trained in the previous iteration (using the different regularization parameters) and the score used to construct the sampling distribution is 1 minus the fraction of models which disagree with the majority predicted label on the instance.

For all experiments we started with two labeled instances for each class and ran 20 iterations of active learning, querying 5 instances each iteration. Experiments were repeated 30 times using different train-test splits. In all plots error bars represent the binomial proportion confidence interval at 95% confidence.

Tables II and III have the cross validation estimate error and generalization performance when incorporating an active learning termination criterion. We experimented with a termination criteria which stops active learning if the cross validation estimate of generalization error hasn't improve for k iterations (where k is a hyperparameter). Specifically, after n iterations of active learning, we compare the estimated error at iteration $n-k$ with the estimates at iterations $n-k+1, n-k+2, ..., n$, and active learning is terminated if the former is larger than all the subsequent estimates. The final column in these tables is for $k = 20$, indicating all 20 iterations of active learning were allowed to run.

Matlab code for our method is available at https://github.com/adgress/ICDM2015.

Summary of Results. As mentioned in the introduction practitioners require an accurate estimate of learner performance for two reasons: i) to determine if the learner accuracy is sufficient to deploy the model and ii) to determine if more rounds of active learning are necessary. question 1 addresses the former and question 2 the later.

To answer question 1 we plotted learning curves for increasing amounts of labeled data (which is actively chosen) and as we can see in Figures 2-6 our method produces the most accurate estimates. It is important to realize that these plots do not plot generalization error. Rather, they plot the absolute error of the cross validation estimates. The takeaway from these experiments is that our method can accurately estimate the generalization performance of the learner and hence help the practitioner make a more informed decision on whether or not to deploy the model.

Question 2 asks how useful our method is when paired with an active learning termination criterion. Table II shows the accuracies of $\hat{\theta}_{CVW}$ and $\hat{\theta}_{CVN}$ for decreasingly stringent termination criterions (columns represent termination thresholds of different numbers of iterations of non decreasing error estimates). Our method makes much more accurate estimates, which is a direct result of the accuracy of our estimator as shown in Figures 2-6.

Finally question 3 addresses the important issue of does *sampling* the instance to query rather than always choosing the most informative diminish the overall performance of the learner. Here we are measuring generalization performance of the learner, unlike the previous Figures where we measure the error of the estimates. Table III shows our method produces models with comparable generalization performance compared to choosing the most informative query and using regular cross validation. A valid question is then **why do we need our method if it produces models with similar generalization performance?** The reason is that though the performance is similar the performance estimates our method produces are much more accuracy, as shown by Figures 2-6.

Data Set Description. We ran experiments on four data sets:

- USPS digits (digits 3 vs 8 and 1 vs 7): A set of aligned and centered grayscale images of handwritten digits. Images are 16×16. Provided by [12].
- Boston Housing: A housing values data set with 13 features with the house value as the output. We converted this to a classification problem by assigning labels based on whether the response is greater than or less than the median response. Provided by the Probabilistic Modeling Toolkit, a Matlab package prepared primarility for [13].
- UCI Yeast (2 most common classes): A protein prediction data set of 8 features. Provided by the UCI Machine Learning Repository [14].
- 20 newsgroups (20NG) (four pairs of computer and recreation subgroups): A text data set of 20000 message take from 20 news groups. For our experiments we created binary classification problems by predicted membership within a group. Specifically, we created the four data sets: CR1 (comp.graphics vs rec.autos), CR2 (comp.os.ms-windows.misc vs rec.motorcycles), CR3 (comp.sys.ibm.pc.hardware vs rec.sports.baseball) and CR4 (comp.sys.mac.hardware vs rec.sports.hockey). Features are word counts. Data was downloaded from http://qwone.com/~jason/20Newsgroups/ which was preprocessed using Rainbow [15] to select features. To accelerate experiments we randomly

selected a subset of 500 messages from each of the relevant groups.

For all but the 20NG data set we standardized the features to have zero mean and unit variance before applying learning algorithms.

Cross Validation Estimators and Active Query Schemes. We compare 3 methods:

- Regular Cross Validation: standard deterministic active learning queries with the standard, unweighted cross validation estimator $\hat{\theta}$.
- Our Method: our method as described in algorithm IV.
- Weighted Cross Validation: same as our method but using $\hat{\theta}_{CVW}$ to estimate generalization performance instead of $\hat{\theta}_{CVN}$.

We did not include termination criterion tables for QBC due to space constraints. We generated them and the results were comparable to the termination criterion experiments using entropy and distance from hyperplane queries.

Classifers Used and Experimental Parameters. We apply these 3 methods to 3 differing learning algorithms:

- ℓ_2 Regularized Logistic Regression (LR) [7] using LIBLINEAR [16].
- ℓ_2 Regularized Support Vector Machine (SVM) [7] with a linear kernel using LIBLINEAR [16].
- Naive Bayes with a Gaussian prior over the features (NB) [7]. We used the implementation provided by Matlab.

For logistic regression and SVM regularization parameters were chosen by 5 times repeated cross validation from the set $\{10^c | c = -5, -4, ..., 4, 5\}$.

For the 20NG data, using naive Bayes, we made the features binary by converting the word counts to 1 if the word is present and 0 otherwise.

A. Discussion of Experiments

Question 1. Figures 2-6 plot the absolute error of the cross validation estimates for regular cross validation (red), our method (green) and weighted cross validation (cyan). On all data sets our method gives better estimates of the generalization error than weighted cross validation and generally give better or comparable estimates to regular cross validation.

Of note is that in some experiments the mean absolute error of $\hat{\theta}_{CVW}$ is greater than 1 (e.g. USPS 3 vs 8 in Figure 2). This is due to the issue noted in section III wherein $\hat{\theta}_{CVW}$ can predict accuracies greater than 1.

Also of note is $\hat{\theta}_{CVW}$ performs particularly poorly for naive Bayes with entropy sampling and for logistic regression and SVM with query by committee. In our experiments naive Bayes led to lower entropy sampling distributions than SVM and logistic regression, so we believe this phenomenon is also directly related to the issues outlined in section III. Similarly, query by committee may lead to lower entropy sampling distributions if the models in the committee all agree on the label of many instances. Furthermore, in many experiments $\hat{\theta}_{CVW}$ performs worse with more labeled data. It is possible that as more labeled data is used, the trained models become more confident in the unlabeled data, leading to lower entropy sampling distributions.

In many experiments the error of regular cross validation estimates increases with more labeled data. This is likely due to the active learning sampling bias exacerbated by the fact that these query strategies focus on instances that are "difficult" for the model.

In some experiments our method performs comparably to regular cross validation, including the phenomenon of becoming more inaccurate with more labeled instances (e.g. Figure 3, Housing). We suspect this is due to the sampling distributions being "too uniform," causing our method is devolve into regular cross validation.

Question 2. In active learning users generally pick a criterion to stop actively acquiring new labels. Figure II shows the error in the cross validation estimates after terminating active learning. These tables show our method provides much more accurate generalization error estimates than weighted cross validation upon termination.

Question 3. Figure III shows the mean generalization accuracies of our method and regular cross validation using the same termination criterion. The results are generally quite comparable. This is valuable because it suggests our method produces significantly more accurate generalization accuracy estimates without compromising generalization performance.

Our method consists of two parts: construction and use of a sampling distribution and a new weighted cross validation estimator. A natural question is how well does our method perform if we instead use the regular cross validation estimator (while still using a sampling distribution). We ran experiments comparing this with our method but did not include them due to space constraints. In these experiments our method produced significantly more accurate error estimates with comparable generalization performance.

VI. Conclusion

Active learning is used extensively in domains where labeling is expensive and challenging. The standard iterative active learning approach starts with few labeled data points and many unlabeled points, builds a model, then suggests unlabeled points that should be labeled by an Oracle/human. The process is repeated until the accuracy of the trained model converges. Though query schemes have received much attention in research communities there is little work on estimating the accuracy of classifiers in the active learning setting. Practitioners require accurate generalization performance estimates of actively

trained classifiers for two reasons: i) to determine if the accuracy of the learner has converged and ii) to determine if its worth deploying. However, traditional cross validation methods are not well suited to this setting due to sampling bias. Furthermore, experimental results of the existing weighted cross validation method are limited due to pasts experiments comparing mean estimates with the mean test error. This setting is unrealistic because practitioners only perform one experiment and we show that this method perform poorly if we measure the accuracy of the estimates for each experiment individually. We explained that this poor performance is due to the unusual properties of this estimator such as the possibility of predicting accuracies greater than 1. Additionally, this method requires a probability distribution over the unlabeled instances but past work only suggests algorithm specific methods for constructing these distributions.

We introduced a flexible new method that constructs a probability distribution over the unlabeled instances using a base active learning query method, samples instances and uses the probability density of these instances in a new cross validation estimator. We showed our method produces much more accurate generalization error estimates than existing methods and when used in an active learning scheme produces significantly more accurate estimates while maintaining comparable generalization performance.

Two important open questions for this method are (i) how to best construct the sampling distribution and (ii) does this method have provably lower variance than weighted cross validation. We leave these questions for future work.

Acknowledgment

The authors gratefully acknowledge support of this research from ONR grant N00014-11-1-010. We thank the reviewers for their helpful suggestions including the use of Rényi Divergences.

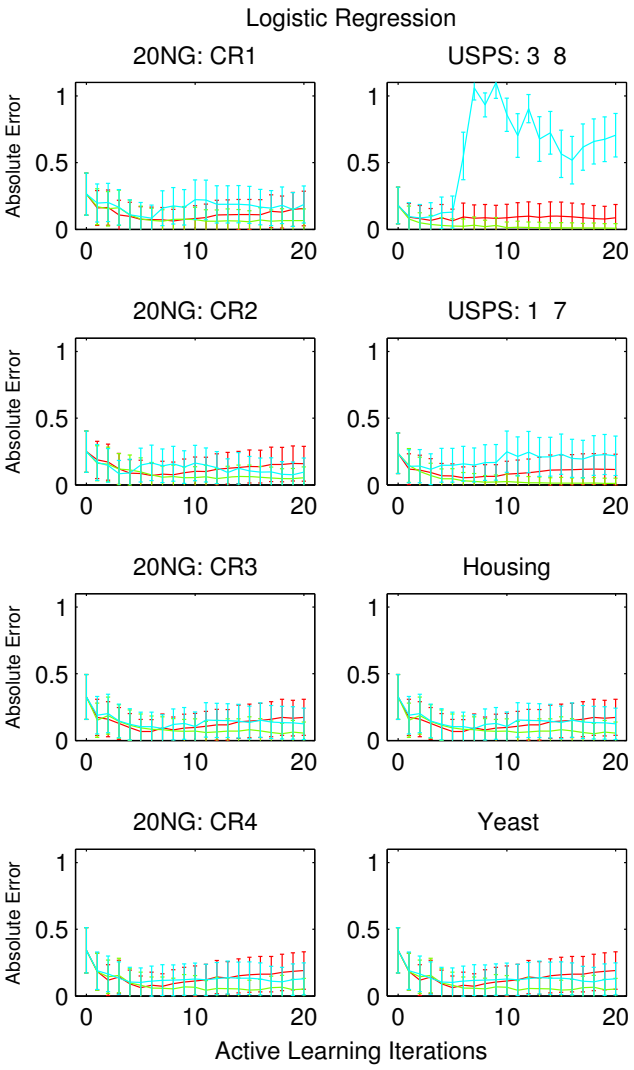

Fig. 2. Cross validation estimate error of all 3 cross validation methods using logistic regression. Plotted methods are regular cross validation (red), our method (green) and weighted cross validation (cyan).

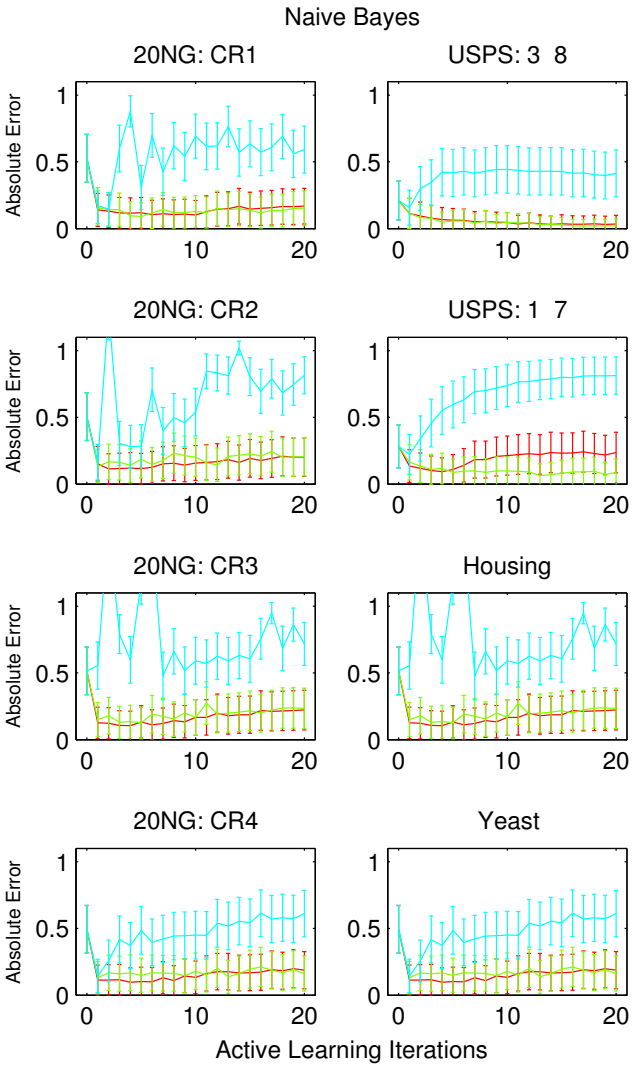

Fig. 3. Cross validation estimate error of all 3 cross validation methods using naive Bayes. Plotted methods are regular cross validation (red), our method (green) and weighted cross validation (cyan).

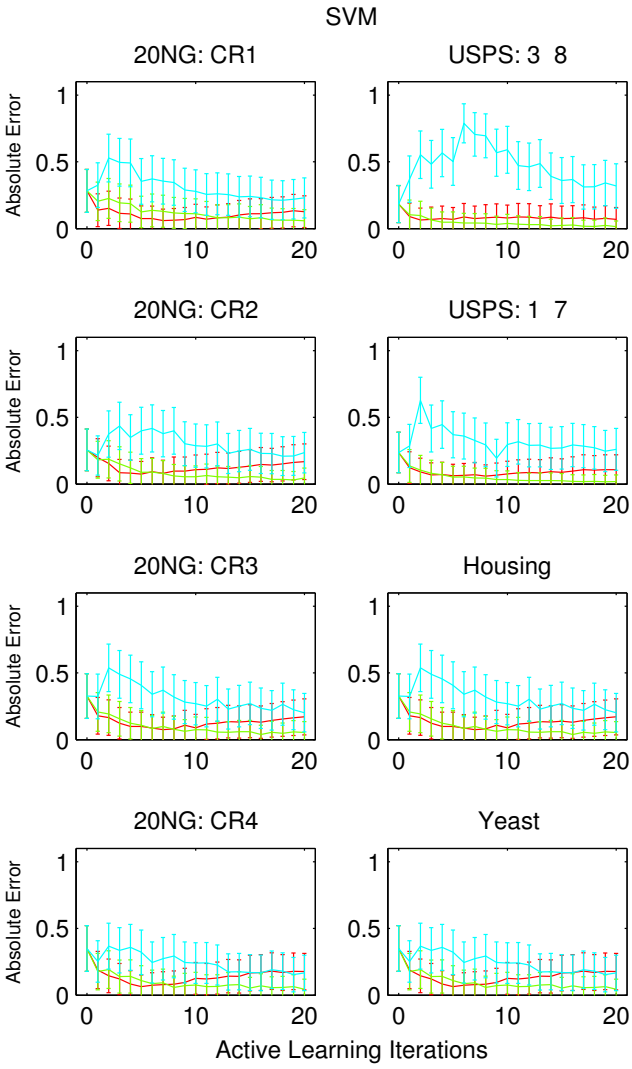

Fig. 4. Cross validation estimate error of all 3 cross validation methods using SVM. Plotted methods are regular cross validation (red), our method (green) and weighted cross validation (cyan).

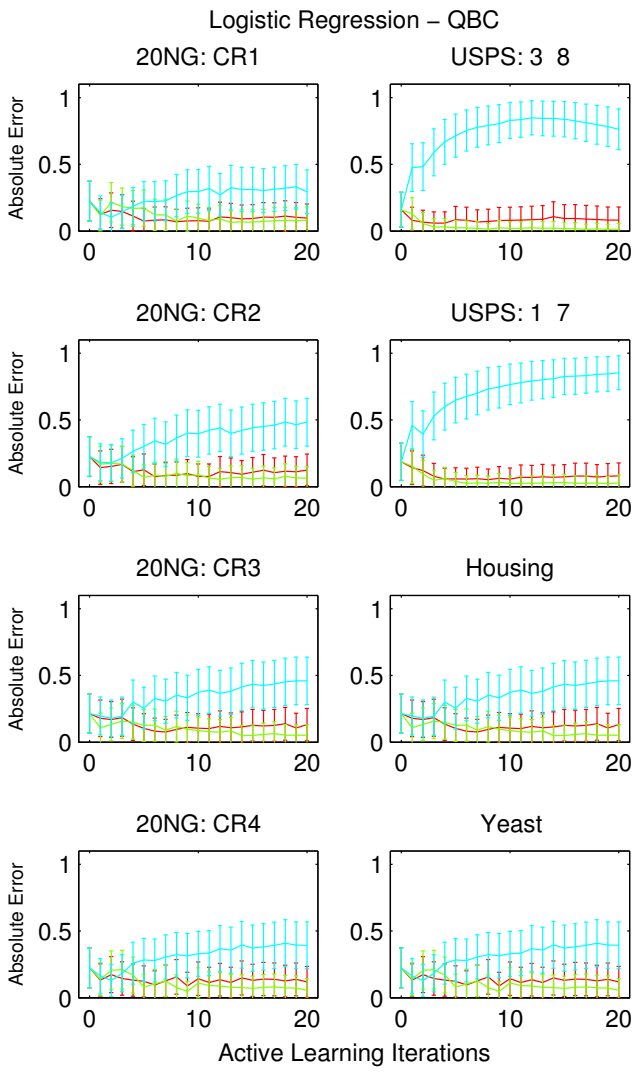

Fig. 5. Cross validation estimate error of all 3 cross validation methods using logistic regression and query by committee queries. Plotted methods are regular cross validation (red), our method (green) and weighted cross validation (cyan).

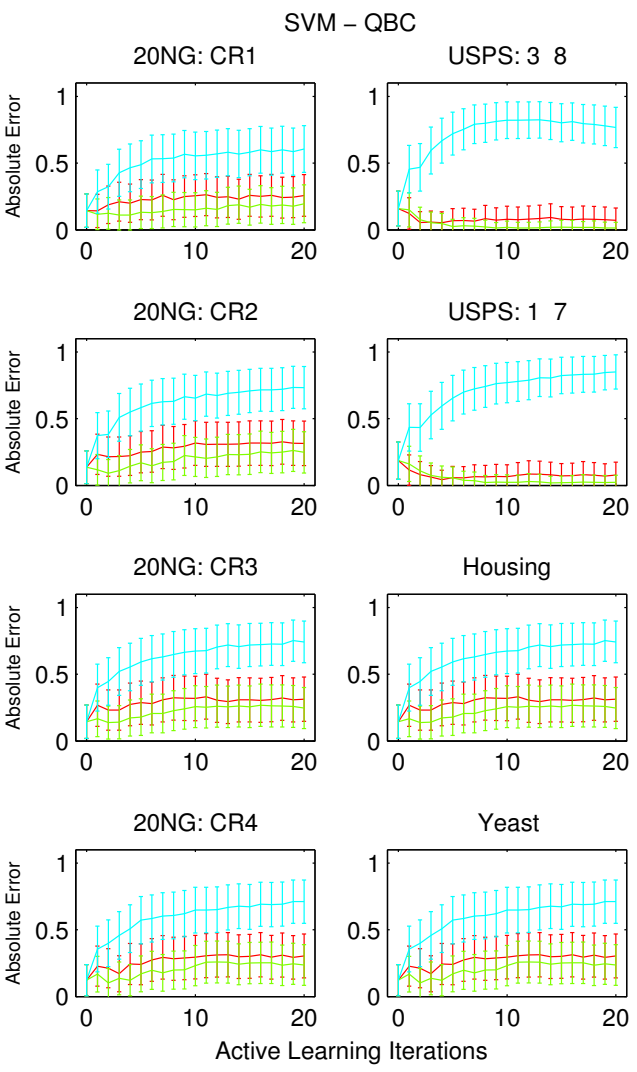

Fig. 6. Cross validation estimate error of all 3 cross validation methods using SVM and query by committee queries. Plotted methods are regular cross validation (red), our method (green) and weighted cross validation (cyan).

TABLE II. Error in accuracy estimates of our method and weighted cross validation. Columns correspond to the number of subsequent cross validation estimates to consider for our termination criteria (explained in section V).

(a) Our Method

	1	3	5	7	20
LR 20NG: CR1	0.14	0.093	0.066	0.055	0.065
LR 20NG: CR2	0.11	0.088	0.062	0.052	0.054
LR 20NG: CR3	0.11	0.12	0.094	0.077	0.055
LR 20NG: CR4	0.15	0.082	0.074	0.053	0.052
LR USPS: 3 8	0.051	0.029	0.018	0.027	0.009
LR USPS: 1 7	0.11	0.068	0.032	0.025	0.012
LR Housing	0.11	0.068	0.032	0.025	0.012
LR Yeast	0.11	0.068	0.032	0.025	0.012
SVM 20NG: CR1	0.17	0.14	0.13	0.096	0.058
SVM 20NG: CR2	0.16	0.081	0.053	0.069	0.045
SVM 20NG: CR3	0.18	0.095	0.083	0.085	0.054
SVM 20NG: CR4	0.14	0.1	0.091	0.092	0.045
SVM USPS: 3 8	0.096	0.061	0.056	0.036	0.014
SVM USPS: 1 7	0.1	0.077	0.054	0.039	0.019
SVM Housing	0.1	0.077	0.054	0.039	0.019
SVM Yeast	0.1	0.077	0.054	0.039	0.019
NB 20NG: CR1	0.11	0.11	0.13	0.14	0.15
NB 20NG: CR2	0.15	0.19	0.19	0.22	0.21
NB 20NG: CR3	0.18	0.22	0.24	0.2	0.23
NB 20NG: CR4	0.14	0.16	0.19	0.19	0.16
NB USPS: 3 8	0.079	0.047	0.048	0.039	0.021
NB USPS: 1 7	0.18	0.11	0.079	0.088	0.086
NB Housing	0.18	0.11	0.079	0.088	0.086
NB Yeast	0.18	0.11	0.079	0.088	0.086

(b) Weighted Cross Validation

	1	3	5	7	20
LR 20NG: CR1	0.14	0.097	0.096	0.14	0.19
LR 20NG: CR2	0.13	0.12	0.15	0.11	0.096
LR 20NG: CR3	0.13	0.083	0.096	0.1	0.12
LR 20NG: CR4	0.15	0.097	0.1	0.13	0.13
LR USPS: 3 8	0.084	0.11	0.63	0.68	0.71
LR USPS: 1 7	0.14	0.11	0.13	0.17	0.22
LR Housing	0.14	0.11	0.13	0.17	0.22
LR Yeast	0.14	0.11	0.13	0.17	0.22
SVM 20NG: CR1	0.34	0.32	0.28	0.25	0.23
SVM 20NG: CR2	0.24	0.24	0.33	0.28	0.24
SVM 20NG: CR3	0.37	0.38	0.29	0.23	0.2
SVM 20NG: CR4	0.3	0.21	0.2	0.22	0.17
SVM USPS: 3 8	0.37	0.4	0.32	0.51	0.32
SVM USPS: 1 7	0.37	0.34	0.22	0.27	0.26
SVM Housing	0.37	0.34	0.22	0.27	0.26
SVM Yeast	0.37	0.34	0.22	0.27	0.26
NB 20NG: CR1	0.19	0.34	0.42	0.52	0.59
NB 20NG: CR2	0.19	0.27	0.4	0.47	0.82
NB 20NG: CR3	0.74	1.2	0.44	0.52	0.72
NB 20NG: CR4	0.2	0.35	0.35	0.46	0.61
NB USPS: 3 8	0.27	0.43	0.46	0.41	0.41
NB USPS: 1 7	0.37	0.53	0.64	0.7	0.81
NB Housing	0.37	0.53	0.64	0.7	0.81
NB Yeast	0.37	0.53	0.64	0.7	0.81

TABLE III. Difference of test set accuracies after termination of our method and regular cross validation. Columns correspond to the number of subsequent cross validation estimates to consider for our termination criteria (explained in section V). Positive values indicate our method performed better on the test set. The important takeaway is that our method's performance is comparable to regular cross validation.

	1	3	5	7	20
LR 20NG: CR1	0.012	-0.043	-0.009	-0.039	-0.039
LR 20NG: CR2	-0.009	-0.01	-0.006	-0.009	-0.004
LR 20NG: CR3	-0.046	-0.016	-0.003	0.008	-0.018
LR 20NG: CR4	0.009	-0.029	-0.036	-0.023	-0.035
LR USPS: 3 8	0.033	0.014	0.01	0	0.011
LR USPS: 1 7	-0.013	0.017	0.03	0.002	-0.008
LR Housing	-0.013	0.017	0.03	0.002	-0.008
LR Yeast	-0.013	0.017	0.03	0.002	-0.008
SVM 20NG: CR1	-0.002	-0.037	-0.024	-0.011	-0.009
SVM 20NG: CR2	-0.014	-0.007	-0.006	0.03	-0.01
SVM 20NG: CR3	-0.043	0.026	0.022	-0.001	-0.011
SVM 20NG: CR4	0.051	0.024	-0.001	-0.015	-0.011
SVM USPS: 3 8	-0.03	-0.008	-0.004	-0.006	-0.008
SVM USPS: 1 7	0.03	0.004	0.007	-0.001	-0.007
SVM Housing	0.03	0.004	0.007	-0.001	-0.007
SVM Yeast	0.03	0.004	0.007	-0.001	-0.007
NB 20NG: CR1	0.021	0.004	-0.004	0.013	0.007
NB 20NG: CR2	-0.004	-0.013	0.01	0.005	0.011
NB 20NG: CR3	-0.037	0	0.017	0.012	0.011
NB 20NG: CR4	0.005	-0.004	0.017	0.021	0.009
NB USPS: 3 8	0.004	0.007	0.005	0.008	0.019
NB USPS: 1 7	0.011	-0.004	0	0.003	-0.005
NB Housing	0.011	-0.004	0	0.003	-0.005
NB Yeast	0.011	-0.004	0	0.003	-0.005

References

[1] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, no. 55-66, p. 11, 2010.

[2] H. Schütze, E. Velipasaoglu, and J. O. Pedersen, "Performance thresholding in practical text classification," in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, ser. CIKM '06, New York, NY, USA, 2006.

[3] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *The Journal of Machine Learning Research*, vol. 8, pp. 985–1005, 2007.

[4] A. Beygelzimer, S. Dasgupta, and J. Langford, "Importance weighted active learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 49–56.

[5] F. R. Bach, "Active learning for misspecified generalized linear models," in *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, vol. 19. Mit Press, 2007, p. 65.

[6] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 114.

[7] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The elements of statistical learning*. Springer, 2009, vol. 2, no. 1.

[8] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *ICML*. Citeseer, 2000, pp. 839–846.

[9] A. Ali, R. Caruana, and A. Kapoor, "Active learning with model selection," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, 2014, pp. 1673–1679.

[10] C. Sawade, N. Landwehr, S. Bickel, and T. Scheffer, "Active risk estimation," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 951–958.

[11] T. van Erven and P. Harremoës, "Rényi divergence and Kullback-Leibler divergence," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 3797–3820, 2014.

[12] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1548–1560, 2011.

[13] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[14] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[15] A. K. McCallum, "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering," 1996, http://www.cs.cmu.edu/ mccallum/bow.

[16] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.