Dual Strategy Active Learning

Pinar Donmez[1], Jaime G. Carbonell[1], and Paul N. Bennett[2]

[1] School of Computer Science, Carnegie Mellon University,
5000 Forbes Ave, Pittsburgh PA, 15213 USA
{pinard, jgc}@cs.cmu.edu
[2] Microsoft Research, 1 Microsoft Way, Redmond, WA 98052 USA
Paul.N.Bennett@microsoft.com

Abstract. Active Learning methods rely on static strategies for sampling unlabeled point(s). These strategies range from uncertainty sampling and density estimation to multi-factor methods with learn-once-use-always model parameters. This paper proposes a dynamic approach, called DUAL, where the strategy selection parameters are adaptively updated based on estimated future residual error reduction after each actively sampled point. The objective of dual is to outperform static strategies over a large operating range: from very few to very many labeled points. Empirical results over six datasets demonstrate that DUAL outperforms several state-of-the-art methods on most datasets.

1 Introduction

Active learning has received significant attention in recent years, but most work focuses on presenting a new algorithm and showing how for some datasets and under some operating range it outperforms earlier methods [17,16,6]. Some active learning methods perform best when very few instances have been sampled, whereas others perform best only after substantial sampling. For instance, density estimation methods perform well with minimal labeled data since they sample from maximal-density unlabeled regions, and thus help establish the initial decision boundary where it affects the most remaining unlabeled data [8]. On the other hand, uncertainty sampling methods "fine tune" a decision boundary by sampling the regions where the classifier is least certain, regardless of the density of the unlabeled data [2,4]. Such methods work best when a larger number of unlabeled points may be sampled, as we show later in this paper. This paper takes a step towards a principled ensemble-based sampling approach for active learning that dominates either method individually, largely by selecting sampling methods based on estimated residual classification error reduction. Different active learning methods use different selection criteria. For example, *Query-by-Committee* [1,2] selects examples that cause maximum disagreement amongst an ensemble of hypotheses. Hence, it reduces the version space [5] and is similar to Tong and Koller's approach [4] which halves the version space in an SVM setting. *Uncertainty sampling* [3] selects the example on which the learner has lowest classification certainty. Version-space reduction methods eliminate areas of the parameter space that have no direct effect on the error rate, but may have indirect effects. Uncertainty sampling is not immune to selecting outliers since they have high uncertainty [6], but the underlying data distribution is ignored. Several active learning strategies including [9,8,6] propose ways to trade-off uncertainty vs. data density. Xu et al. [8] propose a representative sampling method which uses the k-means algorithm to cluster the data within the margin of an SVM classifier and selects the cluster centroids for labeling. McCallum and Nigam [6] also suggest a clustering based approach using the EM algorithm. All these methods aim to balance the uncertainty of the sample with its representativeness, but do so in a fixed manner, rather than by dynamically selecting or reweighting, based on residual error estimation. In this paper, we introduce a **Du**al strategy for **A**ctive **L**earning, DUAL, which is a context-sensitive sampling method. DUAL significantly improves upon the work of [9] by incorporating a robust combination of density weighted uncertainty sampling and standard (uniform) uncertainty sampling. The primary focus of DUAL is to improve active learning for the later portion of the process, rather than traditional methods that concentrate primarily on the initial dataset labeling. Baram et al. [10] present an ensemble active learning method that is complementary to ours, but does not subsume our mid-course strategy-switching method. Baram et al. develop an online algorithm that selects among three alternative active sampling strategies using a variant of the multi-armed bandit algorithm [11] to decide the strategy to be used at each iteration. They focus primarily on selecting which sampling method is optimal for a given dataset; in constrast, we focus on selecting the operating range among the sampling methods. Empirical results demonstrate that DUAL generally leads to better performance. Furthermore, it is also empirically shown that 1) DUAL is reliably better than the best of the single strategies, and 2) it is better across various domains and for both minimal and copious labeled data volumes.

The paper is organized as follows: Section 2 presents further motivation. Section 3 summarizes the method of [9]. Section 4 describes our new DUAL algorithm and presents the results of our emprical studies. In Section 5, we offer our observations and concluding remarks as well as suggestions for potential future directions.

2 Motivation for DUAL

Nguyen and Smeulders [9] suggest a probabilistic framework where clustering information is incorporated into the active sampling scheme. They argue that data points lying on the classification boundary are informative, but using information about the underlying data distribution helps to select better examples. They assume higher density samples lying close to the decision boundary are more informative. We call their method density weighted uncertainty sampling, or DWUS for short. DWUS uses the following active selection criterion:

$$s = \arg\max_{i \in I_u} E[(\hat{y}_i - y_i)^2 \mid x_i]p(x_i) \tag{1}$$

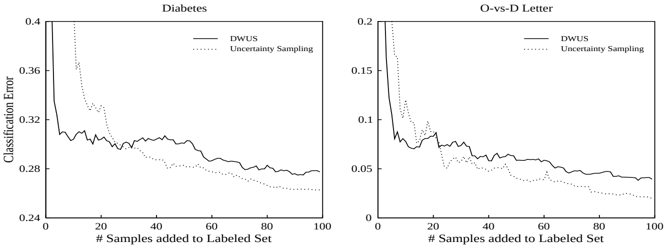


Fig. 1. Comparison of Density Weighted versus (standard) uniformly weighted Uncertainty Sampling on two UCI benchmark datasets

where $E[(\hat{y}_i - y_i)^2 \mid x_i]$ and $p(x_i)$ are the expected error and density of a given data point x_i, respectively. I_u is the index for the unlabeled data. This criterion favors points that have the largest contribution to the current classification error. In contrast, one can use an uncertainty-based selection criterion within the same probabilistic framework as illustrated by the following formula:

$$s = \arg\max_{i \in I_u} E[(\hat{y}_i - y_i)^2 \mid x_i] \tag{2}$$

We refer to the above principle as Uncertainty Sampling for the rest of this paper. Consider Fig. 1, which displays the performance of DWUS and Uncertainty Sampling on two of the datasets that we explore in more detail later. Combining uncertainty with the density of the underlying data is a good strategy to reduce the error quickly. However, after rapid initial gains, DWUS exhibits very slow additional learning while uncertainty sampling continues to exhibit more rapid improvement.[1] A similar behavior is also evident in [8] where their representative sampling method increases accuracy in the initial phase while uncertainty sampling has a slower learning rate, but gradually outperforms their method.

We investigated the Spearman's ranking correlation over candidates to be labeled by density and uncertainty in our scenario, and found that they seldom reinforce each other, but instead they tend to disagree on sample point selection. At early iterations, many points are highly uncertain. Thus, DWUS can pick high density points which are lower down in the uncertainty *ranking* but have a high absolute uncertainty score. Later, points with high absolute uncertainty are no longer in dense regions. As a result, DWUS picks points that have moderate density but low uncertainty because such points are scored highly according to the criterion in Equation 1. Hence, it wastes effort picking instances whose selection does not have a large effect on error rate reduction.

Fortunately, we can do better across the full spectrum of labeled instances by our algorithm DUAL which adopts a dynamically reweighted mixture of density and uncertainty components and achieves performance superior to its competitors over a variety of datasets. In the following section, we review essential parts of DWUS and then describe DUAL.

[1] Although a quick drop in classification error for DWUS is also observed in [9], they did not compare with uncertainty sampling.

3 Density Weighted Uncertainty Sampling (DWUS)

Nguyen and Smeulders [9] assume a clustering structure of the underlying data. $\mathbf{x}$ is the data and $y \in \{+1, 0\}$ is the class label. The cluster label $k \in \{1, 2, .., K\}$ indicates the hidden cluster information for every single data point where K is the number of total clusters. In order to calculate the posterior $P(y \mid x)$, they use the following marginalization:

$$P(y \mid x) = \sum_{k=1}^{K} P(y, k \mid x) = \sum_{k=1}^{K} P(y \mid k, x) P(k \mid x) \tag{3}$$

where $P(y \mid k, x)$ is the probability of the class label y given the cluster k and the data point x, and $P(k \mid x)$ is the probability of the cluster given the data point. But once k is known, y and x are independent since points in one cluster are assumed to share the same label as the cluster; hence knowing the cluster label k is enough to model the class label y. Thus:

$$P(y \mid x) = \sum_{k=1}^{K} P(y, k \mid x) = \sum_{k=1}^{K} P(y \mid k) P(k \mid x) \tag{4}$$

$P(k \mid x)$ is calculated only once unless the data is re-clustered, whereas $P(y \mid k)$ is updated each time a new data point is added to the training set. Before explaining how to estimate these two distributions, we illustrate below how the algorithm works:

1. Cluster the data.
2. Estimate $P(y \mid k)$.
3. Calculate $P(y \mid x)$ (Equation 4).
4. Choose an unlabeled sample based on (Equation 1) and label.
5. Re-cluster if necessary.
6. Repeat steps 2-5 until stop.

We first explain how to induce $P(k \mid x)$ according to [9]. A Gaussian mixture model is used to estimate the data density using the clustering structure such that $p(x)$ is a mixture of K Gaussians with weights $P(k)$. Hence, $p(x) = \sum_{k=1}^{K} p(x \mid k) P(k)$. where $p(x \mid k)$ is a multivariate Gaussian sharing the same variance σ^2 for all clusters k:

$$p(x \mid k) = (2\pi)^{-d/2} \sigma^{-d} \exp\{\frac{-\left|\left|x - c_k\right|\right|^2}{2\sigma^2}\} \tag{5}$$

where c_k is the centroid of the k-th cluster which is determined via the K-medoid algorithm [12]. It is similar to the K-means algorithm since they both try to minimize the squared error between the points assigned to a cluster and the cluster centroid. In K-means, the centroid is the average of all points in the cluster, whereas in K-medoid the most centrally located point in the cluster is the centroid. Moreover, K-medoid is more robust to noise or outliers.

Once the cluster representatives are identified, an EM procedure is applied to estimate the cluster prior $P(k)$ using the following two steps:

E-step: *M-step:*

$$P(k \mid x_i) = \frac{P(k)\exp\{\frac{-||x_i - c_k||^2}{2\sigma^2}\}}{\sum_{\acute{k}=1}^{K} P(\acute{k})\exp\{\frac{-||x_i - c_{\acute{k}}||^2}{2\sigma^2}\}} \qquad P(k) = \frac{1}{n}\sum_{i=1}^{n} P(k \mid x_i) \quad (6)$$

The cluster label distribution $P(y \mid k)$ is calculated using the following logistic regression model: $P(y \mid k) = \frac{1}{1+\exp(-y(c_k.\mathbf{a}+b))}$, $\mathbf{a} \in R^d$ and $b \in R$ are logistic regression parameters. c_k is the k-th cluster centroid, so $P(y \mid k)$ models the class distribution for a representative subset of the entire dataset. Points are assigned to a cluster with the probability $P(k \mid x)$ so that their labels will be affected by their cluster membership probabilities (See Equation 4). Hence, a distribution is learned at each cluster and no cluster purity requirement is forced.

The parameters of the logistic regression model are estimated via the following likelihood maximization:

$$L = \sum_{i \in I_l \cup I_u} \ln p(x_i; c_1, ..., c_K, P(1), ..., P(K)) + \sum_{i \in I_l} \ln P(y_i \mid x_i; \mathbf{a}, b) \quad (7)$$

where I_l and I_u are the indices for labeled and unlabeled data, respectively. The parameters of the first summand have already been determined by the K-medoid algorithm and the EM routine in Equation 6. The second summand is used to estimate the parameters $\mathbf{a}$ and b via Equation 4, as follows:

$$L(\mathbf{a}, b) = \frac{\lambda}{2}||\mathbf{a}||^2 - \sum_{i \in I_l} \ln \left\{ \sum_{k=1}^{K} P(k \mid x_i) P(y_i \mid k; \mathbf{a}, b) \right\} \quad (8)$$

The regularization parameter λ is given initially independently of the data. Since the problem is convex, it has a unique solution which can be solved via Newton's algorithm. Then we can calculate the probability $P(y_i \mid k; \hat{\mathbf{a}}, \hat{b})$ using the logistic regression model and obtain the class posterior probability $P(y_i \mid x_i; \hat{\mathbf{a}}, \hat{b})$ using Equation 4. The label $\hat{y}_i$ is predicted for each unlabeled point x_i according to Bayes rule. Finally, active point selection is done by Equation 1. The error expectation for a given unlabeled point $E[(\hat{y}_i - y_i)^2 \mid x_i]$ in that equation is:

$$E[(\hat{y}_i - y_i)^2 \mid x_i] = (\hat{y}_i - 1)^2 P(y_i = 1 \mid x_i) + (\hat{y}_i)^2 P(y_i = 0 \mid x_i) \quad (9)$$

Since the probability $P(y_i \mid x_i)$ is unknown, its current approximation $P(y_i \mid x_i; \hat{\mathbf{a}}, \hat{b})$ is used instead. Additionally, data points are re-clustered into smaller clusters as the expected error reduces. The reason is that it is important to make significant changes in the decision boundary during the early iterations of active sampling. Later the classification boundary becomes more stable and thus needs to be finely tuned. Additional details can be found in [9].

4 DUAL Algorithm and Experimental Results

4.1 Description of the DUAL Algorithm

DUAL works as follows: It starts executing DWUS up until it estimates a cross-over point with uncertainty sampling by predicting a low derivative of the expected error, e.g. $\frac{\partial \epsilon(DWUS)}{\partial x_t} \leq \delta$. The derivative estimation need not be exact, requiring only the detection of diminishing returns which we explain soon. Then, it switches to execute a combined strategy of density-based and uncertainty-based sampling. In practice, we do not know the future classification error of DWUS, but we can approximate it by calculating the average expected error of DWUS on the unlabeled data. It will not give us the exact cross-over point, but it will provide a rough estimate of when we should consider switching between methods. The expected error of DWUS on the unlabeled data can be evaluated as follows:

$$\hat{\epsilon_t}(DWUS) = \frac{1}{n_t} \sum_{i \in I_u} E[(\hat{y}_i - y_i)^2 \mid x_i] \tag{10}$$

where $E[(\hat{y}_i - y_i)^2 \mid x_i]$ is calculated as in Equation 9. Moreover, it is re-calculated at each iteration of active sampling. t is the iteration number, and n_t is the number of unlabeled instances at the t-th iteration and I_u is the set of indices of the unlabeled points at time t. By monitoring the average expected error at every single iteration, we can estimate when DWUS' performance starts to saturate, i.e., $\frac{\partial \hat{\epsilon}(DWUS)}{\partial x_t} \leq \delta$. δ is assigned a fixed small value in our evaluations [See Section 4.2 for how it was estimated]. When it is near zero, this is equivalent to detecting when a method is stuck in local minima/plateau in gradient descent methods. In fact, this principle is flexible enough to work with any two active learning methods where one is superior for labeling the initial data and the other is favorable later in the process. It generalizes to N sampling methods by introducing additional estimated switchover points based on estimated derivative of expected error for each additional sampling strategy.

We know that the strength of DWUS comes from the fact that it incorporates the density information into the selection mechanism. However, as the number of iterations increases uncertainty sampling outperforms DWUS and DWUS exhibits diminishing returns. We propose to use a mixture model for active sampling after we estimate the cross-over:

$$x_s^* = \arg\max_{i \in I_u} \pi_1 * E[(\hat{y}_i - y_i)^2 \mid x_i] + (1 - \pi_1) * p(x_i) \tag{11}$$

It is desirable for the above model to minimize the expected future error. If we were to select based on only the uncertainty, then the chosen point would be $x_{US}^* = \arg\max_{i \in I_u} E[(\hat{y}_i - y_i)^2 \mid x_i]$. After labeling x_{US}^*, the expected loss is:

$$f_{US} = \frac{1}{n} \sum_j E_{L+\{x_{US}^*, y\}}[(\hat{y}_j - y_j)^2 \mid x_j] \tag{12}$$

The subscript $L + \{x^*_{US}, y\}$ indicates that the expectation is calculated from the model trained on the data $L + \{x^*_{US}, y\}$. Assume f_{US}=0, then we can achieve the minimum expected loss by forcing $\pi_1 = 1$; hence $x^*_s = x^*_{US}$. The appropriate weight in this scenario is inversely related with the expected error of uncertainty sampling. Thus, we can replace the weights by $\pi_1 = 1 - f_{US}$, and $1 - \pi_1 = f_{US}$, and obtain the following model:

$$x^*_s = \arg\max_{i \in I_u} (1 - f_{US}) * E[(\hat{y}_i - y_i)^2 \mid x_i] + f_{US} * p(x_i) \tag{13}$$

Achieving the minimum expected loss is guaranteed only for the extreme case where the expected error, f_{US}, of uncertainty sampling is equal to 0. However, correlating the weight of uncertainty sampling with its generalization performance increases the odds of selecting a better candidate after the cross-over.

In the real world, we do not know the true value of f_{US}. So we need to approximate it. After estimating the cross-over, we are interested in giving higher priority to uncertainty, reflecting how well uncertainty sampling would perform on the unlabeled set. Therefore, we approximate f_{US} as $\hat{\epsilon}(US)$, the average expected error of uncertainty sampling on the unlabeled portion of the data. This leads us to the following selection criterion for DUAL:

$$x^*_s = \arg\max_{i \in I_u} (1 - \hat{\epsilon}(US)) * E[(\hat{y}_i - y_i)^2 \mid x_i] + \hat{\epsilon}(US) * p(x_i) \tag{14}$$

$\hat{\epsilon}(US)$ is updated at every iteration t after the cross-over. Its calculation is exactly the same as in Equation 10. However, the data to sample from is restricted to the already labeled examples by active selection. We construct a set with the actively sampled examples by DWUS until the cross-over, and call it set A. Uncertainty sampling is allowed to choose the most uncertain data point from only among elements in set A by estimating the posterior $P(y_i \mid x_i; \hat{\mathbf{a}}, \hat{b})$ over the initially labeled data. The chosen point is added to to the initial labeled set for uncertainty sampling and removed from set A. The average expected error of uncertainty sampling is calculated on the remaining unlabeled data. Then, DUAL selects the next data point to label via the criterion in Equation 14. This labeled point is also added to set A. Hence, set A is dynamically updated at each iteration with the actively sampled points. Consequently, in order to calculate the expected error of uncertainty sampling the algorithm never requests the label of a point that has not already been sampled during the active learning process. Such a restriction will prevent an exact estimate of the expected error. But, it is a reasonable alternative, and introduces no additional cost of labeling. The pseudo-code for the DUAL algorithm is given as *The Dual Algorithm.*

The DUAL Algorithm

```
program DUAL(Labeled data L, Unlabeled data U, max number of
iterations T, and $\delta$.)
begin
   Set the iteration counter t to 0.
   while(not switching point) do
```

```
        Run DWUS algorithm and compute $\frac{\partial \hat{\epsilon}(DWUS)}{\partial x_t}$.
        if($\frac{\partial \hat{\epsilon}(DWUS)}{\partial x_t} > \delta$)
            Choose the point to label:
            $x_s^* = \arg \max_{i \in I_u} E[(\hat{y}_i - y_i)^2 \mid x_i] p(x_i)$
            t=t+1 (Increment counter t)
        else Hit the switching point.
    while($t < T$)
            Compute $E[(\hat{y} - y)^2|x]$, $p(x)$ via DWUS, and $\hat{\epsilon}_t(US)$ via
            uncertainty sampling.
            Choose the point according to:
            $x_s^* = \arg \max_{i \in I_u} (1 - \hat{\epsilon}_t(US)) * E[(\hat{y}_i - y_i)^2 \mid x_i] + \hat{\epsilon}_t(US) * p(x_i)$
            t=t+1
end.
```

4.2 Experimental Setup

To evaluate the performance of DUAL, we ran experiments on UCI benchmarks: diabetes, splice, image segment, and letter recognition [18]. Some of these problems are not binary tasks so we used the random partioning into two classes as described by [13]. For the letter recognition problem, we picked three pairs of letters (M-vs-N, O-vs-D, V-vs-Y) that are most likely to be confused with each other. Thus, we examine six binary discrimination tasks. For each dataset, the initial labeled set is 0.4% of the entire data and contains an equal number of positive and negative instances. For clustering, we followed the same procedure used by [9] where the initial number of clusters is 20 and clusters are split until they reach a desired volume. The values of the parameters are given in Table 1 along with the basic characteristics of the datasets. These parameters and the δ parameter used for switching criteria were estimated on other data sets and held constant throughout our experiments, in order to avoid over-tuning. We compared the performance of DUAL with that of DWUS, uncertainty sampling, representative sampling[2] [8], density-based sampling and the COMB method of [10]. Density-based sampling adopts the same probabilistic framework as DWUS but uses only the density information for active data selection: $x_s^* = \arg \max_{i \in I_u} p(x_i)$. COMB uses an ensemble of uncertainty sampling, sampling method of [16], and a distance-based strategy choosing the unlabeled instance that is farthest from the current labeled set. COMB uses SVM with gaussian kernel for all three strategies. For further implementation details on COMB, see [10].

The performance of each algorithm was averaged over 4 runs. At each run, a different initial training set was chosen randomly. At each iteration of each algorithm, the active learner selected a sample from the unlabeled pool to be labeled. After it has been added to the training set, the classifier is re-trained and tested on the remaining unlabeled data and the classification error is reported. We also

[2] We used k=10 for k-means clustering as it produced better performance in [8], and selected the centroid of the largest cluster in the linear SVM margin.

Table 1. Characteristics of the Datasets, Values of the Parameters and p-value for significance tests after 40 iterations

Dataset	Total Size	+/- Ratio	dims(d)	sigma(σ)	lambda(λ)	DUAL>DWUS
Diabetes	768	0.536	8	0.5	0.1	$p < 0.0001$
Splice	3175	0.926	60	3	5	$p < 0.0001$
Image	2310	1.33	18	0.5	0.1	$p < 0.0001$
M-vs-N	1575	1.011	16	0.1	0.1	$p < 0.0001$
O-vs-D	1558	0.935	16	0.1	0.1	$p < 0.0001$
V-vs-Y	1550	0.972	16	0.1	0.1	$p < 0.0001$

conducted significance tests between DUAL and DWUS to report whether they perform significantly different. In order to determine whether two active learning systems differ statistically significantly, it is common to compare the difference in their errors averaged over a range of iterations [14,20]. Comparing performance over all 100 iterations would suppress detection of statistical differences since DUAL executes DWUS until cross-over. We conducted the comparison when they start to differ, which is on average after 40 iterations; we compute the two-sided paired t-tests by averaging from the 40th to 100th iteration. Table 1 shows that DUAL statistically outperforms DWUS in that range.

5 Observations and Conclusion

Figure 2 presents the improvement in error reduction using DUAL over the other methods. We only display results on 4 datasets due to space limitations. For the results on all datasets see `www.cs.cmu.edu/~pinard/DualResults`. DUAL outperforms DWUS and representative sampling both with $p < 0.0001$ significance.[3] DUAL outperforms COMB with $p < 0.0001$ significance on 4 out of 6 datasets, and with $p < 0.05$ on Image and M-vs-N data sets. We also calculate the error reduction of DUAL compared to the strong baseline DWUS. For instance, at the point in each graph after 3/4 of the sampling iterations after cross-over occurs, we observe 40% relative error reduction on O-vs-D data, 30% on Image, 50% on M-vs-N, 27% on V-vs-Y, 10% on Splice, and 6% on Diabetes dataset. These results are significant both statistically and also with respect to the magnitude reduction in relative residual error. DUAL is superior to Uncertainty sampling ($p < 0.001$) on 5 out of 6 datasets. We see on the V-vs-Y data that uncertainty sampling has the steepest decrease in error throughout all iterations. The cross-over between DWUS and uncertainty sampling occurs at a very early stage, but the current estimate of the expected error of DWUS to switch selection criteria is not accurate at the very early points in that dataset. Clearly, DUAL might have benefitted from changing its selection criterion at an earlier iteration.

As part of a failure analysis and in order to test this hypothesis, we conducted another set of experiments where we simulated a better relative error estimator

[3] All p-values reported are based on a 2-sided paired t-test on the classification error. All tests include the full operating range except Dual vs DWUS as explained before.

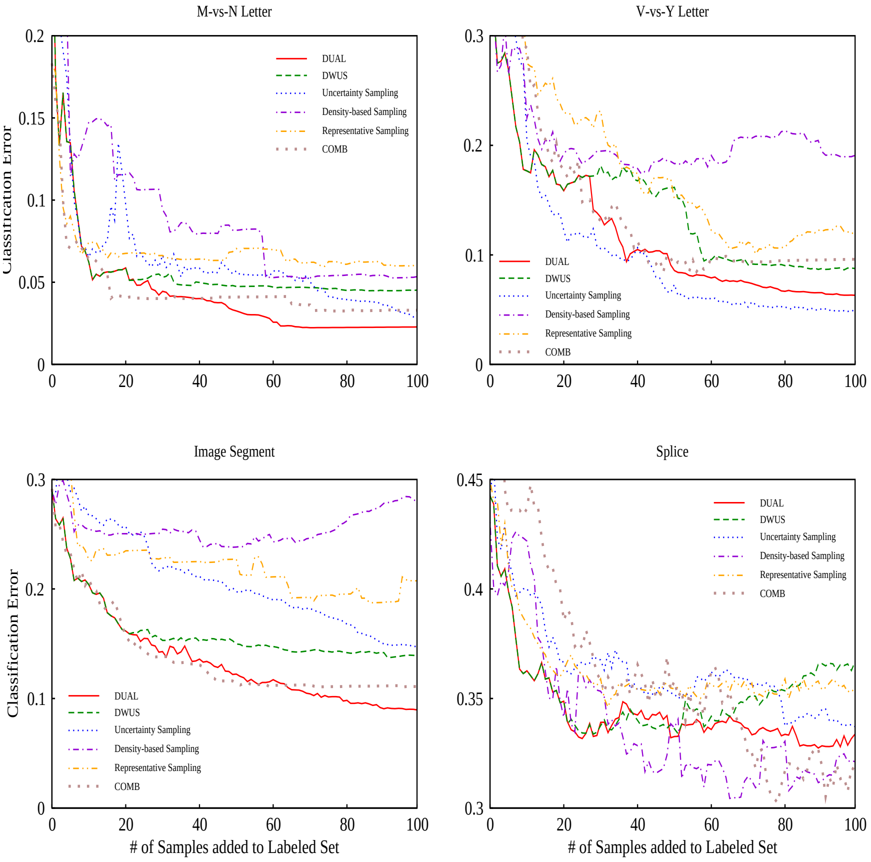
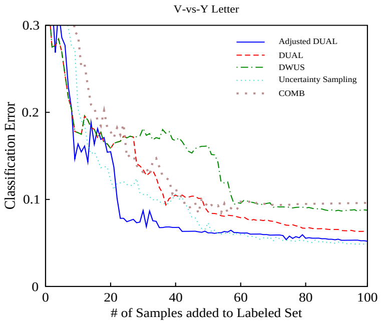


Fig. 2. Results on 4 different UCI benchmark datasets



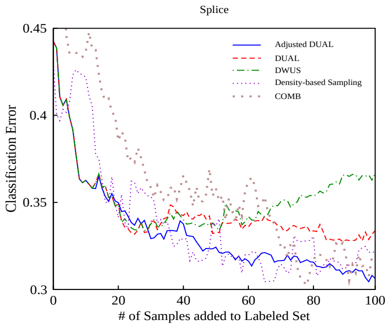


Fig. 3. Results after adjusting the switching point for DUAL on the V-vs-Y Letter data

Fig. 4. Results when DUAL is adjusted using Equation 15 on the splice data

for strategy switching. Fig. 3 demonstrates that DUAL outperforms all other methods when the true cross-over point is identified, indicating that better error estimation is a profitable area of research. In fact, one hypothesized solution is to switch when $P(error(M_2) \mid X) < P(error(M_1) \mid X) + \epsilon$, which considers the probability that over future selected instances method 2, M_2, will have less error than method 1, M_1. We plan to study more robust switching criteria.

DUAL outperforms Density-based sampling ($p < 0.0001$) on all but splice data. Density-based sampling performs worst for almost 40 iterations but then beats all of the others thereafter, totally breaking the pattern observed in the other datasets. Currently, DUAL only estimates how likely the uncertainty score is to lead to improvement, but the density-based method may also be likely to improve. One strategy is to calculate the expected error $\hat{\epsilon}(DS)$ of density-based sampling and modify Equation 14 to obtain the following:

$$x_s^* = \arg\max_{i \in I_u} \{\hat{\epsilon}(DS) * E[(\hat{y}_i - y_i)^2 \mid x_i] + (1 - \hat{\epsilon}(DS)) * p(x_i)\} \tag{15}$$

Fig. 4 presents the result after the modification in Equation 15. The adjustment helps DUAL make a significant improvement on the error reduction. Moreover, it consistently decreases the error as more data is labeled, hence its error reduction curve is smooth as opposed to the higher variance of density-based sampling. This suggests that pure density-based sampling is inconsistent in reducing error since it only considers the underlying data distribution regardless of the current model. Thus, we argue that DUAL may be more reliable than individual scoring based on density due to its combination formula that adaptively establishes balance between two selection criteria. Even though a strategy such as uncertainty or density based sampling performs well individually, Figures 2, 3 and 4 illustrate that it is more advantageous to use their combination.

To conclude, we presented DUAL which robustly combines uncertainty and density information. Empirical evaluation shows that, in general, this approach leads to more effective sampling than the other strategies. Xu et al. [8] also propose a hybrid approach to combine representative sampling and uncertainty sampling. Their method, however, only applies to SVMs and only tracks the better performing strategy rather than outperforming both individual strategies. Baram et al. also reports comparable performance for COMB to the best individual sampling strategy, but it is sometimes marginally better, and sometimes marginally worse and hence is not consistently the best performer. Our performance, on the contrary, exceeds that of the individually best sampling strategy in most cases by statistically significant margins. Hence, DUAL clearly goes beyond COMB in terms of lower classification error and faster convergence. Furthermore, our framework is general enough to fuse active learning methods that exhibit differentiable performance on the whole operating range. It can also be easily generalized to multi-class problems: one can estimate the error reduction globally or per-class using class-weighted or instance-weighted average, and then use the same cross-over criterion. While we use logistic regression, any probabilistic classifier can be adapted for use in DUAL. Our main contributions are in estimating the error of one method using the labeled data selected by another, and robustly integrating their outputs when one method is dominant (Equation 14 vs. Equation 15). Next, we intend to generalize DUAL to estimating which method is currently dominant for a problem or directly using a relative success weight. Our future plan is also to extend this work to ensemble methods that involve more than two strategies, maximizing ensemble diversity [15,14,10]. Moreover, we plan to investigate better methods for estimating the cross-over, such as estimating a smoothed version of $\frac{\partial \hat{\epsilon}}{\partial x_t}$ rather than a local-only version.

References

1. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory. (1992) 287–294
2. Freund, Y., Seung H., Shamir, E., Tishby, N.: Selective sampling using the Query By Committee algorithm. Machine Learning Journal, Vol. 28. (1997) 133–168
3. Lewis, D., Gale, W.: A sequential algorithm for training text classifiers. SIGIR '94 (1994) 3–12
4. Tong, S., Koller D.: Support vector machine active learning with applications to text classification. ICML '00 (2000) 999–1006
5. Mitchell, T.M.: Generalization as search. Artificial Intelligence Journal, Vol. 18 (1982)
6. McCallum, A., Nigam, K.: Employing EM and pool-based active learning for text classification. ICML '98 (1998) 359–367
7. Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. Journal of Artificial Intelligence Research, Vol. 4 (1996) 129–145
8. Xu, Z., Yu, K., Tresp, V., Xu, X., Wang, J.: Representative sampling for text classification using support vector machines. ECIR '03, Springer (2003)
9. Nguyen, H.T., Smeulders, A.: Active learning with pre-clustering. ICML '04 (2004) 623–630
10. Baram, Y., El-Yaniv, R., Luz, K.: Online choice of active learning algorithms. ICML '03 (2003) 19–26
11. Auer, P., Cesa-Bianchi N., Freund Y., Schapire, R. E.: The nonstochastic multi-armed bandit problem. SIAM Journal on Computing, Vol. 32(1) (2002) 48–77
12. Struyf, A., Hubert, M., Rousseeuw, P.: Integrating robust clustering techniques in s-plus. Computational Statistics and Data Analysis, Vol. 26 (1997) 17–37
13. Rätsch, G., Onoda, T., Muller, K. R.: Soft margins for AdaBoost. Machine Learning Journal, Vol. 42(3) (2001) 287–320
14. Melville, P., Mooney, R.J.: Diverse ensembles for active learning. ICML '04 (2004) 584–591
15. Melville, P., Mooney, R.J.: Constructing diverse classifier ensembles using artificial training examples. IJCAI '03 (2003) 505–510
16. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. ICML '01 (2001) 441–448
17. Schohn, G., Cohn, D.: Less is more: Active Learning with support vector machines. ICML '00 (2000) 839–846
18. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases (1998)
19. Saar-Tsechansky, M., Provost, F.: Active learning for class probability estimation and ranking. IJCAI '01 (2001) 911–920
20. Guo, Y., Greiner, R.: Optimistic Active Learning using Mutual Information. IJCAI '07 (2007) 823–829