An Active Learning Algorithm Based on Parzen Window Classification

Liang Lan LANLIANG@TEMPLE.EDU
Haidong Shi HAIDONGSHI@TEMPLE.EDU
Zhuang Wang ZHUANG@TEMPLE.EDU
Slobodan Vucetic VUCETIC@IST.TEMPLE.EDU
Department of Computer and Information Sciences
Temple University, Philadelphia, PA 19122 USA

Editor: I. Guyon, G. Cawley, G. Dror, V. Lemaire, and A. Statnikov

Abstract

This paper describes active learning algorithm used in AISTATS 2010 Active Learning Challenge as well as several of its extensions evaluated in the post-competition experiments. The algorithm consists of a pair of Regularized Parzen Window Classifiers, one trained on full set of features and another on features filtered using Pearson correlation. Predictions of the two classifiers are averaged to obtain the ensemble classifier. Parzen Window classifier was chosen because is an easy to implement lazy algorithm and has a single parameter, the kernel window size, that is determined by the cross-validation. The labeling schedule started by selecting random 20 examples and then continued by doubling the number of labeled examples in each round of active learning. A combination of random sampling and uncertainty sampling was used for querying. For the random sampling, examples were first clustered using either all features or the filtered features (whichever resulted in higher cross-validated accuracy) and then the same number of random examples was selected from each cluster. Our algorithm ranked as the 5th overall, and was consistently ranked in the upper half of the competing algorithms. The challenge results show that Parzen Window classifiers are less accurate than several competing learning algorithms used in the competition, but also indicate the success of the simple querying strategy that was employed. In the post-competition, we were able to improve the accuracy by using an ensemble of 5 Parzen Window classifiers, each trained on features selected by different filters. We also explored how more involved querying during the initial stages of active learning and the pre-clustering querying strategy would influence the performance of the proposed algorithm.

Keywords: Parzen Window, ensemble classifiers, clustering, feature filter, active learning

1. Introduction

Active learning addresses the problem in which large amounts of unlabeled data are available at low cost but where acquiring labels is expensive. The objective of active learning is to achieve high accuracy by using the least labeling effort. The AISTATS 2010 Active Learning Challenge (Guyon et al., 2010) considered the pool-based active learning scenario, where labels can be queried from a large pool of unlabeled data. In the challenge, the participants were given 6 binary classification tasks covering a wide variety of domains. All datasets were high-dimensional and with significant class imbalance.

In each task the participants were given a single seed example from a minority class and a large amount of unlabeled data to be queried. The virtual cash was used to acquire the class labels for selected examples from the server. Before acquiring labels, participants were asked to provide predictions for a separate test data set, which were used to measure performance of each active learning algorithm. At the completion of the challenge, the organizer released learning curves describing the evolution of the Area Under the ROC Curve (AUC) accuracy as a function of the number of labels, and also the Area under the Learning Curve (ALC) score summarizing the overall performance. The ALC score was designed to reward high accuracy when labeled datasets are small.

We used the Regularized Parzen Window Classifier (RPWC) (Chapelle, 2005) as the baseline classifier because of its ease of implementation and ability to solve highly nonlinear classification problems. We developed an ensemble of RPWC to enhance the overall performance. Due to the time constraints, we used only two component classifiers that differed in the features being used. Feature selection is a critical choice for Parzen Window classifiers because it influences the measure of distance between examples. For the challenge, we considered several feature selection filters.

Active learning has been a very active research topic in machine learning for many years, and we had a large choice of querying strategies. Uncertainty sampling (Lewis and Gale, 1994) is one of the most popular choices due to its simplicity and established effectiveness on many domains. In uncertainty sampling, a learner queries unlabeled examples that are the most uncertain, on which the existing models disagree the most (Seung et al., 1992) or that are expected to produce the greatest increase in accuracy (Cohn et al., 1996). By observing that pure uncertainty sampling can be too aggressive, a variety of approaches have been developed that combine uncertainty sampling with information about data density obtained from unlabeled data; see (Nguyen and Smeulders, 2004) and references within. Since classifiers trained on small labeled datasets could be very unreliable, pure random sampling during the initial stages of active learning can be very competitive to the more sophisticated active learning strategies.

The querying strategy used in our algorithm consisted of selecting a mixture of the most uncertain examples and randomly sampled examples. Instead of pure random sampling, which might over-emphasize dense regions of the feature space, we used random sampling of unlabeled data clusters. Due to the time limitations, we used an aggressive schedule that at each round of active learning doubled the number of labeled examples.

2. Methodology

In this section we describe the problem setup and explain the details of the active learning algorithm we used in the competition and for the post-competition experiments.

2.1. Problem Setup

Let us denote the labeled dataset as $L = \{(x_i, y_i), i = 1...N_l\}$, where x_i is an M-dimensional feature vector for the ith example and y_i is its class label. Initially, L contains a single seed example ($N_l = 1$). We are also given a pool of unlabeled data $U = \{(x_i), i = 1...N_u\}$.

The unlabeled dataset is assumed to be an *i.i.d.* sample from some unknown underlying distribution $p(x)$. Each example x_i has a label y_i coming from an unknown distribution $p(y|x)$.

At each stage of the active-learning process, it is possible to select a subset Q of examples from U, label them, and add them to L. Then, using the available information in L and U a classifier is trained and a decision to label another batch of unlabeled examples is made. At each stage, accuracy of the classifier is tested. Accuracies from all stages are used to evaluate the performance of the proposed active learning algorithm. The success in active learning depends on a number of design decisions, including data preprocessing, choice of classification algorithm, and querying strategy.

2.2. Proposed Active Learning Approach

The proposed active learning algorithm is outlined in Algorithm 1. At each round of the algorithm, we first trained F different base classifiers from the available labeled examples. Each classifier was built using the Regularized Parzen Window Classifier, described in §2.4. The classifiers differed by the features that were used. Several different feature selection filters explained in §2.6 were used. The final classifier was obtained by averaging predictions of the individual classifiers. To query new unlabeled examples, we used a combination of uncertainty sampling and random sampling, as explained in §2.8. At each round we doubled the number of queried examples. This querying schedule was used to quickly cover a range of the labeled set sizes, consistent with the way the organizers evaluated the performance of competing active learning algorithms.

Algorithm 1: Outline of the Active Learning Algorithm

```
Input: labeled set $L$, unlabeled set $U$; querying schedule $q_t = 20 \times 2^t$
Initialization: $Q \leftarrow$ randomly selected 20 unlabeled examples from $U$;
for $t$ = 1 to 10 do
    $U \leftarrow U - Q$;
    $L \leftarrow L + labeled(Q)$;
    for $j$ = 1 to $f$ do
        $F_j = feature\_filter_j(L)$;   /*feature selection filter*/
        $C_j = train\_classifer(L, F_j)$;   /*train classifier $C_j$ from $L$ using features $F_j$, determine model parameters by CV on $L$ */
        $A_j = accuracy(C_j, L)$;   /*estimate accuracy of classifier $C_j$ by CV on $L$*/
    end
    $C_{avg} \leftarrow average(C_j)$;   /*build ensemble classifier $C_{avg}$ by averaging*/
    $j^* = \arg\max A_j$;   /* find the index of the most accurate classifier*/
    Apply $k$-means clustering on $L + U$ using features from $F_{j^*}$;
    $Q \leftarrow 2q_t/3$ of the most uncertain examples in $U$ by the ensemble $C_{avg}$;
    $Q \leftarrow Q + q_t/3$ of random examples chosen from randomly selected clusters of $U$;
end
```

2.3. Data Preprocessing

Data preprocessing is an important step for successful application of machine learning algorithms to real word data. We used the *load_data* function provided by the organizers to load the data sets, which replaces the missing values with zeros and replaces Inf value with a large number. All non-binary features were normalized to have mean zero and standard deviation one.

2.4. Regularized Parzen Window Classifier

Choice of classification algorithm is very important in active learning, particularly during the initial stages when only a few labeled examples are available. For the active learning competition, our team selected the Regularized Parzen Window Classifier (RPWC) (Chapelle, 2005) as the base classifier. There are several justifications for our choice. RPWC is easy to implement and very powerful supervised learning algorithm able to represent highly non-linear concepts. It belongs to the class of lazy algorithms as it does not require training. RPWC gives an estimate of the posterior class probability of each example, which can be used to calculate the classification uncertainty for each unlabeled example.

RPWC can be represented as

$$p(y=1|x) = \frac{\sum_{y_i=1} K(x,x_i) + \varepsilon}{\sum K(x,x_i) + 2\varepsilon}$$

where ε is a regularization parameter, which should be a very small positive number and is relevant when calculating the classification uncertainty for examples which are underrepresented by the labeled examples. We set $\varepsilon = 10^{-5}$ in all our experiments. K is the Gaussian Kernel of the form

$$K(x,x') = \exp(\frac{-\|x-x'\|^2}{2\sigma^2})$$

where the σ represents the kernel size.

2.5. Kernel Size Selection

The kernel size σ is a free parameter which has a strong influence on the resulting estimate. When σ is small RPWC resembles nearest neighbor classification, while it resembles trivial classifier that always predicts the majority class when σ is large. Therefore, it has large influence on the representational power of the RWPC and should be selected with care. In this competition, we used leave-one-out cross-validation (CV) to select the optimal value for σ. We explored the following set of values for $2\sigma^2 : M/9, M/3, M, 3M$, where M is the number of features.

2.6. Feature Selection

Feature selection becomes a critical question when learning from high-dimensional data. This is especially the case during the initial stages of active learning when the number of labeled examples is very small. Feature selection filters are an appropriate choice for active learning because they are easy to implement and do not require training of a classifier. Typically, feature filters select features that have different statistics in positive and negative examples. In this competition, we considered statistical feature selection filters (Radivojac et al., 2004) based on the Pearson correlation and the Kruskal-Wallis test. We used only the Pearson correlation in the competition, while we also used the Kruskal-Wallis in our post-competition experiments.

2.6.1. All Features

As the baseline feature selection method, we simply used all the available features. This choice can be reasonable during the initial stages of active learning when the number of labeled examples is too small even for the simple statistical filters to make an informed decision.

2.6.2. Pearson Correlation

The Pearson correlation measures correlation between a feature and the class label and is calculated as

$$r_j = \frac{\sum_{i=1}^{N}(x_i^j - \bar{x}^j)(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i^j - \bar{x}^j)^2(y_i - \bar{y})^2}}$$

where $\bar{x}^j$ is the mean of the jth feature, $\bar{y}$ is the mean of the class label (negative examples were coded as 0 and positive as 1), and N the number of labeled examples. Value r_j measures linear relationship between jth feature and target variable and its range is between 1 or -1. Values of r close to 0 indicate the lack of correlation.

We used the Student's t-distribution to compute the p-value for the Pearson correlation that measures how likely it is that the observed correlation is obtained by the independent variables. We selected only features with p-values below 0.05. Since the number of selected features using the 0.05 threshold would largely depend on the size of the labeled dataset, as an alternative, we selected only the top M^* (e.g. $M^* = 10$) features with the lowest p-values.

2.6.3. Kruskal-Wallis Test

The t-statistics test for Pearson correlation assumes that the variables are Gaussian, which is clearly violated when the target is a binary variable. That is why the nonparametric Kruskal-Wallis (KW) test was also considered in our algorithms. For jth feature, KW sorts its values in ascending order and calculates average rank of all positive examples (denoted as r_j^+) and all negative examples (denoted as r_j^-). Then, the KW statistics of jth feature is calculated as

$$kw_j = \frac{12}{N(N+1)}(N^+(r_j^+ - \frac{N+1}{2})^2 + N^-(r_j^- - \frac{N+1}{2})^2)$$

where N^+, N^- are the numbers of positive and negative labeled examples, respectively. The KW statistics becomes large when the average ranks deviate from the expected rank $(N+1)/2$. The p-value of the KW statistic is calculated easily because kw_j follows the standard χ^2 distribution.

Similarly to the feature filter based on the Pearson correlation, given the p-values, features can be selected in two ways. In first, all features with p-values below 0.05 are selected. In second, M^* (e.g. $M^* = 10$) features with the smallest p-values are selected.

2.7. Ensemble of RPWC

Ensemble methods construct a set of classifiers and classify an example by combining their individual predictions. It is often found that ensemble methods improve the performance of the individual classifiers. (Dietterich, 2000) explains why ensemble of classifiers can often perform better than any single classifier from statistical, computational, and representational viewpoints. The following describes how we constructed an ensemble of RPWCs.

2.7.1. Base Classifiers

For the active learning challenge, we considered five base classifiers: (1) RPWC using all available (normalized) features; (2) RPWC on *filter_data1* (p-value of the Pearson correlations < 0.05); (3) RPWC on *filter_data2* (10 features with the smallest p-value of the Pearson correlations); (4) RPWC on *filter_data3* (p-value of the Kruskal-Wallis test < 0.05); (5) RPWC on *filter_data4* (10 features with the smallest p-value of the Kruskal-Wallis test).

2.7.2. Averaging Ensemble

In general, an ensemble classifier can be constructed by weighted averaging of base classifiers. For example, cross-validation can be used on labeled data to determine what choice of weights results in the highest accuracy. Since in an active learning scenario, the size of labeled data is typically small, we felt that weight optimization might lead to overfitting. Therefore, we decided to construct the ensemble predictor by simple averaging of the base RPWC as

$$p(y = 1|x) = \frac{1}{f}\sum_{i=1}^{f} p(y = 1|x, C_i).$$

2.8. Active Learning Strategy - Exploration and Exploitation

Our active learning strategy outlined in Algorithm 1 combined exploration and exploitation. We used a variant of random sampling for exploration and uncertainty sampling for exploitation. The queried examples at each stage were mixture of randomly sampled and uncertainty sampled examples from the unlabeled dataset.

Uncertainty Sampling Uncertainty sampling is probably the most popular active learning strategy. It measures prediction uncertainty of the current classifier and selects the most uncertain examples for labeling. There are two common ways to estimate uncertainty. If an ensemble of classifiers is available, uncertainty is defined as the entropy of their classifications. Using this definition, selected examples are either near the decision boundary or within underexplored regions of feature space. If a classifier can give posterior class probability, $p(y = 1|x_i)$, the alternative approach is to select examples whose probability is the closest to 0.5. We used this approach in our experiments since RPWC provides posterior class probabilities. It is worth to note that, thanks to the use of regularization parameter ε, examples with $p(y = 1|x_i)$ close to 0.5 in RPWC are both those near the decision boundary and those that are within the underexplored regions of the feature space. Both types of examples are interesting targets for active learning.

Clustering-Based Random Sampling Relying exclusively on uncertainty-based selection may be too risky, especially when the labeled dataset is small and the resulting classifier is weak. In addition, the uncertainty sampling for RPWC decribed in previous section is prone to selecting a disproportionate number of outliers. Therefore, in our approach we also relied on exploration to make sure examples from all regions of the feature space are represented in labeled data.

A straightforward exploration strategy is random sampling. However, this strategy can be wasteful if data exhibits clustering structure. In this case, dense regions would be over-explored while the less dense regions would be under-explored. This is why we decided to use k-means clustering to cluster the unlabeled examples as a preprocessing step at each stage of the active learning procedure. Then, our exploration procedure selected the same number of random unlabeled examples from each cluster. In k-means, we used randomly selected k examples as cluster seeds.

The result of clustering greatly depends on the choice of distance metric. To aid in this, we leveraged the existing base RPWCs. For clustering, we used the normalized features of the most accurate base classifier.

Combination of Uncertainty and Clustering Sampling As the initial selection in all 6 data sets, we randomly sampled 20 examples from the unlabeled data. Only after 20 examples were labeled we trained the first ensemble of RPWCs and continued with the procedure described in Algorithm 1. The justification is that we did not expect RPWC to be accurate with less than 20 labeled examples. For the same reason, we reasoned that clustering-based random sampling would not be superior to pure random sampling this early in the procedure.

In the following rounds of active learning, we sampled 2/3 of the examples using uncertainty sampling and 1/3 using the clustering-based random sampling.

Alternative Sampling Strategy Our sampling strategy that combines uncertainty-based and random sampling is similar to the pre-clustering idea from (Nguyen and Smeulders, 2004). They proposed to measure interestingness of unlabeled examples by weighting uncertainty with density. Examples with the highest scores are those that are both highly uncertain and come from dense data clusters. In our post-competition experiments we implemented the pre-clustering algorithm with two modifications. First, we used RPWC instead of the logistic regression to compute the conditional density. Second, instead of selecting unlabeled examples with the highest scores, we randomly selected from a pool of the highest ranked examples to avoid labeling of duplicate examples.

3. Experiments and Results

In this section we will describe the results of the experiments performed during the competition and for the post-competition analysis.

3.1. Data Description

The organizers provided datasets from 6 application domains. The details of each dataset are summarized in Table 1. All datasets are for binary classification and each is characterized by large class imbalance, with ratio between positive and negative examples near 1:10. The number of features in different datasets ranges from a dozen to several thousand.

Table 1: Dataset Description

Data	Feature	# Feature	Sparsity (%)	MissValue (%)	# Train	Maj Class (%)
A	mixed	92	79.02	0	17535	92.77
B	mixed	250	46.89	25.76	25000	90.84
C	mixed	851	8.6	0	25720	91.85
D	binary	12000	99.67	0	10000	74.81
E	cont.	154	0.04	0.0004	32252	90.97
F	mixed	12	1.02	0	67628	92.32

3.2. Experimental Setup

We applied the approach outlined in Algorithm 1. For the competition, we used an ensemble of only two RPWCs: one used all features and the other used features whose Pearson correlation p-value was below 0.05. This choice was made to speed-up our submission process. Before ordering labels for the first 20 random examples, we provided predictions that were proportional to the distance from the seed example. For k-means clustering we used $k = 10$ in all rounds of active learning experiments for all 6 datasets.

For all the experiments, we used the custom-made Matlab code. We used Matlab because it allowed us to implement the proposed Algorithm 1 in only a few days and complete all tasks form the competition within a week. During code development, we used several functions from the Statistics Matlab toolbox, such as the k-means clustering, the Kruskal-Wallis test, and the t-test.

3.3. Performance Evaluation

The classifier performance at each round of the active learning procedure was calculated by the challenge organizers as the Area Under the ROC Curve (AUC) on test data whose labels were withheld from the challenge participants. Performance of an active learning algorithm was calculated using the Area under the Learning Curve (ALC) calculated using the trapezoid method on the $log2$ scale. The log-scale enforced that larger emphasis is given to accuracy during the initial stages of active learning. The final score for each dataset was calculated as

$$global\ score = (ALC - Arand)/(Amax - Arand)$$

where $Amax$ is ALC of the *ideal learning curve*, which is obtained when the perfect predictions are made for the whole range of the labeled dataset sizes, (AUC =1). $Arand$ represents the ALC of the *'lazy' learning curve*, which is obtained by always making random predictions (where the expected value of AUC is always 0.5). Therefore, *global score* close to 0 indicates that the resulting active learning algorithm is not better than the random predictor, while value close to 1 indicates that very accurate classifier could be made using very few labeled examples.

3.4. Competition Results

We summarize the performance of our active learning algorithm in Tables 2 and 3 and Figure 1. Table 2 lists the *global score* on all 6 datasets. It compares the results from the best performer, the highest and second highest overall ranked teams, INTEL and ROFU, and of our team (TUCIS). The ranking of our algorithm was in the top half among all competing algorithms on all 6 datasets. Thanks to this consistency, our algorithm was ranked as the 5-th best algorithm overall. Our algorithm was better than INTEL on dataset D and better than ROFU on datasets C and F.

Table 3 summarizes the AUC accuracy of the final classifier trained using all purchased labels. As can be seen, RPWC was less accurate than the best reported classifier, the random forest used by INTEL, on all 6 datasets. The difference was the smallest on dataset F, while it was around 10

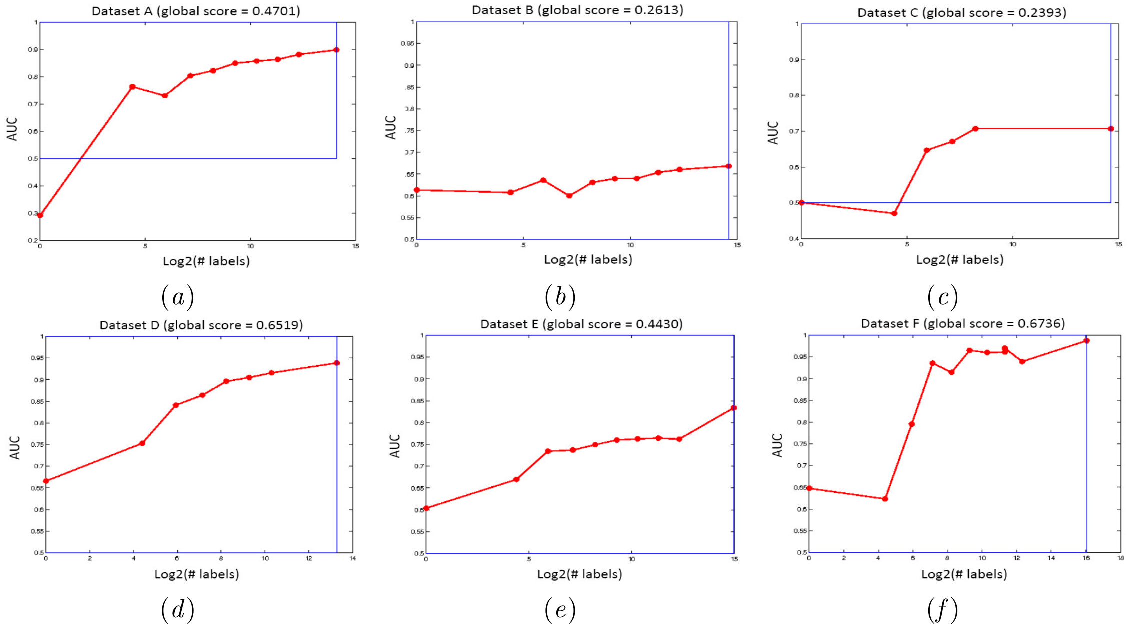
Figure 1 shows the learning curves that represent AUC classification accuracy as a function of the number of purchased labels. In all cases, a trend of increasing AUC with sample size can be observed. However, there are some significant differences in behavior of our algorithm on different datasets. On dataset A, the behavior is in accord to an intuitive notion that AUC should increase sharply until reaching the convergence region. On datasets B, D, and E, we can observe an almost linear increase in AUC with a logarithm of the sample size. It is interesting to observe that the linear AUC growth on the log-scale was assumed to be the lower bound by the competition organizers (Guyon et al., 2010). Our results on datasets B, C, and E indicate that this assumption was overly optimistic. As a consequence, building only a single predictor on a fully labeled dataset (the strategy cleverly used by ROFU team) would yield the same *global score* as our active learning algorithm on datasets B, D, and E. Performance on datasets C and F was characterized by a drop in AUC accuracy after labeling the first 20 examples that was significantly improved in the following labeling stages. The observed behavior could possibly be attributed to a significant class imbalance that was 91.85% for dataset C and 92.32% for dataset F. This result clearly illustrates that classifiers trained on small samples could be very unreliable and should be used with care in uncertainty sampling.

Table 2: Summary of the Competition Results: the (*global score*)

Dataset	Best	INTEL (1st)	ROFU (2nd)	TUCIS (5th)	TUCIS RANK
A	0.629	0.527	0.553	0.470	7
B	0.376	0.317	0.375	0.261	8
C	0.427	0.381	0.199	0.239	7
D	0.861	0.640	0.662	0.652	5
E	0.627	0.473	0.584	0.443	7
F	0.802	0.802	0.669	0.674	8

3.5. Post-Competition Results

Upon completion of the active learning challenge, we performed several additional experiments to better characterize the proposed algorithm and explore some alternatives. In order to finish this task we first queried labels of all training examples from the challenge server.



(a) (b) (c)

(d) (e) (f)

Figure 1: AUC versus the sample size on 6 datasets: the official results

Table 3: Summary of the Competition Results: the Final AUC scores

Dataset	Best	INTEL (1st)	ROFU (2nd)	TUCIS (5th)
A	0.962	0.952	0.928	0.899
B	0.767	0.754	0.733	0.668
C	0.833	0.833	0.779	0.707
D	0.973	0.973	0.970	0.939
E	0.925	0.925	0.857	0.834
F	0.999	0.999	0.997	0.987

Table 4: Comparison of two querying strategies

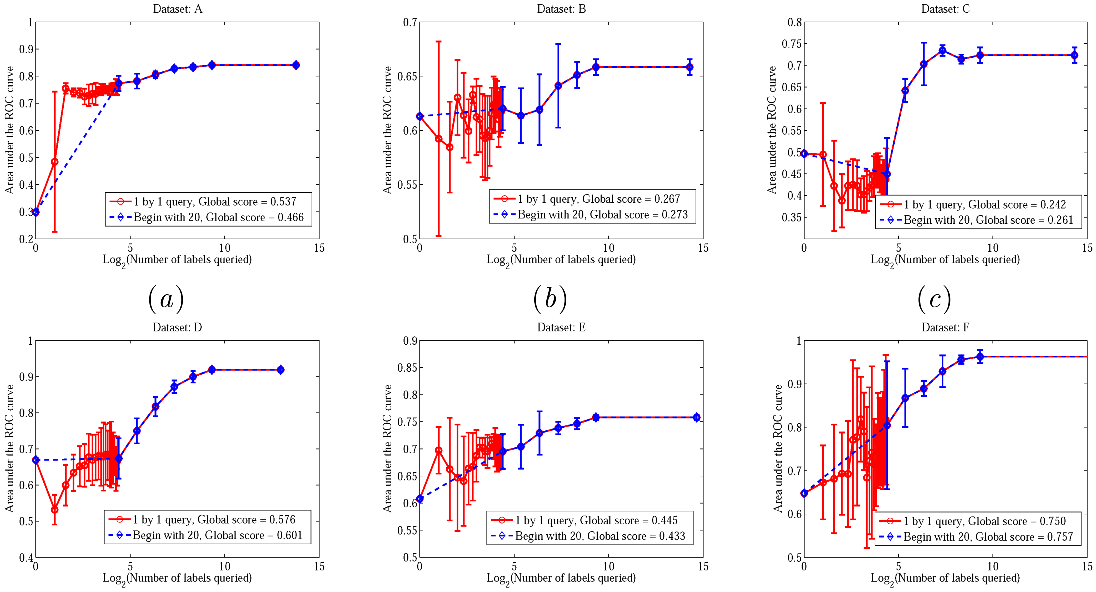
dataset	first 20 labels one by one	begin with 20 labels
A	**0.537±0.036**	0.466±0.015
B	0.267±0.022	**0.273±0.021**
C	0.242±0.034	**0.261±0.039**
D	0.576±0.035	**0.601±0.027**
E	**0.445**±0.028	0.433±0.017
F	0.750±0.046	**0.757±0.062**

Then, we randomly selected 80% of the labeled examples for training and 20% for testing. To be consistent with the challenge rules, in all experiments we used the same seed as used in the competition. All shown post-competition results are based on repeating the active learning experiment 5 times. In order to quickly finish the experiments, each active learning experiment was terminated after 320 examples were labeled (after 5 rounds of Algorithm 1).

Early Start Our competition algorithm started by labeling 20 randomly sampled and building a classifier from them. Our first question was what would be the *global score* if we trained classifiers after every single labeled example, $N_l = 1, 2, ..., 20$. Since $log2$ scale was used in calculating ALC, any success on the first 20 examples would be visible in the final score. Similarly, any failure to make a good learning progress would be reflected negatively. In Table 4, we compared the results of the submitted algorithm (tested on 20% of training examples, as explained above) to the same algorithm that built a series of classifiers until labeled data size reached 20 (after that, we followed exactly the same procedure as in Algorithm 1). Consistent with Algorithm 1, for the first 20 queries we used exclusively random sampling. As can be seen, the *global score* of the alternative approach increased only on dataset A, decreased on dataset D, and remained similar on the remaining 4 datasets. This result confirmed our original strategy from Algorithm 1. Another result from this set of experiments is worth mentioning. Table 4 also lists the standard deviation of *global score* after repeating each experiment 5 times. It can be seen that the variability is relatively low (around 0.046). The largest difference 0.05 was observed on dataset F.

To more clearly illustrate the evolution of AUC at different sample sizes, in Figure 2 we are showing the learning curves on the 6 datasets. It can be seen from Figure 2 that the variability in AUC values slowly decreases with the number of labels in all datsets. On dataset A it could be seen that the accuracy rapidly increases after only two additional examples are labeled and that it approaches the maximum achievable accuracy. This clearly explains the increase in ALC reported in the first row of Table 4. However, the similar behavior is not replicated on the remaining 5 datasets. In fact, on 4 of the datasets (B, C, D, F) the accuracy grows slower than the postulated log-scale linear growth. On three of them, we could even observe the initial drop in accuracy that explains significant drop in ALC reported in Table 3.

These results clearly demonstrate the challenges in design of a successful active learning strategy when the number of labeled examples is small. This is particularly the case when a dataset is highly dimensional, with a potentially large number of irrelevant, noisy, or weak features. In this case, there simply might be too little data to make informed decisions



(a) (b) (c)

(d) (e) (f)

Figure 2: Competition and Post-Competition AUC curves

Table 5: Comparison of different ensemble methods

Data	1 classifier (full)	1 classifier(PR)	1 classifier(KW)	2 Classifiers	5 Classifiers
A	0.463±0.080	0.433±0.040	0.442±0.026	**0.466±0.015**	0.459±0.017
B	0.261±0.005	0.246±0.035	0.283±0.011	0.273±0.021	**0.304±0.023**
C	0.321±0.023	0.292±0.038	0.297±0.059	0.261±0.039	**0.337±0.050**
D	**0.670±0.027**	0.540±0.029	0.545±0.051	0.601±0.027	0.551±0.050
E	0.426±0.007	0.417±0.024	0.412±0.050	0.433±0.017	**0.450±0.007**
F	0.620±0.023	0.785±0.026	**0.792±0.039**	0.757±0.062	0.779±0.026

about feature selection, classifier training, and sampling strategy. This view is supported by the competition results where each dataset seems to have favored different active learning approach.

Larger Ensemble Due to the need to finalize all experiments before the challenge deadline, our ensemble consisted only of two RPWCs. Here, we explored what would be the impact of accuracy if we used 5 RPWC, each using a different feature selection filter, as explained in §2.7.1. Table 5 summarizes the results. It can be seen that the 5-classifier ensemble is more accurate than our competition algorithm on 4 challenge datasets and is less accurate only on dataset D. This behavior is consistent with the large body of work on ensemble methods Dietterich (2000). The obtained result indicates that using a larger variety of feature selection methods would result in improved performance of our approach. Table 5 also lists results of several individual classifiers. Their performance was overall worse than the performance of both ensembles. It is interesting to observe that RPWC that used all feature was the most successful individual classifier on 4 of the 6 datasets,

Table 6: Comparison of two querying strategies

dataset	1/3 RAND + 2/3 UNCTN	Precluster
A	**0.466±0.015**	0.447±0.012
B	**0.273±0.021**	0.170±0.057
C	0.261±0.039	**0.293±0.049**
D	**0.601±0.027**	0.507±0.036
E	**0.433±0.017**	0.378±0.058
F	**0.757±0.062**	0.709±0.062

and that its performance was particularly impressive on dataset D and particularly poor on dataset F.

Active Learning by Preclustering Finally, we explored the performance of preclustering algorithm by Nguyen and Smeulders (2004) outlined in §2.8. The results in Table 6 shows that preclustering was inferior than the 2/3 uncertain + 1/3 random approach used in our challenge algorithm. We explain this result by a strong reliance of pre-clustering on the classification model that might be unreliable when trained with small number of labeled examples. Nevertheless, this is a slightly surprising result deserving further analysis.

4. Conclusion

The results of the AISTATS 2010 Active Learning Challenge show that our proposed algorithm based on Parzen Window classification and mixture of uncertainty-based and random sampling gives consistent results, that are somewhat lower than the winning algorithms. Our analysis reveals that, although it was less accurate than the winning decision forests, Parzen Window classification was a reasonable choice for solving highly dimensional classification problems. In addition, since several of the challenge datasets could be accurately solved using linear classifiers Guyon et al. (2010), the representative power of Parzen Window classifiers becomes a disadvantage. Our results also show that accuracy of Parzen Window classifiers can be boosted by using their ensembles. On the other hand, the ease of implementation, coupled with respectable accuracy achieved on the challenge tasks, indicates that Parzen Window classifiers should be given consideration on any active learning problem.

Using the mixture of uncertainty and random sampling proved to be a successful strategy. It shows that good performance on active learning problems requires a right mix of exploration and exploitation strategies. Our results confirm that at initial stages of active learning, when there are few labeled examples, the exploration should be given precedence over exploitation. This can be easily attributed to low quality of knowledge that can be acquired from small labeled datasets. This paper also illustrated the benefits of analyzing properties of unlabeled data (e.g. through clustering, or semi-supervised learning) and importance of feature selection when addressing high-dimensional active learning problems.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation Grant IIS-0546155.

References

O. Chapelle. Active learning for parzen window classifier. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2005.

D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. In *Journal of Artificial Intelligence research*, volume 4, 1996.

T. Dietterich. Ensemble methods in machine learning. In *J. Kittler and F. Roli, editors, the 1st International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science*. Springer-Verlag, 2000.

I. Guyon, G. Cawley, G. Dror, and V. Lemaire. Results of the active learning challenge. In *Journal of Machine Learning Research: Workshop and Conference Proceedings*, volume 10, 2010.

D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, 1994.

H. Nguyen and A. W. Smeulders. Active learning using pre-clustering. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004.

P. Radivojac, Z. Obradovic, A. K. Dunker, and S.Vucetic. Characterization of permutation tests for feature selection. In *Proceedings of the 15th European Conference on Machine Learning (ECML)*, 2004.

H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the 5th ACM Workshop on Computational Learning Theory (COLT)*, 1992.