Paired-Sampling in Density-Sensitive Active Learning

Pinar Donmez and **Jaime G. Carbonell**
Language Technologies Institute, Carnegie Mellon University,
5000 Forbes Ave., Pittsburgh, PA 15213, USA
{pinard, jgc}@cs.cmu.edu

Abstract

Active learning consists of principled on-line sampling over unlabeled data to optimize supervised learning rates as a function of the number of labels requested from an external oracle. A new sampling technique for active learning is developed based on two key principles: 1) Balanced sampling on both sides of the decision boundary is more effective than sampling one side disproportionately, and 2) exploiting the natural grouping (clustering) of unlabeled data establishes a more meaningful non-Euclidean distance function with respect to estimated category membership. Our new paired-sampling density-sensitive method embodying these principles yields significantly superior performance in multiple active learning data sets over all other sampling methods in our comparative study: representative sampling, uncertainty sampling, density-based sampling, and random sampling.

1 Introduction

In many domains ripe for supervised machine learning techniques, obtaining large amounts of unlabeled data is easy but obtaining class labels is costly and time-consuming. For instance, it is easy to crawl the web, but much more costly to pay an army of human topic labelers. Likewise, it is simple to collect images, but much harder to obtain good linguistic content labels. It is also easier to obtain geological data pertaining to regions that may contain oil, but much more costly to drill multiple deep test holes to know which ones really contain oil. Active learning consists of optimizing sampling strategies over the unlabeled data in order to maximize the accuracy of supervised machine learning methods and to minimize the number of samples that require definitive categorization for training. Typically, the learner starts with a very small number of labeled examples, trains a classifier or ranker, selects new sample(s) from the unlabeled data in an on-line fashion, one or few at a time, re-trains the learner and iterates. The objective is to optimize accuracy at every step in the sampling-learning cycle.

Considerable research has focused in sampling strategies from a large volume of unlabeled data to optimize learning from the fewest number of labeled instances (Lewis & Gale 1994; Cohn, Ghahramani, & Jordan 1996; McCallum & Nigam 1998; Schohn & Cohn 2000; Tong & Koller 2000; Melville & Mooney 2004). These approaches range from uncertainty sampling (Lewis & Gale 1994), to representative sampling (Xu *et al.* 2003), to density-based sampling (Nguyen & Smeulders 2004) to active ensemble methods (Melville & Mooney 2004; Donmez, Carbonell, & Bennett 2007). While these methods all provide interesting insight and functional active learning strategies, other factors could be considered as well, in order to further improve active sampling. With this goal in mind, we developed a new sampling strategy based on: 1) maximizing the likelihood of straddling the decision boundary with paired samples, 2) a transformed distance function to effectively reduce distance as a function of local density, and 3) rely on a utility-based conditional-entropy maximization criterion to combine factors in making the sampling decision. As we show in the empirical results section, the new sampling strategy proves to be quite effective vis-$\grave{a}$-vis the popular active-learning sampling methods: representative sampling, density-based sampling, uncertainty sampling and random sampling.

In the sections that follow, we first outline a transformation of the data exploiting the cluster hypothesis, which states that the decision boundary should lie in low density regions (i.e. inter-cluster, vs intra-cluster). In section 2.2, we derive a sampling criterion that favors pairs of points straddling the decision boundary with maximum utility. We present experimental results in section 3 that demonstrate the superiority of the proposed method and finally we provide conclusions in section 4.

2 Density-Sensitive Sampling

In order to sample points that are likely to be maximally informative to an active learner, we first seek to maximize the chance that we will sample on both sides of a decision boundary – sampling disproportionately on either side will not optimize boundary placement in the learning process. Maximizing the distance between two points is a step in the right direction, but Euclidean distance may not be the optimal measure; instead we investigate density-sensitive distance functions.

2.1 Density-Sensitive Distance Estimation

According to the cluster hypothesis, the decision boundary should lie in low density regions, and hence should not cut

clusters (Chapelle & Zien 2005). Our goal is to represent the data in such a way that points in separate clusters are assigned high-distances (equivalent to low similarities). In order to enforce this criterion, we chose to derive pairwise similarities/dissimilarities in a fully-connected graph-based representation of the data. Let $G = (V, E)$ be a graph where V is the set of nodes each of which denotes a data point and E denotes the edges between nodes. Edge weights are Euclidean distances, i.e. $\|x - y\|$. $p \in V^l$ is defined as a path of length $l = |p|$ that connects the nodes x_i and x_j if $(p_k, p_{k+1}) \in E$ for $1 \leq k < l$, and $p_1 = x_i$ and $p_l = x_j$. Points in the same cluster can be connected via a path traveling in that cluster, thereby a high density region. Conversely, any path connecting points in different clusters has to travel along a low density region. The density-sensitive distance between any two points can be approximated by first selecting the longest distance edge along each path, i.e. the weakest link, then repeating this process for every path that connects these two points, and finally finding the minimum among the longest distance edges. This approach was first proposed by (Fischer, Roth, & Buhmann 2004) and used for clustering:

$$d(x_i, x_j) = \min_{p \in P_{i,j}} \max_{1 \leq k < |p|} \|p_k - p_{k+1}\| \quad (1)$$

where $P_{i,j}$ is the set of all paths that connects x_i and x_j. The above formulation does not take into account the length of the paths. A long path connecting two points in different clusters might have a very short edge; hence that single outlier would dramatically disrupt the distance approximation. In order to avoid this problem, we incorporate the path length into the above equation by taking the sum over the edge distances instead of the maximum:

$$d(x_i, x_j) = \frac{1}{\rho} \left\{ \ln(1 + \min_{p \in P_{i,j}} \sum_{k=1}^{|p|-1} (e^{\rho \|p_k - p_{k+1}\|} - 1)) \right\} \quad (2)$$

Equation 2 is proposed by Chapelle & Zien (2005). Equation 1 and 2 are equivalent when $\rho \to \infty$. For large values of ρ, the distances between points in the same cluster are decreased whereas the distances between points in different clusters are still dominated by the gaps between clusters. For small values of ρ, every edge contributes to the distance calculation. We follow their approach by applying Multidimensional Scaling (MDS) (Cox & Cox 1994) to the dissimilarity matrix D, where $D_{ij} = d(x_i, x_j)$ in Equation 2 to obtain a Euclidean representation of a set of objects while preserving their distance relationships. MDS first transforms the distance matrix D into a new matrix A by definining $a_{ij} = -\frac{1}{2} D_{ij}^2$. Matrix A is used to derive matrix $\Delta = [\delta_{ij}]$ such that $\delta_{ij} = a_{ij} - \bar{a}_i - \bar{a}_j + \bar{a}$, where $\bar{a}_i$ and $\bar{a}_j$ are row and column means of A, respectively; and $\bar{a}$ is the mean of all elements in A. The eigenvalues $(\lambda_1, \lambda_2, ..., \lambda_k)$ and eigenvectors $(u_1, u_2, ..., u_k)$ of Δ are computed, and the latter is scaled so that $\sqrt{u_k' u_k} = \sqrt{\lambda_k}$. Chapelle & Zien (2005) showed that it is safe to discard the eigenvectors with small eigenvalues; hence we followed their formulization by taking only the first p eigenvectors that satisfy the following inequality:

$$\sum_{i=1}^{p} \lambda_i \geq (1 - \delta) \sum max(0, \lambda_i)$$
$$where \;\; \lambda_p \leq \delta \lambda_1 \;\; and \;\; \lambda_1 \geq ... \geq \lambda_n \geq 0 \quad (3)$$

The δ parameter is fixed at 0.1 as specified in (Chapelle & Zien 2005), though it could potentially be optimized. Let U be an n x p matrix whose columns are the scaled eigenvectors, then the rows of U are the coordinates of the objects in MDS space, i.e. $\widetilde{x}_{i,\cdot} = U_{i,\cdot}$. The time complexity to compute the distance matrix D is $O(n^2(n + logn))$ when Dijkstra's shortest path length algorithm is adopted to implement the search for the next closest unexplored node in the graph using a binary heap (Chapelle & Zien 2005). This is the implementation we used in the paper. The MDS transformation takes $O(n^3)$ time since it computes the eigenvectors of an n x n matrix. However, if a k nearest neighbor graph is used instead of a fully-connected graph, and if only the first p eigenvectors are considered, the time complexity for both steps can be reduced.

2.2 Density-Sensitive Paired Sampling

Given a set of training data points in MDS space $(\boldsymbol{X}, y) = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_m, y_m)\}$, we use logistic regression to obtain the posterior class distribution. But our approach is designed to be used with any probabilistic classifier including Gaussian processes or Bayesian optimal classifiers. We focus on binary problems in our evaluations, though our method can be easily adapted to multi-class cases. We provide information on handling multi-class problems as appropriate throughout the paper. The logistic regression model is

$$P(y \mid \boldsymbol{x}, \boldsymbol{w}) = \sigma(y \boldsymbol{w}^T \boldsymbol{x}) = \frac{1}{1 + exp(-y \boldsymbol{w}^T \boldsymbol{x})} \quad (4)$$

where $y \in \{-1, +1\}$. We use the regularized version to find the parameter vector $\boldsymbol{w}$ which minimizes the negative log-likelihood:

$$l(w) = \sum_{i=1}^{m} \log(1 + exp(-y_i \boldsymbol{w}^T \boldsymbol{x}_i)) + \frac{\lambda}{2} \boldsymbol{w}^T \boldsymbol{w} \quad (5)$$

The minimization problem is convex so it can be solved by a number of iterative algorithms. We use iteratively reweighted least squares method: $\boldsymbol{w}_{new} = \boldsymbol{w}_{old} - \boldsymbol{H}^{-1} \boldsymbol{g}$, where $\boldsymbol{g}$ and $\boldsymbol{H}$ are the gradient and Hessian of $l(w)$, respectively:

$$\begin{aligned} \frac{\partial l(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \lambda \boldsymbol{w} + \sum_{i=1}^{m} -y_i x_i (1 - p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w})) \\ \frac{\partial^2 l(\boldsymbol{w})}{\partial^2 \boldsymbol{w}} &= \lambda + \sum_{i=1}^{m} \boldsymbol{x}_i \boldsymbol{x}_i^T p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w})(1 - p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w})) \end{aligned} \quad (6)$$

If there are m instances of d dimensions, it takes $O(md^2)$ time per iteration.

In order to maximize the likelihood of straddling the decision boundary, and to halve the computational time, we

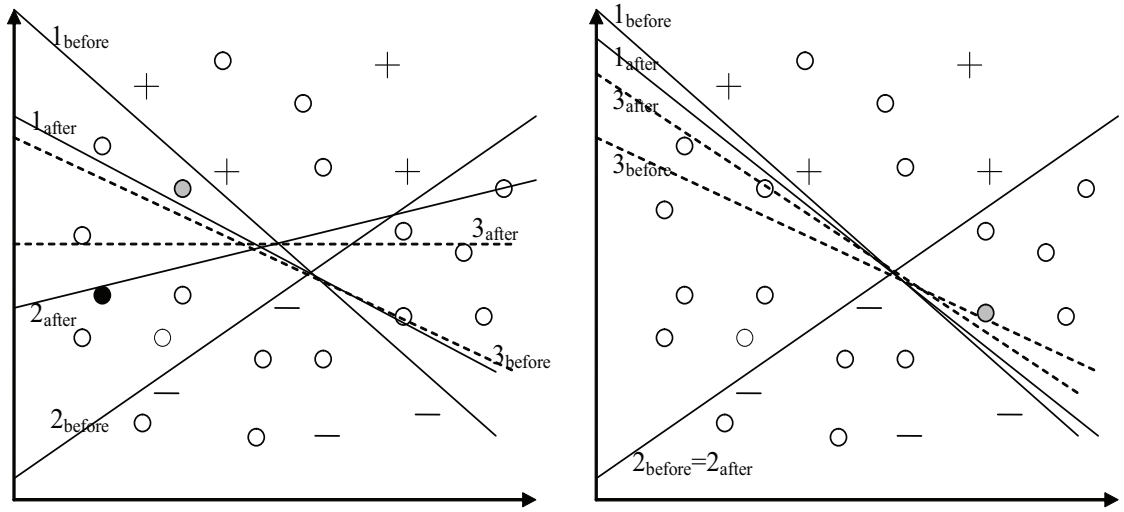

Figure 1: Illustrative Example: The plus (minus) sign and circles indicate the positively (negatively) labeled points and unlabeled data, respectively. x_{after} and x_{before} indicate the line before and after data is sampled for labeling. The selected points are labeled either positive (shown in grey) or negative (shown in black). This example illustrates our motivation to sample two points with opposite labels at a time instead of a single point.

sample a pair of points to label at a time, in contrast to the traditional active learning methods that select one point at each iteration. Figure 1 illustrates the motivation for paired sampling in active learning. Here we assume for simplicity the data is linearly separable. The dashed line shows the current decision boundary while the two solid lines define the region where the true boundary is expected to lie; namely the version space. The left figure in Figure 1 is an example of sampling a pair for labeling from opposite sides of the current boundary. It greatly reduces the version space since both points affect how the version space will be bounded. The current boundary also shifts significantly. On the other hand, the figure on the right shows that only a single point is sampled for labeling. The amount of shift in the current hypothesis is relatively small. The version space is not reduced as significantly as in the previous scenario since only one point contributes to the reduction. These two scenarios illustrate why it is more advantageous to straddle the decision boundary in order to reduce the set of candidate hypotheses rapidly. With this goal in mind, we strive to sample two points with opposite class labels. In multi-class scenarios, this is equivalent to sampling as many points as the number of classes at each iteration of active learning, seeking to maximize the chance of sampling each class once per round. Since the labels of the unlabeled data are unknown, we need to approximate the likelihood that any two points have opposite class labels, $P(y_i \neq y_j \mid x_i, x_j)$, for all $i, j \in I_u$ where I_u is the set of indices of the unlabeled points in the data. By our cluster assumption, points in different clusters are likely to have different labels. In the new representation of the data, points in different clusters are assigned low similarity. It is then reasonable to define $P(y_i \neq y_j \mid x_i, x_j)$ as proportional to the distance between x_i and x_j, i.e. $P(y_i \neq y_j \mid x_i, x_j) \propto \|x_i - x_j\|^2$. For an empirical analysis justifying this claim, see Appendix.

As the goal of active learning is to learn the model parameters accurately with the least number of labeled examples, the selected instances need to be informative, e.g. the points whose labels we are most uncertain about. Uncertainty-based active learning strategies have been proposed by a number of researchers (Lewis & Gale 1994; Tong & Koller 2000; Campbell, Cristianini, & Smola 2000; Schohn & Cohn 2000). Such strategies work fairly well in practice, and have nice theoretical properties related to VC dimension reduction (Tong & Koller 2000). Thus, in order to obtain a faster learning rate we need to select two points that are likely to have opposite labels *and* high uncertainty. We first define a scoring function for each pair of unlabeled points as follows:

$$\begin{aligned} S(i,j) &= P(y_i \neq y_j \mid x_i, x_j) * U(i,j) \\ &= c\|x_i - x_j\|^2 * U(i,j) \end{aligned} \tag{7}$$

where c is a normalization constant for $P(y_i \neq y_j \mid x_i, x_j)$, and $U(i,j)$ is a complex utility score which will be explained soon. Before doing so, let us give an outline of how our method works:

1. Compute the distance matrix D using Equation 2 and transform the entire data into the MDS space
2. Compute the pairwise Euclidean distances, $\|x_i - x_j\|$, of the transformed data

3. Train the logistic regression classifier using the current training set in its transformed form and estimate the posterior class probabilities $P(y \mid \boldsymbol{x}, \hat{\boldsymbol{w}})$
4. For all $i \neq j \in I_u$
   (a) Compute the score $S(i,j)$ using Equation 7
5. Choose for labeling the points $\boldsymbol{x}_{i^*}, \boldsymbol{x}_{j^*}$ which have the highest score $S(i,j)$, add them to the training set and remove i^*, j^* from I_u.
6. Repeat 3-5 until a desired amount is sampled

Another important factor for active sampling is to select points from high density regions. It is shown to boost the performance in various studies (Cohn, Ghahramani, & Jordan 1996; Zhang & Chen 2002; Xu *et al.* 2003; Nguyen & Smeulders 2004; Donmez, Carbonell, & Bennett 2007). Obtaining the label of an instance with high density has the advantage that it will significantly increase our confidence in the labels of the neighbors. One drawback with this approach is that it does not take into account the current learner's predictions. High density points may already be correctly labeled by the current learner with high confidence. In this case, there is no much benefit in querying points with dense neighborhoods because it will not provide much information about the labels of the remaining unlabeled instances.

For a given point $\boldsymbol{x}$, $p(\boldsymbol{x})$ can be estimated as the average similarity to the remaining points, $\frac{\frac{\sum_{i=1}^{n} exp(-\|\boldsymbol{x}-\boldsymbol{x}_i\|^2)}{n-1}}{Z_n}$, where n is the total number of points, and Z_n is the normalization constant. From an active learning point of view, however, we are more interested in the close neighborhood of a point since it will directly be affected by the labeling of that point. Thus, we constrain the density estimation to the points in a local neighborhood. That is, the density estimate for a given point will depend only on those unlabeled neighbors whose distance to the point is smaller than a pre-defined threshold:

$$\hat{p}(\boldsymbol{x}) = \frac{\sum_{k \in N_x} exp(-\|\boldsymbol{x} - \boldsymbol{x}_k\|^2)}{Z'_n} \tag{8}$$

where $N_x = \{r \in I_u |\ \|x - x_r\| < t\}$ is the set of indices of the unlabeled points whose distance to $\boldsymbol{x}$ is smaller than the threshold t. Z'_n is again the normalization constant. Note that Equation 8 is not an average; it does not divide by the size of the neighborhood; $|N_x|$. By enforcing the estimate in Equation 8, we guarantee that it depends on *the number of neighbors* as well as *their proximity*. As we discussed earlier, a density measure itself cannot fully capture the information content of a point in terms of the amount of surprise we would get if we knew the true label. The conditional entropy of the unknown label y given the instance $\boldsymbol{x}$ and the model $\boldsymbol{w}$ is:

$$H(Y \mid \boldsymbol{x}, \boldsymbol{w}) = -\sum_{y} P(y \mid \boldsymbol{x}, \boldsymbol{w}) \log P(y \mid \boldsymbol{x}, \boldsymbol{w}) \tag{9}$$

It measures the amount of information (uncertainty) of the discrete random variable Y, and is maximum when $P(y \mid \boldsymbol{x}, \boldsymbol{w}) = \frac{1}{|Y|}$, where $|Y|$ is the number of values that the class variable Y can get. For binary problems, i.e., $y \in \{-1, +1\}$, we have the following equality:

$$\underset{i \in I_u}{\operatorname{argmax}} H(Y_i \mid \boldsymbol{x_i}, \boldsymbol{w}) = \underset{i \in I_u}{\operatorname{argmax}} \{ \min_{y_i \in \{\pm 1\}} \{P(y_i \mid \boldsymbol{x_i}, \boldsymbol{w})\}\} \tag{10}$$

We adopted the latter for the experiments reported in this paper. For multi-class problems, the conditional entropy can be equivalently used. Since we do not know the true model $\boldsymbol{w}$, we used its approximation $\hat{\boldsymbol{w}}$ from the logistic regression classifier trained with the data seen up to the present point. Finally, we propose using an uncertainty weighted density measure:

$$\hat{p}(\boldsymbol{x}) = \sum_{k \in N_x} \exp(-\|\boldsymbol{x} - \boldsymbol{x}_k\|^2) * \min_{y_k \in \{-1,+1\}} \{P(y_k \mid \boldsymbol{x_k}, \hat{\boldsymbol{w}})\} \tag{11}$$

For simplicity, we leave out the normalization constant since we are interested in the relative density rather than the absolute density. Equation 11 captures both the density of a given point and also the information content of its neighbors. Furthermore, each neighbor's contribution to the density score is weighed by its uncertainty; hence it reduces the effect of the neighbors at which the current learner has high confidence. Formally, we define the utility $U(i,j)$ of a pair of points as the sum of the density estimate for each point. By the definition of N_x, it includes the point $\boldsymbol{x}$ in consideration. Hence, Equation 11 includes the uncertainty of the point itself, $\min_{y \in \{-1,+1\}} \{P(y \mid \boldsymbol{x}, \boldsymbol{w})\}$, as a summand with weight equals to $\exp(-\|\boldsymbol{x} - \boldsymbol{x}\|^2) = 1$. We propose to give more flexibility to that uncertainty term by introducing a regularization coefficient. It quantifies a trade-off of the information content of an instance with the proximity weighted information content of its neighbors. This allows us to define the utility function as follows:

$$\begin{aligned} U(i,j) &= \log\{\hat{p}(\boldsymbol{x_i}) + \hat{p}(\boldsymbol{x_j})\} = \\ &\log\Bigg\{ \sum_{k \neq i \in N_{x_i}} \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_k\|^2) * \min_{y_k \in \{\pm 1\}} \{P(y_k \mid \boldsymbol{x_k}, \hat{\boldsymbol{w}})\} \\ &+ \sum_{r \neq j \in N_{x_j}} \exp(-\|\boldsymbol{x}_j - \boldsymbol{x}_r\|^2) * \min_{y_r \in \{\pm 1\}} \{P(y_r \mid \boldsymbol{x_r}, \hat{\boldsymbol{w}})\} \\ &+ s * (\min_{y_i \in \{\pm 1\}} \{P(y_i \mid \boldsymbol{x_i}, \hat{\boldsymbol{w}})\} + \min_{y_j \in \{\pm 1\}} \{P(y_j \mid \boldsymbol{x_j}, \hat{\boldsymbol{w}})\}) \Bigg\} \end{aligned} \tag{12}$$

Note x_i and x_j are treated separately in the last summand where s is the regularization constant. We tried a range of values from 1 to 3 for s on another dataset that is not reported in this paper. Different values did not effect the results in any significant way; hence we picked $s = 2$ which is reasonable given the restriction on the size of the neighborhood. Equation 12 is substituted into Equation 7 to get the final score $S(i,j)$. Thus, our strategy is to select instances for labeling that have the largest score:

$$\{i^*, j^*\} = \underset{i \neq j \in I_u}{\operatorname{argmax}} S(i,j) = \underset{i \neq j \in I_u}{\operatorname{argmax}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 * U(i,j) \tag{13}$$

The pseucode of the algorithm is given as Algorithm 1.

Algorithm 1 Paired Sampling

Input: Data $(\boldsymbol{X}, y) = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_m, y_m)\}$
Output: Logistic Regression Classifier
Program
Compute the distance matrix D
for all $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \boldsymbol{X}$ **do**
$D_{ij} = \frac{1}{\rho}\left\{\ln(1 + \min_{p \in P_{i,j}} \sum_{k=1}^{|p|-1}(e^{\rho\|p_k - p_{k+1}\|} - 1))\right\}$
end for
Apply MDS to D to obtain the data in MDS space $(\widetilde{\boldsymbol{X}}, y) = \{(\widetilde{\boldsymbol{x}}_1, y_1), ..., (\widetilde{\boldsymbol{x}}_m, y_m)\}$
Divide the data into training set T and unlabeled set U s.t. $(\widetilde{\boldsymbol{X}}, y) = T \cup U$
repeat
Train logistic regression on T to get $P(y \mid \widetilde{\boldsymbol{x}}, \hat{\boldsymbol{w}})$
for all $i \neq j \in I_u$ **do**
Compute $S(i,j) = \|\widetilde{\boldsymbol{x}}_i - \widetilde{\boldsymbol{x}}_j\|^2 * U(i,j)$ using Equation 12
end for
Pick $\{i^*, j^*\} = \text{argmax}_{i \neq j \in I_u} S(i,j)$
Update $T = T \cup \{(\widetilde{\boldsymbol{x}}_{i^*}, y_{i^*}), (\widetilde{\boldsymbol{x}}_{j^*}, y_{j^*})\}$ and $I_u = I_u - \{i^*, j^*\}$
until stopping criterion

Data	Breast	Heart	Flare	Face	Glass2	g50c
Size	277	270	1066	2500	163	550
+/-	0.413	0.800	1.234	1	1.144	1
Dim	9	13	13	400	9	50

Table 1: Properties of the datasets used in our experiments

3 Experimental Results

3.1 Data

We conducted a set of experiments in order to evaluate our method on six binary datasets: one artificial dataset, four real world datasets from UCI Repository (Newman *et al.* 1998), and one face detection dataset used in (Pham, Worring, & Smeulders 2002). The original face dataset has 393360 images in total from which we used a random subsample of size 2500. The artifial dataset, called g50c, is used in (Huang & Kecman 2005; Chapelle & Zien 2005; Collobert *et al.* 2006). It is generated from two unit-covariance normal distributions with equal probabilities, and the class means are adjusted so that the Bayes error is 5%. Table 1 gives information about the datasets. All the predefined parameters are tuned and fixed on a separate dataset not reported in this paper; i.e. λ in Equation 5 is fixed at 0.1, the scaling parameter ρ in Equation 2 is fixed at 1, the threshold t for determining the number of neighbors for each unlabeled point $\boldsymbol{x}$ is fixed so that the size of N_x will not exceed 15.

3.2 Experiments

For each dataset, we conducted 10 runs. For each run, we randomly picked just 2 instances, one from each class, to form the initial training set. This number is usually larger for many active learning studies including (Nguyen & Smeulders 2004; Schein & Ungar 2005). We left the remaining data as the unlabeled pool. We ran each active learning method for 20 iterations and at each iteration we selected 2 instances to label. Hence, we actively sampled 40 instances in total. Every time a new pair of samples is added to the training set, the classifier is re-trained and evaluated on the remaining unlabeled portion of the data. At each iteration, we reported the error of the active sampling method. We averaged those results over 10 runs for comparison. We compared our proposed method with four other strategies:

1. Most Uncertain: We rank the unlabeled points according to their uncertainty, i.e., $\min_y\{P(y \mid \boldsymbol{x}, \hat{\boldsymbol{w}})\}$ (via Equation 4), in descending order. Then, we select the top two points with the most uncertainty.
2. Density Only: It differs from the proposed method by considering only the proximity of the neighbors for computing the density.
3. Representative Sampling (Xu *et al.* 2003): The unlabeled points that fall inside the margin are clustered using k-means in a linear SVM framework. The centroids of the two largest clusters are chosen to be labeled. Penalty factor C in SVM, and k in clustering are optimized minimizing the test error to obtain the best possible performance[1].
4. Random Sampling

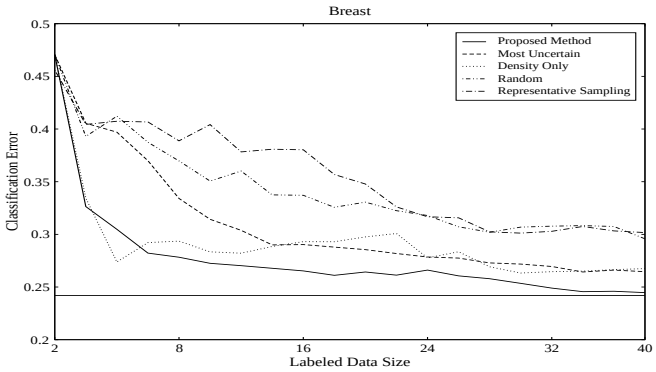

Figure 2: Results on UCI Breast data. The solid horizontal line indicates the 10-fold cross-validation error using the entire data as the training data.

3.3 Results

Figure 2 shows the results on the UCI Breast data comparing the five methods. Our method has the steepest decrease in error as well as the density-only version of our method. However, our method has the lowest final error rate and does better than all the other methods. Even though the most uncertain and the density only version are not individually the best performers, our approach combines the best of each approach and yields superior results. Representative sampling

[1]Parameter tuning minimizing the test error has only been used for representative sampling. Parameters in other methods are tuned as explained in Section 3.1.

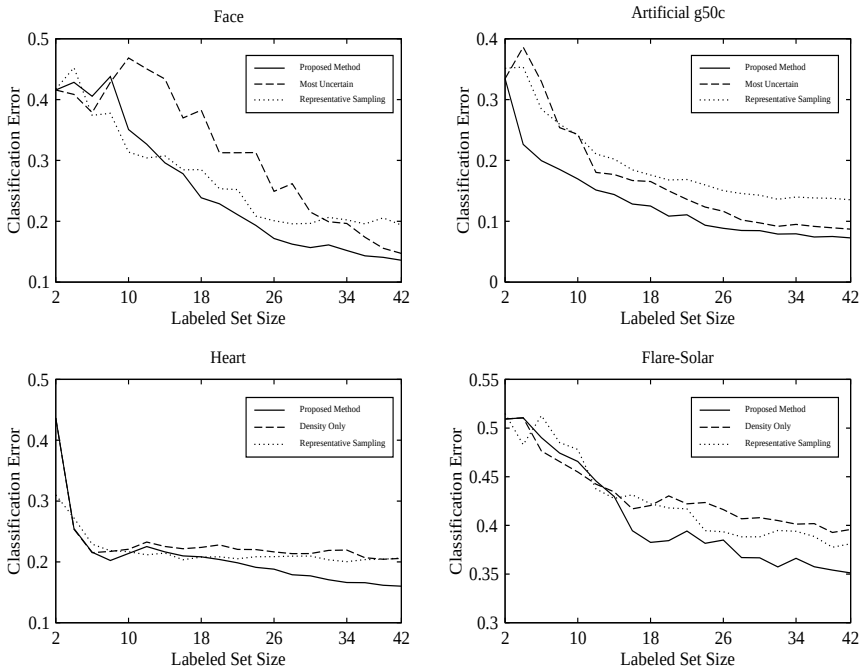

Figure 3: Results on four different datasets

does worse than random sampling at the beginning, but their performances converge towards the end. We noted that the final error rate for our method is close to the 10-fold cross-validation error on the entire data for all 6 problems, which we explicitly show on Breast data. We also noticed that our method selects a pair with opposite labels for the majority of the time. Figure 3 shows the results on four of the remaining datasets. We only show three methods in each graph to ease visual legibility. The top two graphs in Figure 3 compare our method against uncertainty sampling and representative sampling, whereas the bottom two graphs compare it against the density only version and representative sampling. Our method outperforms the others on each data. The density only version performs slightly better than our method for the initial iterations on Flare-Solar, and similarly representative sampling performs slightly better on early iterations on Face detection. But our method readily achieves significantly better performance on both cases as more data is sampled. A more thorough comparison of all methods on six datasets is given in Table 2.

In Table 2, we show the error rates for each method at three different points in iteration: 5th, 11th and 17th iterations. The first column in Table 2 shows the dataset and the corresponding iteration at which the error rates are compared. The percentage error reduction against the random sampling baseline is given in parenthesis. Lowest error rates are given in bold. Our method wins on the majority of the cases. Whenever it loses, there is only a slight difference between our method and the winner so our method is still comparable on cases where it is not the best. Furthermore, it can be seen that each method does worse than random sampling on Flare 5th. We note the poor separability of the data; thus we plan to examine the relation between the difficulty of a classification task and the capacity of active learning methods as a follow-up work.

We see that our method is the best on all except few cases. To quantify this, we did a 2-sided paired t-test at the 95% confidence level on the entire reported operating range to test the hypothesis that our method has significantly lower error than each of its competitors. Thus, it was tested against each method separately and the corresponding p-values were recorded. Our method always performed significantly better ($p < 0.001$) than the density only version on all datasets. It also outperformed most uncertain with $p < 0.001$ on all except the Heart data where $p < 0.05$. It outperformed random sampling on Flare with $p < 0.05$, on Face with $p < 0.01$ and with $p < 0.001$ on the rest. Moreover, it outperformed representative sampling with $p < 0.001$ on Breast, Flare, g50c, and with $p < 0.05$ on Face whereas both are comparable on Glass2 and Heart datasets. However, Table 2 shows that our method improves more steeply and wins in the later iterations on these two datasets. When we only compared the errors for the last 10 iterations on Glass2 and Heart, then our method wins with $p < 0.05$ and $p < 0.001$, respectively.

We also conducted another set of experiments to evaluate the cluster assumption. We re-ran our method without transforming the data. In other words, we computed the Euclidean pairwise distances in the original input space, and selected the instances to label according to Equation 13. It performed worse than or comparable with our original method. On Heart and g50c they both did equally well. In fact, the average absolute difference between the errors of the two methods on Heart data is 0.016 ± 0.009, and 0.01 ± 0.005 on g50c data. On Glass2, Flare, Face and Breast datasets the

Data	Proposed Method	Most Uncertain	Density Only	Representative	Random
Breast 5	**0.278 (-24.6%)**	0.334 (-9.04%)	0.293 (-20.5%)	0.380 (+2.9%)	0.369
Breast 11	**0.264 (-20%)**	0.285 (-13.6%)	0.297 (-10%)	0.347 (+5.1%)	0.330
Breast 17	**0.249 (-18.8%)**	0.269 (-12.3%)	0.264 (-14%)	0.302 (-1.6%)	0.307
Heart 5	**0.213 (-18.3%)**	0.245 (-6.1%)	0.220 (-15.7%)	0.216 (-17.2%)	0.261
Heart 11	**0.198 (-4.3%)**	0.208 (+0.4%)	0.220 (+6.2%)	0.205 (-0.9%)	0.207
Heart 17	0.166 (-13.5%)	**0.164 (-14.5%)**	0.219 (+14%)	0.20 (+4.1%)	0.192
Flare 5	0.465 (+5.2%)	0.454 (+2.7%)	0.454 (+2.7%)	0.478 (+8.1%)	**0.442**
Flare 11	**0.394 (-1.6%)**	0.451 (+10%)	0.422 (+2.9%)	0.417 (+1.7%)	0.410
Flare 17	**0.366 (-8.7%)**	0.449 (+11.9%)	0.401 (0%)	0.393 (-1.9%)	0.401
Face 5	0.350 (-1.9%)	0.468 (+31%)	0.420 (+17.6%)	**0.313 (-12.3%)**	0.357
Face 11	**0.210 (-23.3%)**	0.312 (+13.8%)	0.287 (+4.7%)	0.252 (-8%)	0.274
Face 17	**0.151 (-32.5%)**	0.196 (-12.5%)	0.189 (-15.6%)	0.202 (-9.8%)	0.224
Glass2 5	0.339 (-11%)	0.442 (+16%)	0.392 (+2.8%)	**0.326 (-14.4%)**	0.381
Glass2 11	0.317 (-7%)	0.341 (0%)	0.324 (-4.9%)	**0.31 (-9%)**	0.341
Glass2 17	**0.266 (-8.9%)**	0.292 (0%)	0.275 (-5.8%)	0.30 (+2.7%)	0.292
g50c 5	**0.169 (-46.3%)**	0.242 (-23.1%)	0.187 (-40.6%)	0.241 (-23.4%)	0.315
g50c 11	**0.110 (-37.8%)**	0.136 (-23.1%)	0.128 (-27.6%)	0.168 (-5%)	0.177
g50c 17	**0.079 (-34.1%)**	0.094 (-21.6%)	0.102 (-15%)	0.139 (+15.8%)	0.120

Table 2: Comparison of five different active learners on all datasets

untransformed version is outperformed by our method with $p < 0.001$ significance.

4 Conclusion

In this paper, we explored a proximity-weighted conditional-entropy-based criterion for active learning. This approach is unique in two ways: First, it combines the density, uncertainty and dissimilarity-across-classification-boundary strategies into a unified framework. Second, it uses a density-sensitive distance metric to measure the dissimilarity between pairwise instances, maximizing the likelihood of sampling both sides of a decision boundary in a totally unsupervised process. Distances of points within the same cluster are reduced while those from different clusters are dominated by the inter-cluster distances. We presented empirical results on various domains. The results demonstrate that our method outperforms others in terms of both error reduction and fewer number of labeling queries required to obtain a certain level of accuracy. We note that the time complexity of the data transforming process prohibit the application to very large datasets. In the future, we plan to address efficiency improvements, for instance by extending kd-trees and by computing a k-nearest-neighbor fanout graph, vs the full graph. We further note that our scoring function $S(i, j)$ must be computed for each pair of points in the unlabeled pool, which takes $O(|I_u|^2)$ time per iteration. In order to reduce computational cost, we rank the unlabeled points from most to least uncertain. The top p% is selected and pairwise scores are computed for this subset. The algorithm then picks instances to label from this representative subset of unlabeled data. This is only enforced on the Flare and Face datasets by setting p to 30% and 20%, respectively. We also plan to extend this work to other probabilistic classifiers, such as Gaussian Process Classifiers, which should require minimal effort, and we also plan to explore the effects of different kernels on the active learning technique we proposed in this paper.

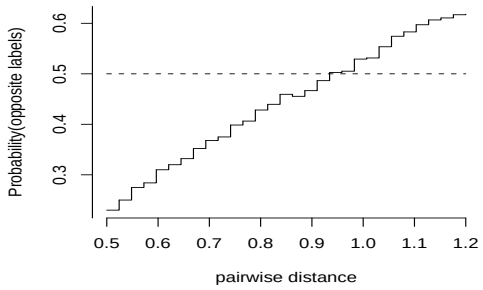

Figure 4: Graph of $\hat{P}(y_i \neq y_j \mid x_i, x_j)$ versus $\|x_i - x_j\|$ on g50c dataset

5 Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010.

6 Appendix

We estimated the probability $P(y_i \neq y_j \mid x_i, x_j)$ as a function of the pairwise distance $\|x_i - x_j\|$. Figure 4 is generated on g50c dataset. We sorted the pairwise distances in increasing order and divided them into 30 equal intervals. For each interval, all pairs (x_i, x_j) with distance $\|x_i - x_j\|$ falling within that interval were examined. $P(y_i \neq y_j \mid x_i, x_j)$ was estimated as the relative frequency of pairs in that interval with opposite class labels. As shown in Figure 4, $P(y_i \neq y_j \mid x_i, x_j)$ monotonically increases with the pairwise distance. This analysis empirically shows that $\hat{P}(y_i \neq y_j \mid \|x_i - x_j\|) \geq \hat{P}(y_i \neq y_k \mid \|x_i - x_k\|) \Leftrightarrow \|x_i - x_j\| \geq \|x_i - x_k\|$. The curve may differ for other datasets, but if

the class membership is a well-defined (e.g. smooth) function, the same principle applies. The dotted line is the probability $P(y_i \neq y_j) = P(y_i = 1, y_j = -1) + P(y_i = -1, y_j = 1)$, independent of any knowledge regarding the data distribution. Since binary classes are equally balanced on this dataset, this probability is 0.5. The absolute difference between the two curves at any point indicates the loss $|P(y_i \neq y_j) - \hat{P}(y_i \neq y_j \mid \|x_i - x_j\|)|$ introduced by relying on $\|x_i - x_j\|$. Hence, sampling distant pairs increases the likelihood that they have opposite class labels without sacrificing a large penalty. This procedure is conducted only to support our claim, i.e. $P(y_i \neq y_j \mid x_i, x_j) \propto \|x_i - x_j\|^2$; the proposed active sampling strategy is carried on a completely unsupervised manner.

References

Campbell, C.; Cristianini, N.; and Smola, A. 2000. Query learning with large margin classifiers. In *ICML '00*.

Chapelle, O., and Zien, A. 2005. Semi-supervised classification by low density separation. In *AISTATS '05*.

Cohn, D.; Ghahramani, Z.; and Jordan, M. 1996. Active learning with statistical models. *Journal of Artificial Intelligence Research* 4.

Collobert, R.; Sinz, F.; Weston, J.; and Bottou, L. 2006. Large scale transductive svms. *Journal of Machine Learning Research* 7:1687–1712.

Cox, T., and Cox, M. 1994. *Multidimensional Scaling*. Chapman & Hall.

Donmez, P.; Carbonell, J.; and Bennett, P. 2007. Dual strategy active learning. In *ECML '07*, 116–127.

Fischer, B.; Roth, V.; and Buhmann, J. 2004. Clustering with the connectivity kernel. In *NIPS '04*, volume 16.

Huang, T., and Kecman, V. 2005. Performance comparisons of semi-supervised learning algorithms. In *ICML Workshop on Partially Classified Training Data*, 45–49.

Lewis, D., and Gale, W. 1994. A sequential algorithm for training text classifiers. In *ACM-SIGIR Conference on Research and Development in Information Retrieval*, 3–12.

McCallum, A., and Nigam, K. 1998. Employing em and pool-based active learning for text classification. In *ICML '98*, 359–367.

Melville, P., and Mooney, R. J. 2004. Diverse ensembles for active learning. In *ICML '04*, 584–591.

Newman, D.; Hettich, S.; Blake, C.; and Merz, C. 1998. UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences.

Nguyen, H., and Smeulders, A. 2004. Active learning using pre-clustering. In *ICML '04*, 623–630.

Pham, T.; Worring, M.; and Smeulders, A. 2002. Face detection by aggregated bayesian network classifiers. *Pattern Recognition Letters* 23(4):451–461.

Schein, A., and Ungar, L. 2005. Active learning for multiclass logistic regression. *Learning*.

Schohn, G., and Cohn, D. 2000. Less is more: Active learning with support vector machines. 839–846.

Tong, S., and Koller, D. 2000. Support vector machine active learning with applications to text classification. In *ICML '00*.

Xu, Z.; Yu, K.; Tresp, V.; Xu, X.; and Wang, J. 2003. Representative sampling for text classification using support vector machines. In *ECIR '03*.

Zhang, C., and Chen, T. 2002. An active learning framework for content-based information retrieval. *IEEE Trans on Multimedia* 4:260–268.