Active Learning via Random Walk

Ashin Mukherjee , Jie Cheng

December 17, 2009

1 Introduction

The basic goal in any classification problem is to identify a decision rule which assigns each test instances into one of the possible classes with a high degree of **accuracy**. Learning a good classifier requires a sufficient number of labeled training instances. However, the labels are often difficult, expensive or time consuming to obtain, as they may require experienced human annotators, for example, *Text Categorization* etc. On the other hand, unlabeled data may be available in large quantity. Randomly selecting unlabeled instances for labeling is inefficient in many situations. Hence **Active Learning** methods have been adopted to improve the generalization performance of a classifier.

We have a set of instances $X = X_l \cup X_u$ where $X_l = \{x_1, x_2, ..., x_l\}$ is the set of labeled instances and $X_u = \{x_{l+1}, ..., x_n\}$ is the set of unlabeled instances. Usually, we will have a very small number of labeled instances to start with and a large pool of unlabeled data. An *Active Learning System* will sequentially select the most informative instances from the unlabeled pool to label, with the goal of developing an efficient classifier.

In this project, we propose a novel approach to develop an active learning system based on random walk on graphs. We start by constructing a graph whose nodes stand for the instances and edge weights are induced by certain measures(e.g., k-nearest neighbor or radial basis kernel) to reflect similarity or closeness. A random walk on this graph is designed with transition probability proportional to the edge weights. Under this framework, we use the *conditional expected hitting times* from unlabeled nodes to labeled nodes to build the classification rule and also to choose the most informative set of unlabeled instances to query for the next step. The goal is to improve the generalization performance of the learning algorithm while labeling as few instances as possible to reduce the cost.

The rest of the report is organized as follows. Section 2 briefly reviews the existing methods on active learning algorithms. In section 3 we present the theoretical framework for the proposed method and section 4 describes the algorithm and discusses various aspects of it. Then we show the results on simulation data in section 5 and conclude with comments and discussions in section 6.

2 Previous Work

We summarize several recently proposed approaches in active learning in this section. Most of these use some heuristic score or greedy procedure to select the instances or batches. SVM has been the most popular choice for the classifier since it performs well in high-dimensional problems, especially for text-categorization or image classification type problems, which is one huge area of applied interest.

Schohn and Cohn (2000) proposed a SVM based heuristic method, which selects the k closest points to the separating hyperplane to label. This takes care of the *instance uncertainty* factor but fails to account for outliers and *batch diversity* of the subset selected. That is we do not want to pick points that are very close in the instance space even if they have high uncertainty, since we expect high overlap of information between them. Xu et. al. (2003) improves on it by taking into account of the *diversity* using clustering structure of support vectors. More recently Guo and Schuurmans (2007) comes up with a model based approach, where they optimize a combination of log-likelihood and negative entropy, which stands for the *instance uncertainty*. Several graph based methods have been proposed in the field of semi-supervised learning. Zhu et. al.(2002) proposed the idea of label-propagation through network structure. More recently Camps-Valls (2007) used a spectral clustering type of algorithm to address the same problem.

3 Theoretical Formulation

Suppose we are given a sample $\mathcal{X} = \mathcal{X}_l \cup \mathcal{X}_u = \{x_1, \ldots, x_l\} \cup \{x_{l-1}, \ldots, x_n\}$ with $\mathcal{X}_l$ denoting the labeled instances and $\mathcal{X}_u$ the unlabeled instances. We construct an undirected graph $G = (V, E, W)$ based on the data. V denotes the node set, where we identify each node with an instance; E denotes the set of edges and $W = (w_{ij})_{n \times n}$ denotes the matrix of edge-weights. W is a symmetric similarity matrix. The common choices for construction of W are radial basis kernel, (i.e., $w_{ij} = e^{-\|x_i - x_j\|^2/2\sigma^2}$) or the k-NN kernel.

A random walk is defined on this graph in the following fashion: given the current state x_i, the probability of moving to x_j is –

$$P_{ij} = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}} \tag{1}$$

We consider only the two-class problem, where the two classes are denoted by $'+'$ and $'-'$. By appropriately arranging the order of nodes, the transition matrix could be rewritten as,

$$P = \begin{pmatrix} P_{++} & P_{+-} & P_{+u} \\ P_{-+} & P_{--} & P_{-u} \\ P_{u+} & P_{u-} & P_{uu} \end{pmatrix} \tag{2}$$

Consider a walk that starts from an unlabeled node x_u and stops whenever it hits a labeled node. Let T be the number of steps it takes till the random walk stops. The

distance between the node x_u and the class $'+'$ is defined as –

$$\mathbf{d(x_u, +)} = \mathbb{E}(\text{T} \mid \text{the random walk starts from } \mathbf{x_u}, \text{ and stops at class '+'}) \quad (3)$$

Result 1. Given a transition probability matrix $\mathbf{P}$ in the form mentioned above, the distance vector of unlabeled nodes from the class $'+'$, $d(X_u, +)$, is given by

$$d(X_u, +) = (I - P_{uu})^{-2} P_{u+}\mathbf{1}./(I - P_{uu})^{-1} P_{u+}\mathbf{1} \quad (4)$$

where ./ is the element-wise division, $\mathbf{1}$ denotes a column of ones of length equal to the set of points labeled '+', and $d(X_u, +)$ is a vector of length same as the number of unlabeled nodes. $d(X_u, -)$ can be computed in the same way.

Proof: For a single node x_u,

$$\begin{aligned}
d(x_u, +) &= \mathbb{E}(T \mid \text{the random walk starts from } x_u, \text{ and stops at class '+'}) \\
&= \sum_{k=1}^{\infty} k \cdot \mathbb{P}(\text{the r.w. stops at k-th step} \mid \text{the r.w. starts from } x_u, \text{ and stops at class '+'}) \\
&= \sum_{k=1}^{\infty} k \cdot \frac{\mathbb{P}(\text{the r.w. starts from } x_u \text{ and stops at k-th step and hits the '+' class})}{\mathbb{P}(\text{the r.w. starts from } x_u, \text{ and stops at class '+'})} \\
&= \frac{\sum_{k=1}^{\infty} k \cdot \mathbf{1}_u^T P_{uu}^{k-1} P_{u+}\mathbf{1}}{\sum_{k=1}^{\infty} \mathbf{1}_u^T P_{uu}^{k-1} P_{u+}\mathbf{1}} \\
&= \frac{\mathbf{1}_u^T (I - P_{uu})^{-2} P_{u+}\mathbf{1}}{\mathbf{1}_u^T (I - P_{uu})^{-1} P_{u+}\mathbf{1}} \qquad \text{where} \quad \mathbf{1_u} = (0, 0, \ldots, \underbrace{1}_{\text{u-th}}, 0, \ldots, 0)^T \qquad \square
\end{aligned}$$

This result shows a way of computing the conditional expected hitting times by a few matrix operations, and it also forms the basis of the proposed active learning algorithm.

4 Active Learning Algorithm

The two main components of an *Active Learning System* is a classifier and a criterion to choose the next batch of most informative unlabeled instances to query. Notice that these two steps are not independent, the classifier determines the set of most informative points. Thus it is important to have both the classifier and the criterion for choosing the unlabeled points to be defined under the same framework. Under the random walk framework, first we compute the distance vectors of the unlabeled points $d(X_u, +)$ and $d(X_u, -)$, then the classification rule follows naturally as

$$f(x_u) = \begin{cases} +1 & \text{if } d(x_u, +) > d(x_u, -) \\ -1 & \text{otherwise} \end{cases} \quad (5)$$

After training the classifier on the labeled set, we aim to find the unlabeled instances whose labels are hardest to predict using the current classifier. Intuitively points which are almost same distance away from both class '+' and class '-' are the most uncertain ones. Thus labeling them would provide the maximum information about the underlying structure of the data. We propose the measure of informativeness of a point as follows

$$s(x_u) = \frac{|d(x_u, +) - d(x_u, -)|}{d(x_u, +) + d(x_u, -)} \tag{6}$$

The reason we choose the relative difference instead of the absolute difference $|d(x_u, +) - d(x_u, -)|$ is to encourage the selection of points away from the cloud of the labeled instances, i.e. to enhance diversity in the labeled instance space. For example, if we have two unlabeled nodes x and y with $(d(x, +), d(x, -)) = (1.1, 1.2)$ and $(d(y, +), d(y, -)) = (2.1, 2.2)$. Notice that the two nodes have the same absolute difference of distances 0.1, however node y is considered to be more informative since it stays further from the labeled data cloud. Finally, the batch of unlabeled nodes with the smallest scores are chosen to query.

Table 1: Pseudocode of the proposed Active Learning Algorithm

1. Input: $\{(X_l, Y_l),\ X_u\}$, Batchsize, b = k

2. Construct the graph $G = (V, E, W)$.

3. Iterate: t = 1 to T

- Compute the distance vectors $d(X_u, +)$ and $d(X_u, -)$ in (4) using the current set of labeled and unlabeled data (X_l^t, Y_l^t, X_u^t).
- Compute the score vector $s(X_u)$ in (6), choose the k unlabeled nodes with the smallest scores (s) to query.
- Update the sets (X_l^{t+1}, Y_l^{t+1}), X_u^{t+1} by adding the newly labeled instances to the labeled data set.

4. Stop: Certain budget constraint is met.

To understand the algorithm better, we try to provide some intuitive insights about the technical details of the proposed algorithm in the following subsections and follow it up with an extensive simulation study to illustrate those features in the next section. We always use the Gaussian kernel to construct the graph.

4.1 Separability of the training data and choice of bandwidth parameter σ

The bandwidth parameter σ in the Gaussian kernel plays a very important role in the performance of the proposed method. Recall that for each unlabeled node x_u, the transition probability to any node x_v is proportional to $exp(-\frac{\|x_u - x_v\|^2}{2\sigma^2})$. Mathematically, the transition matrix $\mathbf{P}$ has the following properties

$$\lim_{\sigma \to \infty} \mathbf{P} = \begin{pmatrix} \frac{1}{n} & \frac{1}{n} & \dots & \frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n} & \dots & \frac{1}{n} \end{pmatrix} \quad ; \quad \lim_{\sigma \to 0} \mathbf{P} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} \tag{7}$$

Thus, a large σ would imply that it is equally likely for the random walk to make transition to any other node, whereas a small σ would mean that transitions can only occur within the very small neighborhood around x_u. Hence if the data is separable, a small choice of σ would be beneficial even if the boundary of each class is very complex, since it restricts the random walk to stay within one class and makes it highly unlikely to hit the other class which is further away. However, if the data overlaps a lot between the two classes, then a small value of σ would tend to cause confusion between moving to the right class and the wrong class, since even in a small neighborhood, data from both classes are present. The proposed algorithm tends to work well on data set with high or moderate separability while may have poor performance on the highly overlapped data.

4.2 The proportion of each class in the training sample

Consider the case where the proportion of class '+' in the training sample is much higher than the class '-', then we expect more '+' points in the labeled set as well. The random walk would tend to hit the '+' class more often than the '-' class, as a result more unlabeld points would be misclassified even if the σ is chosen correctly. Thus the algorithm performs poorly as the ratio of labeled instances of the two classes goes to extremes. We illustrate this through a simulated example in the next section.

4.3 Stopping criterion

Usually in active learning the stopping criterion is decided by some budget constraint. That is, we continue querying new unlabeled points until a certain percentage of the total data set is labeled, and the objective is to reduce the test error rate by cleverly picking the points to label. We have experimented with a different approach here. At each iteration the algorithm produces a classification of the data set. Following the ideas from unsupervised learning set-up we define the stopping criterion to be $\text{Err} < \epsilon$, where

$$\text{Err} = \{\#\text{points whose predicted label changes from the previous step to the current step}\} \tag{8}$$

$\epsilon > 0$ is some pre-defined small number. With the proposed algorithm, often times we see that the misclassification error rate goes down in a non-monotone fashion, i.e. points might switch predicted labels in a sequence $'+' \to' -' \to' +'$, thus the misclassification error rate would remain more or less the same hence would not be a good indicator for the convergence, whereas the proposed measure would be able to detect these cases.

5 Simulation Study

In this section we present the results of applying the proposed method two simulated data sets. One is the much documented "Spiral" data set for the classification problem and the second one is a bivariate Gaussian mixture model. The main goal is to illustrate different aspects of the proposed algorithm and to assess its performance under different combination of tuning parameters (e.g. bandwidth σ, proportion of the two classes, batch-size, stopping criterion). Also we compare the performance of the proposed algorithm against the naïve method where we choose the batch at each step randomly.

5.1 Effect of overlap

The underlying data set is a mixture of two Gaussian populations with 200 points in each class, we apply the algorithm where the convergence is determined by the proposed rate in the previous section (8). We start with the means being wide apart and let them move closer so that the overlap increases. The following table gives us the final misclassification error rate. It is easy to see that the proposed algorithm does not do

Table 2: Performance comparison on gaussian data with different degrees of overlap

Euclidean distance between the means	5	4	3	2	1
No. of iterations	3	3.7	7.6	12	18.2
Avg misclassification error rate	0.0008	0.0151	0.0530	0.1347	0.2927

very well both in terms of error rate and time to converge as the overlap increases which is common to every learning algorithm.

5.2 Bivariate Spiral Data

We have 180 points in each class arranged in two intertwined spiral. This is a generally hard classification problem, we start with an labeled set of randomly chosen 5 points from each class, with batch size of 4 and $\sigma = 0.05$. The Misclassification error rate is computed on a test data set. Below we show the result of first few iterations.

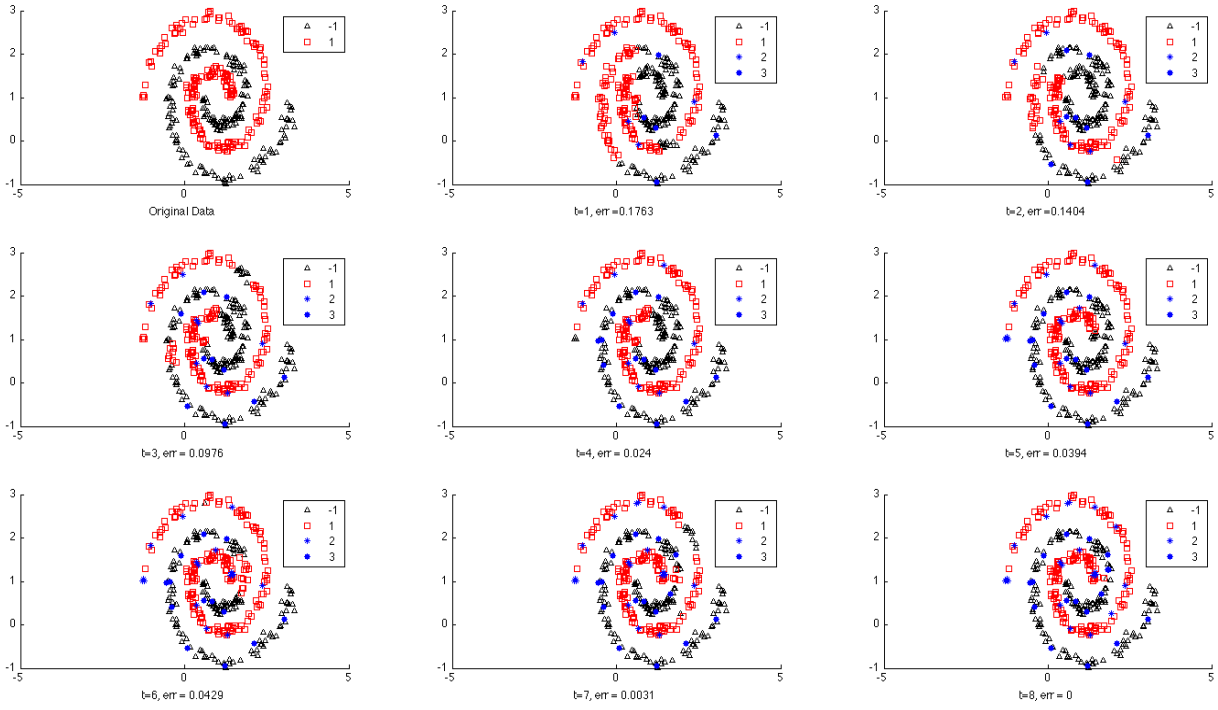

Figure 1: Active Learning algorithm on the spiral data

In the $(1,1)$-th figure $\square$ and the $\triangle$ denote the observations from the two classes. From the 2-nd subplot onwards we actually plot the predicted labels and the $*$'s denote the batch of points queried at each iteration. We see that within 8 iteration misclassification error rate drops down almost to 0, and also the non-monotonic behavior is visible. The following figure shows a comparison between the proposed measure of error (8) against the misclassification error rate

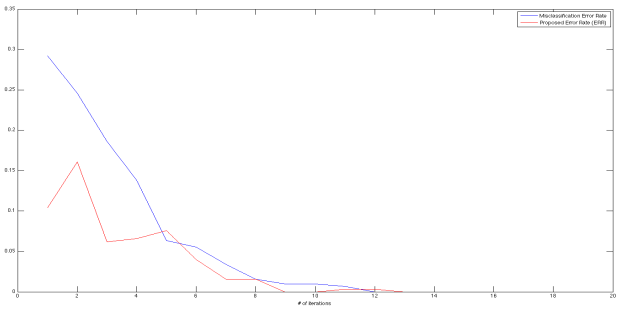

Figure 2: Active Learning algorithm on the spiral data

It shows that the misclassification can decrease even though many points changes their predicted label which is reflected by the jumps in the red curve and hence a stopping criterion based only on the misclassification error rate may not be a very good measure.

5.2.1 Effect of Proportion of the two classes in the training sample

Once again the underlying data is generated from the bivariate spiral model. Size of the data set is fixed at 400. And the proportion of the '+' class s varied from 0.1 to 0.9. As before the size of the initial labeled data set is 10 with 5 from each group and the batch size is 4. We plot the estimated misclassification error each averaged over 10 simulation runs.

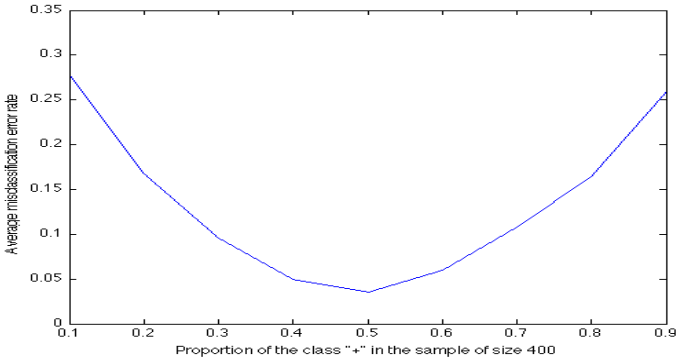

Figure 3: Active Learning algorithm on the spiral data

As expected we find that the algorithm performs the best when the proportion of the two classes in the training sample are equal i.e. 0.5 each. On the other hand as we move away from the equal proportion on the either side the average misclassification error rates increase.

5.2.2 Comparing the proposed method with the naïve method of choosing the batches randomly

We try to illustrate that the active learning algorithm is actually selecting points cleverly which improves both performance and convergence. We compare it against the naïve active learner where at each step the batch is chosen randomly. The bandwidth parameter σ vary between $(0.01, 0.25)$ since that appeared to be the active region for the bandwidth parameter for this particular data set. We also vary the stopping criterion from 10% of the entire data set to 50% of the total data set. All the results are averaged over 10 simulation run at each setting.

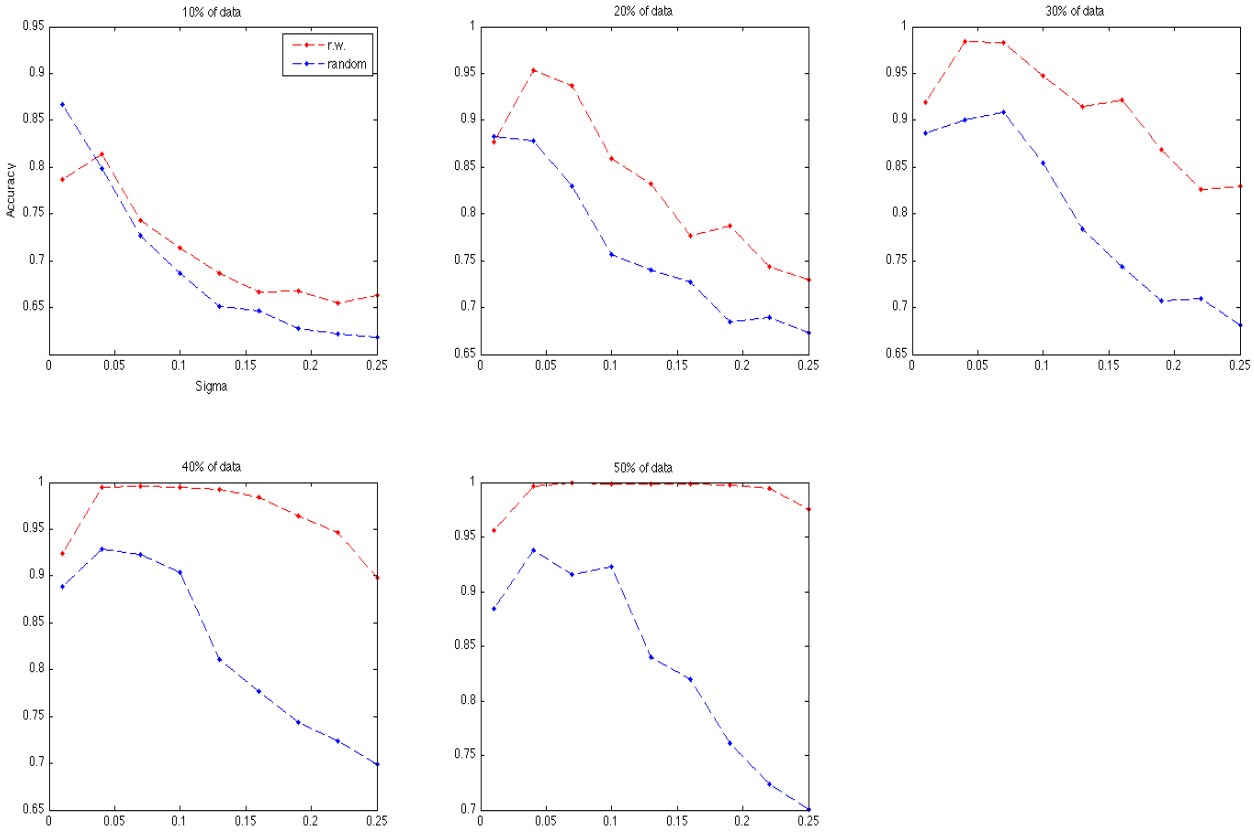

Figure 4: Comparison between the proposed learning algorithm and selecting the batches randomly

On the horizontal axis we have σ and on the vertical axis we plot the *accuracy*=

$1 - MER$. The blue line corresponds to the naïve method of choosing the batches randomly and the red line corresponds to the proposed algorithm where we choose the most informative set of points using the scoring function in (6).

In almost every situations the proposed algorithm outperforms the naïve algorithm, though there are some interesting patterns which can be explained intuitively. When σ is well-tuned $\sigma \in (0.05, 0.1)$ we find that the difference between the naïve approach and the proposed method is the least, and as σ goes further from that region the performance gap increases. This means if the performance of the classifier is poor we can gain a lot by choosing the most informative points cleverly. As the stopping condition is relaxed to allow more and more points to labeled the performance of the proposed algorithm gets better and better.

6 Comments and Discussions

In this project, we have proposed a new approach to active learning using random walks on the induced graph. We studied its theoretical properties and applied it to some simulated problems where it shows attractive performance as well as certain drawbacks. Our results show that the proposed algorithm works very well on separable data set with similar class proportions, even if the true class boundaries are very complex. We also address the role of the bandwidth parameter σ in constructing the graph which is also important in determining the performance of algorithm.

At each iteration, the algorithm involves the inversion of a matrix whose size is the number of unlabeled data points $O(n)$, and the inversion such a matrix has computational cost $O(n^3)$; with the number of iterations $O(n)$, the final computational cost is $O(n^4)$. Thus if the sample size is too large, this algorithm might not be practical since the inversion of a large n-dimensional matrix is computationally expensive. Based on the simulated experiments for sample sizes below 1000 the algorithm works satisfactorily fast.

The proposed algorithm is more readily adaptable for continuous data since it depends on the notion of distance between the instances, so to extend it to handle categorical or mixed data, the definition of a proper distance is required. Also, we need to experiment with real-life data and high-dimensional data to assess its performance.

7 References

[1] G.Schohn and D.Cohn. Less is more: Active learning with support vector machines. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.

[2] Z.xu, K.Yu, V.tresp, X.Xu, and J.Wang. Representative sampling for text classification using support vector machines. In *Proceedings of the 25th European Conference on Information Retrieval Research*, 2003.

[3]Y.Guo and D.Schuurmans. Discriminative Batch Mode Active Learning. In *Proceedings of Advances in Neural Information Processing Systems*, 2007.

[4] X. Zhu and Z. Ghahramani. Learning from Labeled and Unlabeled Data with Label Propagation. In *Technical Report CMU-CALD-02-107, Carnegie Mellon University*, 2002.

[5] G. Camps-Valls, T. Bandos, and D. Zhou. Semi-supervised graph-based hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing, Volume 45, Issue 10, Oct. 2007, pp. 3044 - 3054.*