Gaussian Process and SVM Active Learning Using Manifold-Preserving Graph Reduction

Shiliang Sun

Shanghai Key Laboratory of Multidimensional Information Processing, Department of Computer Science and Technology, East China Normal University, Shanghai, China.

(Joint work with Z. Hussain, J. Shawe-Taylor, J. Zhou)

CIKM 2014 Workshop on Interactive Mining for Big Data

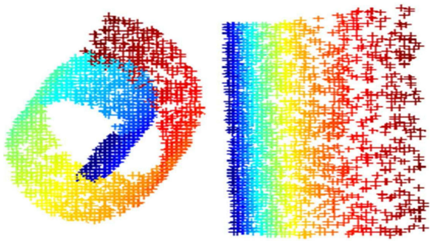
Outline

Motivation for MPGR

- Data lying in a high-dimensional space can often be assumed to be intrinsically of low dimensionality.

Motivation for MPGR

- Data lying in a high-dimensional space can often be assumed to be intrinsically of low dimensionality. That is, data can be well characterized by far fewer parameters or degrees of freedom than the actual ambient representation.
  $\Rightarrow$ Manifold Learning



Motivation for MPGR

- Manifold learning has been an active research topic during the past 15 years, with a variety of successful applications such as nonlinear dimensionality reduction, and semi-supervised learning.
- A usual procedure includes constructing a weighted graph using all the training examples and then performing learning based on the graph. However, it has two shortcomings:

Motivation for MPGR

- Manifold learning has been an active research topic during the past 15 years, with a variety of successful applications such as nonlinear dimensionality reduction, and semi-supervised learning.
- A usual procedure includes constructing a weighted graph using all the training examples and then performing learning based on the graph. However, it has two shortcomings:
    1. Possible outliers and noisy points, likely to damage the manifold structure, are retained.
    2. The evaluation of predictors learned from the graph for new examples can be time-consuming if the predictors involve computations on all the examples in the graph.

Motivation for MPGR

- Although some methods such as random sampling or k-means clustering can be used to reduce the size of the graph, they have no guarantees of preserving the manifold structure or effectively removing outliers and noisy examples.
- In particular, the k-means method is sensitive to outliers, and time-consuming when the number of clusters is large.

Motivation for MPGR

- Although some methods such as random sampling or k-means clustering can be used to reduce the size of the graph, they have no guarantees of preserving the manifold structure or effectively removing outliers and noisy examples.
- In particular, the k-means method is sensitive to outliers, and time-consuming when the number of clusters is large.
- To overcome these two shortcomings, we propose the idea of manifold-preserving sparse graphs and the manifold-preserving graph reduction algorithm.

Manifold-Preserving Sparse Graphs

Since graphs are deemed as a discrete representation of manifolds, the definition of manifold-preserving sparse graphs naturally corresponds to sparsified manifolds.

Manifold-Preserving Sparse Graphs

Since graphs are deemed as a discrete representation of manifolds, the definition of manifold-preserving sparse graphs naturally corresponds to sparsified manifolds.

Definition (Sparse graph candidates)

Given a graph $G(V, E, W)$ corresponding to a manifold with vertex set $V = \{x_1, \ldots, x_m\}$, edge set E, and symmetric weight matrix W, the graph $G_c(V_c, E_c, W_c)$ with V_c, E_c and W_c being respectively subsets of V, E, and W is called a sparse graph candidate of the original graph G.

In a graph, a large weight represents a high similarity.

Manifold-Preserving Sparse Graphs

Then, we give the definition of graph distance as a characterization of the loss between two graphs G, G_c.

Definition (Graph distance)

Given a graph $G(V, E, W)$ and its sparse graph candidate $G_c(V_c, E_c, W_c)$, the graph distance between G and G_c is the loss of weights from W to W_c, that is $\sum_{i \in V \setminus V_c, j \in V_c} W_{ij}$, where $V \setminus V_c$ denotes the set of vertices not included in V_c.

Manifold-Preserving Sparse Graphs

Then, we give the definition of graph distance as a characterization of the loss between two graphs G, G_c.

Definition (Graph distance)

Given a graph $G(V, E, W)$ and its sparse graph candidate $G_c(V_c, E_c, W_c)$, the graph distance between G and G_c is the loss of weights from W to W_c, that is $\sum_{i \in V \setminus V_c, j \in V_c} W_{ij}$, where $V \setminus V_c$ denotes the set of vertices not included in V_c.

Seeking manifold-preserving sparse graphs is to find sparse graph candidates with manifold-preserving properties, i.e., a point outside of the sparse graphs should have a high connectivity with a point retained.

Manifold-Preserving Sparse Graphs

Definition: Manifold-preserving sparse graphs

- Given a graph G with m vertices and the $\sharp$ of vertices in the desired sparse graphs, the manifold-preserving sparse graphs G_s are those candidates having a high space connectivity with G.

Manifold-Preserving Sparse Graphs

Definition: Manifold-preserving sparse graphs

- Given a graph G with m vertices and the $\sharp$ of vertices in the desired sparse graphs, the manifold-preserving sparse graphs G_s are those candidates having a high space connectivity with G.
- By a high space connectivity, we mean that for a candidate with t vertices the quantity $\frac{1}{m-t}\sum_{i=t+1}^{m}(\max_{j=1,\dots,t} W_{ij})$ is maximized, where W is the weight matrix of G, and indices $1, \dots, t$ correspond to an arbitrary ordering of the vertices in the sparse graph candidate.

The MPGR Algorithm

- Given the number of retained vertices in sparse graphs, the problem of exactly seeking manifold-preserving sparse graphs is NP-hard.
- Here, we give a simple and efficient greedy algorithm to construct such sparse graphs. Due to its simplicity and high efficiency, applying this algorithm to large-scale data would be quite straightforward.

The MPGR Algorithm

- Given the number of retained vertices in sparse graphs, the problem of exactly seeking manifold-preserving sparse graphs is NP-hard.
- Here, we give a simple and efficient greedy algorithm to construct such sparse graphs. Due to its simplicity and high efficiency, applying this algorithm to large-scale data would be quite straightforward.
- Define the degree $d(i)$ associated with vertex i to be $d(i) = \sum_{i \sim j} W_{ij}$ where $i \sim j$ means (i, j) are connected by an edge (if two vertices are not linked, their similarity is regarded as zero).

The MPGR Algorithm: Pseudo Code

```
Input: Graph $G(V, E, W)$ with $m$ vertices;
       $t$ for the number of the vertices in the desired
         sparse graph $G_s$.
1: for $j = 1, \ldots, t$
2:     compute degree $d(i)$ $(i = 1, \ldots, m - j + 1)$
3:     pick one vertex $v$ with the maximum degree
4:     remove $v$ and associated edges from $G$; add $v$ to
         $G_s$
5: end for
Output: Manifold-preserving sparse graph $G_s$ with $t$
          vertices.
```

Efficient? Applicability to Big Data

Suppose t vertices are sought from an original graph with m vertices.
Let d_E be the maximum number of edges linked to a vertex in the original graph.

Efficient? Applicability to Big Data

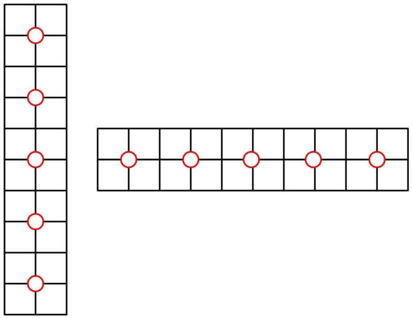
Suppose t vertices are sought from an original graph with m vertices.
Let d_E be the maximum number of edges linked to a vertex in the original graph.
The computational complexity of our algorithm is less than

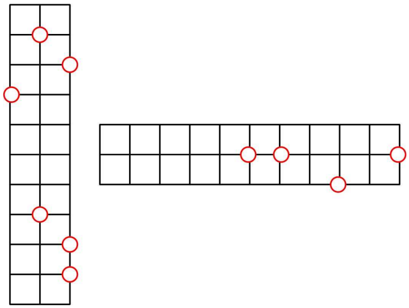
$$O\left[d_E\Big(m + (m-1) + \ldots + (m-t+1)\Big)\right] = O(d_E mt),$$

which is respectively linear with respect to d_E, m and t.
Thus the manifold-preserving graph reduction algorithm is very efficient.

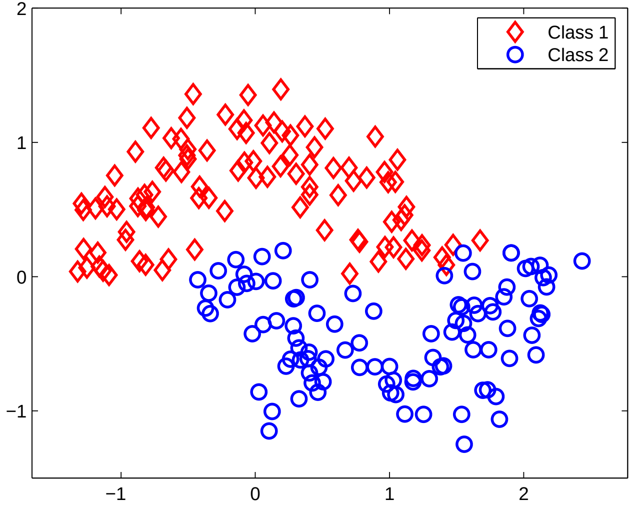
The MPGR Algorithm: Case Studies (1)

Points retained in sparse manifolds with different methods.

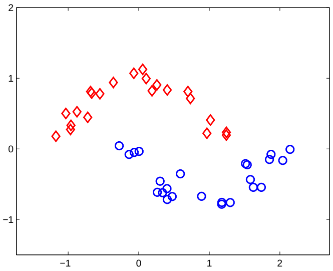

(a) MPGR


(b) Random sampling

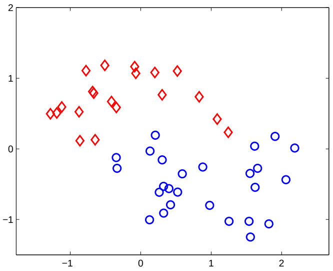
The MPGR Algorithm: Case Studies (2)



The MPGR Algorithm: Case Studies (2)



(a) MPGR



(b) Random sampling

Outline

Motivation

- In machine learning, labeled examples are very useful to offer effective discriminative information. However, although people can get large numbers of unlabeled examples easily, it usually needs much manual labor to label them, which can be expensive and time-consuming.

Motivation

- In machine learning, labeled examples are very useful to offer effective discriminative information. However, although people can get large numbers of unlabeled examples easily, it usually needs much manual labor to label them, which can be expensive and time-consuming.
- Thus, learning from both labeled and unlabeled data has drawn more and more attentions.
  Active learning is one of the effective strategies to solve this kind of problem.

Active Learning

- Active learning, sometimes called "experimental design", can actively query the user for labels. In other words, unlabeled examples that are considered the most informative and important can be optimally selected for human labeling.

Active Learning

- Active learning, sometimes called “experimental design”, can actively query the user for labels. In other words, unlabeled examples that are considered the most informative and important can be optimally selected for human labeling.
- Compared with supervised learning algorithms, active learning can perform as effectively as a regular supervised learning framework but with fewer labels by interactive queries.

Categorization

Mainly three kinds of active learning strategies.

1. The first one relies on the estimation of the posterior probability distribution of the classes, e.g., Gaussian process active learning.

Categorization

Mainly three kinds of active learning strategies.

1. The first one relies on the estimation of the posterior probability distribution of the classes, e.g., Gaussian process active learning.
2. The second class is the margin sampling (MS) strategy. MS selects for labeling the unlabeled data which are the nearest to the classification margin of classifiers, e.g., support vector machine (SVM) active learning.

Categorization

Mainly three kinds of active learning strategies.

1. The first one relies on the estimation of the posterior probability distribution of the classes, e.g., Gaussian process active learning.
2. The second class is the margin sampling (MS) strategy. MS selects for labeling the unlabeled data which are the nearest to the classification margin of classifiers, e.g., support vector machine (SVM) active learning.
3. The third is based on the query-by-committee sampling (QBC). QBC selects for labeling the unlabeled examples whose classification is the most uncertain among the committee member classifiers.

Categorization

- Our focus will be the first two kinds of active learning strategies.

Categorization

- Our focus will be the first two kinds of active learning strategies.
- However, we note that there is another different categorization of active learning methods according to the number of feature sets used to represent data.
  1. Single-view active learning
  2. Multi-view active learning

  Reference: S. Sun. A survey of multi-view machine learning. Neural Computing and Applications, 2013.

Outline

Gaussian Processes ($\rightarrow$GPAL$\rightarrow$GPMAL)

- Gaussian processes provide probabilistic prediction estimates and are well-suited for active learning.

Gaussian Processes (→GPAL→GPMAL)

- Gaussian processes provide probabilistic prediction estimates and are well-suited for active learning.
- A Gaussian process is a stochastic process specified by its mean and covariance function. Given a data set with N examples $X = \{x_1, x_2, \ldots x_N\}$, the corresponding class labels are $\mathbf{t} = [t_1, t_2, \ldots t_N]^\top$ and latent variables are $Y = \{y_1, y_2, \ldots y_N\}$. The prior distribution is assumed to be Gaussian:

  $$p(Y|X, \theta) = N(Y|\mathbf{0}, K)$$

  where K is a kernel matrix parameterized by the hyperparameter θ.

Gaussian Processes

- The likelihood models the probabilistic relationship between the label t and the latent variable y. In this work we assume that t and y are related via a Gaussian noise model

$$p(t|y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(t-y)^2}{2\sigma^2}}$$

  where σ^2 is the noise model variance.
- Although the Gaussian noise model is originally developed for regression, it has also been proved effective for classification.

Gaussian Processes

- When there is a new point x_u, the posterior $P(y_u|X, \mathbf{t})$ over the latent label y_u is also a Gaussian which can be written as:

  $$P(y_u|X, \mathbf{t}) \sim N(Y_u, \Sigma_u),$$

  where

  $$\begin{aligned} Y_u &= K_t(x_u)^\top(\sigma^2 I + K)^{-1}\mathbf{t}, \\ \Sigma_u &= k(x_u, x_u) - K_t(x_u)^\top(\sigma^2 I + K)^{-1}K_t(x_u). \end{aligned}$$

Gaussian Processes

- When there is a new point x_u, the posterior $P(y_u|X, \mathbf{t})$ over the latent label y_u is also a Gaussian which can be written as:

$$P(y_u|X, \mathbf{t}) \sim N(Y_u, \Sigma_u),$$

  where

$$\begin{aligned} Y_u &= K_t(x_u)^\top(\sigma^2 I + K)^{-1}\mathbf{t}, \\ \Sigma_u &= k(x_u, x_u) - K_t(x_u)^\top(\sigma^2 I + K)^{-1}K_t(x_u). \end{aligned}$$

- The predictive distribution over the unknown label t_u is also a Gaussian: $P(t_u|X, \mathbf{t}) \sim N(Y_u, \Sigma_u + \sigma^2)$.

Active Learning of Gaussian Processes

There are usually three active learning criteria for example selection in Gaussian processes.

1. $x_{a\ell} = \arg\min_{x_u \in X_u} |Y_u|$. Nearly symmetric distribution for $t_u = \pm 1$.

Active Learning of Gaussian Processes

There are usually three active learning criteria for example selection in Gaussian processes.

1. $x_{a\ell} = \arg\min_{x_u \in X_u} |Y_u|$. Nearly symmetric distribution for $t_u = \pm 1$.
2. $x_{a\ell} = \arg\max_{x_u \in X_u} \Sigma_u$. Uncertainty.

Active Learning of Gaussian Processes

There are usually three active learning criteria for example selection in Gaussian processes.

1. $x_{a\ell} = \arg\min_{x_u \in X_u} |Y_u|$. Nearly symmetric distribution for $t_u = \pm 1$.
2. $x_{a\ell} = \arg\max_{x_u \in X_u} \Sigma_u$. Uncertainty.
3. Exploiting both the posterior mean as well as the posterior variance:

$$x_{a\ell} = \arg\min_{x_u \in X_u} \frac{|Y_u|}{\sqrt{\Sigma_u + \sigma^2}}. \quad \checkmark$$

GPAL Algorithm

Algorithm 1: Active Learning of Gaussian Processes (GPAL)

Input: Labeled set T, unlabeled pool U, selected batch size Q
Output: Error rates of classification

```
1  for p=1 to maximum number of iterations do
2      Select the most informative $Q$ points by the following steps:
3      while $q=1$ to $Q$ do
4          Select the most uncertain point (according to Eq. (7))
5          Move the point from $U$ to $T$:
6          $T = T \cup (x_{a\ell}, label(x_{a\ell})), U = U - x_{a\ell}$
7      end
8      Train a classifier with the new $T$;
9      Calculate the error rate on the new $U$;
10 end
```

Active Learning with MPGR

- There are some shortcomings in traditional active learning methods, such as not exploiting the space connectivity and not considering spatial diversity among the examples.
- In order to overcome these shortcomings, we apply the MPGR algorithm to the original active learning method of Gaussian processes. $\Rightarrow$ GPMAL.

Active Learning with MPGR

- There are some shortcomings in traditional active learning methods, such as not exploiting the space connectivity and not considering spatial diversity among the examples.
- In order to overcome these shortcomings, we apply the MPGR algorithm to the original active learning method of Gaussian processes. $\Rightarrow$ GPMAL.
- GPMAL tends to select globally representative examples (the examples that are closer to surrounding examples are deemed as representative and having important information) and examples with high space connectivity.

GPMAL Algorithm

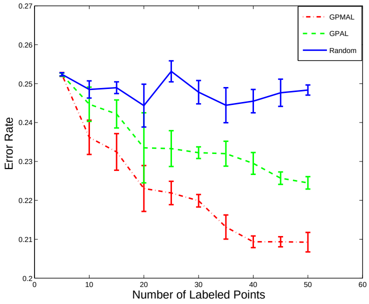
Algorithm 2: Active Learning of Gaussian Processes with MPGR (GPMAL)

```
Input: Labeled set $T$, unlabeled pool $U$, selected batch size $Q$
Output: Error rates of classification
for $p$=1 to maximum number of iterations do
    Construct graph $G$ with all the unlabeled points;
    Select a subset $T_s$ with $m$ points ($m > Q$) by the following steps:
    while $i$=1...$m$ do
        Compute degree $d(j)$ ($j$=1...$m - i + 1$)
        Pick a point $h$ with the maximum degree, add $h$ to $T_s$
        Remove $h$ and associated edges from $G$
    end
    Reselect $Q$ points from $T_s$ by GPAL;
    Add $Q$ points to $T$ and correspondingly remove these points from $U$;
    Train a classifier with the new $T$;
    Calculate the error rate on the new $U$;
end
```

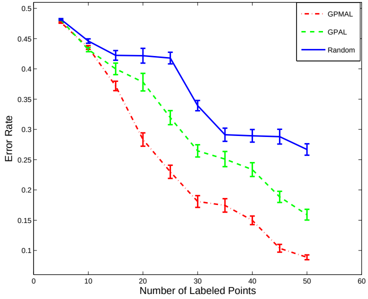
Experiments

- Experiments are performed on six data sets including binary and multiclass problems.
- Parameters are selected through five-fold cross-validation on the training set and experiments are conducted ten times on each data set.
- GPMAL gets better results.

Illustration of Some Results (1)

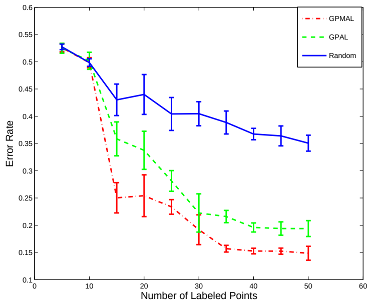


(a) BTSC data

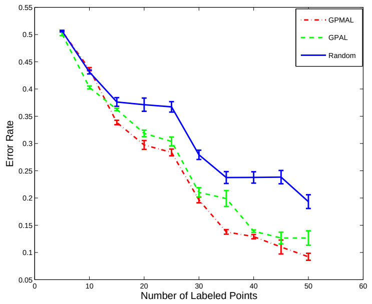


(b) MP data

Illustration of Some Results (2)



(c) Artificial data



(d) VC data

Quantitative Comparison

The error rates for all five real data sets with the maximum numbers of labeled points.

The three rows correspond to GPMAL, GPAL, and random selection, respectively.

BTSC	**MP**	**VC**	**BS**	**CCG**
21.20 ± 0.21	8.87 ± 0.40	9.55 ± 1.40	2.81 ± 1.01	11.83 ± 0.38
22.45 ± 0.16	15.91 ± 0.81	12.65 ± 1.32	4.47 ± 1.32	14.21 ± 0.46
24.84 ± 0.13	26.68 ± 0.94	19.33 ± 1.26	13.66 ± 1.26	21.91 ± 0.41

Outline

SVM Active Learning: Margin Sampling

- An SVM uses a linear optimal hyperplane to discriminate classes, which is induced from the maximum margin principle between two classes.

SVM Active Learning: Margin Sampling

- An SVM uses a linear optimal hyperplane to discriminate classes, which is induced from the maximum margin principle between two classes.
- Consider a binary problem. The distance of a sample to the decision boundary is given by

$$f(q_j) = \sum_{i=1}^{m} \alpha_i y_i K(x_i, q_j) + b,$$

  where K is a kernel matrix, which defines the similarity between the candidate q_j and the support vector x_i.

Margin Sampling (MS) and Improved MS

- In MS, the candidate selected into the training set is the one respecting the condition

$$x' = \arg\min_{q_j \in U} |f(q_j)|.$$

Margin Sampling (MS) and Improved MS

- In MS, the candidate selected into the training set is the one respecting the condition

$$x' = \arg\min_{q_j \in U} |f(q_j)|.$$

- MS is optimal only when a single candidate is chosen per iteration. When selecting several examples simultaneously, the problem of oversampling on a small area is unavoidable.

Margin Sampling (MS) and Improved MS

- In MS, the candidate selected into the training set is the one respecting the condition

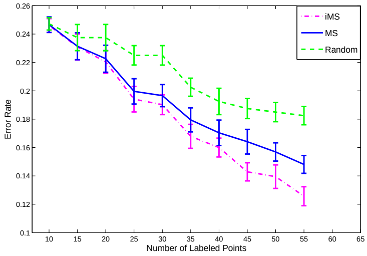
$$x' = \arg\min_{q_j \in U} |f(q_j)|.$$

- MS is optimal only when a single candidate is chosen per iteration. When selecting several examples simultaneously, the problem of oversampling on a small area is unavoidable.
- In order to remedy the problem, we propose an improvement of MS (iMS) by considering the space connectivity and the distribution of the unlabeled candidates with MPGR.

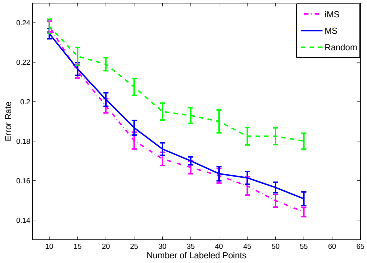
Experiments

- Experiments are performed on three real data sets including binary and multiclass problems.
- Parameters are selected through five-fold cross-validation on the training set and experiments are conducted 15 times on each data set.
- iMS gets better results.

Illustration of Some Results (1)

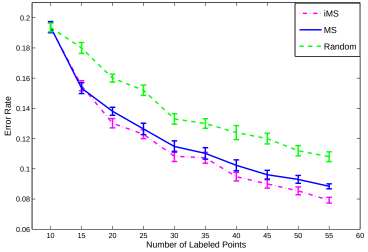


(a) Ionosphere data

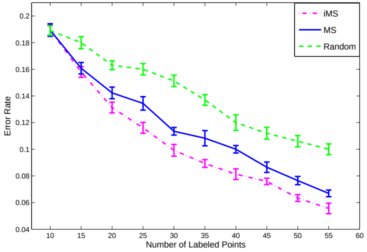


(b) VC data

Illustration of Some Results (2)



(c) Left vs rest



(d) Right vs rest

Outline

Summary

- The MPGR algorithm was originally proposed for sparse semi-supervised learning.

Summary

- The MPGR algorithm was originally proposed for sparse semi-supervised learning.
- Here we have further applied it another different context active learning.
- Both Gaussian process active learning and SVM active learning were considered.

Summary

- The MPGR algorithm was originally proposed for sparse semi-supervised learning.
- Here we have further applied it another different context active learning.
- Both Gaussian process active learning and SVM active learning were considered.
- As shown in the experiments, the performance of the new active learning methods is very good.

Discussion

Some interesting future work:

- If the GP assumption is violated, more complex models are desirable, e.g., mixtures of GPs.

Discussion

Some interesting future work:

- If the GP assumption is violated, more complex models are desirable, e.g., mixtures of GPs.
- A comprehensive comparison between the proposed GPMAL and iMS is needed.

The End

References:

1. S. Sun, Z. Hussain, J. Shawe-Taylor. Manifold-preserving graph reduction for sparse semi-supervised learning. Neurocomputing, 2014.
2. J. Zhou, S. Sun. Active learning of Gaussian processes with manifold-preserving graph reduction. Neural Computing and Applications, 2014.
3. J. Zhou, S. Sun. Improved margin sampling for active learning. CCPR, 2014.
4. J. Shawe-Taylor, S. Sun. Kernel methods and support vector machines. Book Chapter for E-Reference Signal Processing, Elsevier, 2013.