Noname manuscript No.
(will be inserted by the editor)

Active Learning of Gaussian Processes with Manifold-Preserving Graph Reduction

Jin Zhou · Shiliang Sun

Received: date / Accepted: date

Abstract As a recently proposed machine learning method, active learning of Gaussian processes can effectively use a small number of labeled examples to train a classifier, which in turn is used to select the most informative examples from unlabeled data for manual labeling. However, in the process of example selection, active learning usually need consider all the unlabeled data without exploiting the structural space connectivity among them. This will decrease the classification accuracy to some extent since the selected points may not be the most informative. To overcome this shortcoming, in this paper we present a method which applies the manifold-preserving graph reduction (MPGR) algorithm to the traditional active learning method of Gaussian processes. MPGR is a simple and efficient example sparsification algorithm which can construct a subset to represent the global structure and simultaneously eliminate the influence of noisy points and outliers. Thereby, when actively selecting examples to label, we just choose from the subset constructed by MPGR instead of the whole unlabeled data. We report experimental results on multiple data sets which demonstrate that our method obtains better classification performance compared with the original active learning method of Gaussian processes.

Keywords Manifold-preserving graph reduction · Space connectivity · Active learning · Gaussian process

J. Zhou
Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, China

S. Sun
Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, China
Tel.: +86-21-54345186
Fax: +86-21-54345119
E-mail: slsun@cs.ecnu.edu.cn

1 Introduction

In machine learning, labeled examples are very useful which can offer effective discriminative information. In many real-world problems, sufficient labeled examples are required to train a good classifier. Although people can get large numbers of unlabeled examples easily, it usually needs much manual labor to label them, which can be grueling, difficult or time-consuming. As a consequence, the number of available unlabeled examples are generally much larger than labeled ones. 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, sometimes called "experimental design", is a learning mechanism which 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 [19]. It is a process of guiding the sampling process by following some criteria to select those that can enhance the classification accuracy during the iteration from a large pool of unlabeled data. Compared with supervised learning algorithms, active learning can perform as efficiently as a regular supervised learning framework but with fewer labels by interactive queries [21]. Due to this interactive setting, the number of examples needed to be labeled can be much less. There are mainly three strategies used in active learning [22]. The first strategy of active learning methods is based on the query-by-committee sampling (QBC) [5,6,26]. The committee members are composed of some classification models. QBC selects for labeling the unlabeled examples whose classification is the most uncertain among the committee member classifiers. The second class is the margin sampling (MS) strategy [3,16]. MS selects for labeling the unlabeled data which is the nearest to the classification margin of classifiers. The third one relies on the estimation of the posterior probability distribution function of the classes [9]. It selects the examples for manual labeling based on the values of their posterior probabilities. In this paper, we mainly focus on the last strategy using the probabilistic model of Gaussian processes (GP), since they can provide probabilistic prediction estimates at unlabeled examples. The uncertainty model provided by GP can be well-suited for active learning. In addition, active learning of Gaussian processes (GPAL) has been successfully used for object categorization [11–13].

Despite the excellent performance of active learning algorithms, there are still some shortcomings that we should not omit. For example, in the process of active example selection, we usually need take all the unlabeled instances into account without considering the structural information and spatial diversity among them. This will lead to a result that in the same area there are more than one point to be selected, and thus it is possible to produce redundancy which can decrease the classification accuracy. In order to avoid the influence of noisy points and simultaneously consider the space connectivity among instances, we introduce an algorithm called manifold-preserving graph reduction (MPGR) [20] beyond the original active learning method. MPGR is a simple example sparsification algorithm which can effectively remove outliers and noisy points. By using MPGR, we can construct a subset which can represent the global manifold structure using fewer examples. This can eliminate the influence of noisy points and promote the example-selection quality of predictors. Learning with this kind of manifold assumption has been successfully applied to many machine learning tasks [1, 2,18,20,27].

The main contribution of this paper is that we introduce the MPGR algorithm to the original GPAL. We denote this new method which effectively exploits the manifold information as GPMAL. In our method, when selecting unlabeled points to label, it chooses from a subset constructed by MPGR instead of the whole unlabeled data set. This subset has more important discriminative information and excludes noisy points and outliers. We can see that the classification accuracy has improved a lot compared with the original active learning method.

The remainder of this paper proceeds as follow. In Section 2, we briefly review some background about GPAL. In Section 3, we describe our method which applies MPGR to the original active learning method. In Section 4, we show the experimental results on one artificial data set and five real data sets to demonstrate the effectiveness of our method. Finally, we provide concluding remarks in Section 5.

2 Background

In this section, we briefly review some basic knowledge related to active learning of Gaussian processes.

2.1 Active learning

Active learning is a process of guiding the sampling process by actively selecting and labeling the most informative candidates from a large pool of unlabeled examples. As a method of constructing an effective training set, the goal of active learning algorithms is to find informative examples which can enhance the classification accuracy of the model during the iteration, thereby reducing the size of the training set and improving the efficiency of the model within the limited time and resources. Instead of randomly picking unlabeled examples, active learning selects valuable examples according to some certain criteria. Through this, a predictor trained on a small set of well-chosen examples performs as effectively as a predictor trained on a larger number of randomly chosen examples [4, 14, 24].

As a topic of recent interest, a lot of active learning methods have been proposed for selecting unlabeled examples for tagging. As mentioned above, this paper is mainly focused on the third strategy to select unlabeled points, which can be used in both two-class [9] and multi-class problems [15]. Generally speaking, the posterior probability reflects the confidence level of the category an example belongs to. Take two-class classification (positive/negative) for example. If the posterior of being positive of an example is closer to 0.5, the example possesses more information, and it is more likely to be selected.

Compared with traditional supervised methods, active learning has the following advantages: It can handle large training data set, choose the discriminative points, and reduce the number of training data and artificially labeled examples. Active learning has been applied to many real-world applications, such as video classification and retrieval [7], medical image classification [8], cancer diagnosis [10], information extraction [17], and image classification and retrieval [25].

2.2 Gaussian processes

Here we briefly summarize Gaussian processes to facilitate the subsequent introduction of GPAL.

As mentioned above, Gaussian processes provide probabilistic prediction estimates and thus are well-suited for active learning. The Gaussian process model is a very flexible tool for data modeling, which can adapt itself to data through changing the involved parameters. When it is used for prediction, both the training and test data are assumed to be modeled by the same underlying process. The model parameters are fixed from the training data, which together with the training data will then be used to predict test data. However, in some cases when people know that the data don't have a normal distribution, they can use more complex models such as mixtures of Gaussian processes to model the data. A Gaussian process is a stochastic process specified by its mean and covariance function [14]. Given a data set with N examples $X = \{x_1, x_2, \dots x_N\}$, the corresponding class labels are $\mathbf{t} = [t_1, t_2, \dots t_N]^\top$ and latent variables are $Y = \{y_1, y_2, \dots y_N\}$. For simplicity, here we just discuss the two-class problem. Hence, the label t_i belongs to $\{1, -1\}$. As to multi-class classification, we can just use one-vs-rest to convert the multi-class problem to multiple two-class problems.

The prior distribution that defines the probabilistic relationship between the examples X and the latent variables Y is assumed to be Gaussian:

$$p(Y|X, \theta) = N(Y|\mathbf{0}, K) \tag{1}$$

with a zero mean and a covariance matrix K where K is a kernel matrix parameterized by the hyperparameter θ. 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. For regression, a Gaussian observation likelihood often uses a Gaussian noise model:

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

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, and its performance typically is comparable to the more complex probit and logit likelihood models used in classification problems [11]. For its simplicity and a closed form solution for iterations, we use the GP noise model in our experiments. Given the data points, the marginal likelihood will be:

$$P(\mathbf{t}|X) = N(\mathbf{t}|\mathbf{0}, K + \sigma^2 I). \tag{3}$$

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:

$$P(y_u|X, \mathbf{t}) \sim N(Y_u, \Sigma_u), \tag{4}$$

where

$$Y_u = K_t(x_u)^\top (\sigma^2 I + K)^{-1} \mathbf{t}, \tag{5}$$

$$\Sigma_u = k(x_u, x_u) - K_t(x_u)^\top (\sigma^2 I + K)^{-1} K_t(x_u). \tag{6}$$

Here, k is the covariance function and $K_t(x_u)$ is the vector of covariances between x_u and the training points, which is given by $K_t(x_u) = [k(x_u, x_1), ..., k(x_u, x_N)]^\top$. Since the Gaussian noise model links t_u to y_u, the predictive distribution over the unknown label t_u is also a Gaussian: $P(t_u|X, \mathbf{t}) \sim N(Y_u, \Sigma_u + \sigma^2)$.

2.3 Active learning of Gaussian processes

Considering a large pool of unlabeled data, the task of active learning of Gaussian processes is to actively query labels for the examples with the most uncertainty and then update training data by adding them to the existing labeled data set to train a classifier. Here we use the uncertainty sampling criterion. With the uncertainty estimates provided by GP, we should select the points which are the most uncertain (Take two-class classification problems for example, the posterior of being positive is nearest to 0.5).

Generally speaking, there are usually three active learning criteria for example selection in Gaussian processes [11]: one criterion is exploiting the distance from the classification boundary to identify points for active learning. Used with Gaussian processes classification models, it would be inclined to select the next point with the minimum posterior mean by examining the magnitude of the posterior mean $|Y_m|$ ($x_{a\ell} = \arg\min_{x_u \in X_u} |Y_m|$). The second criterion is considering the variances and selects the point that has the maximum variance ($x_{a\ell} = \arg\max_{x_u \in X_u} \Sigma_u$). However, the mean and variance are both important parameters of a Gaussian process model. If we just consider the posterior mean or variance in the process of selecting unlabeled points, the available information will be very limited. The third criterion is to consider their posterior mean and variance simultaneously, which includes more comprehensive information. Here, we follow an approach suggested in [12] to select the unlabeled points, exploiting both the posterior mean as well as the posterior variance:

$$x_{a\ell} = \arg\min_{x_u \in X_u} \frac{|Y_m|}{\sqrt{\Sigma_u + \sigma^2}}. \tag{7}$$

Note that the value $p(t_u \geqslant 0) = \psi(\frac{Y_m}{\sqrt{\Sigma_u + \sigma^2}})$, where $\psi(\cdot)$ denotes the cumulative distribution function of a standard normal Gaussian distribution $N(0,1)$. When selecting examples, we want to choose the points with the most uncertainty. It means the examples should have a value for $p(t_u \geqslant 0)$ which is nearest to 0.5 using the posterior probability for representation. That is equivalent to say, the value of $\frac{|Y_m|}{\sqrt{\Sigma_u + \sigma^2}}$ should be very close to zero. Therefore, we just select an example x_u which minimizes $\frac{|Y_m|}{\sqrt{\Sigma_u + \sigma^2}}$. The GPAL algorithm used in this paper is represented in Algorithm 1.

Algorithm 1: Active Learning of Gaussian Processes (GPAL)

```
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
    Select the most informative $Q$ points by the following steps:
    while $q$=1 to $Q$ do
        Select the most uncertain point (according to Eq. (7))
        Move the point from $U$ to $T$:
        $T = T \cup (x_{a\ell}, label(x_{a\ell})), U = U - x_{a\ell}$
    end
    Train a classifier with the new $T$;
    Calculate the error rate on the new $U$;
end
```

3 Our proposed approach

In order to perform active learning, people use the labeled examples to train a classifier, which is then used on unlabeled points to select the most informative examples for user labeling. But when selecting the unlabeled points, they usually consider all the points without considering the structural information and spatial diversity among them. This will lead to a result that in one small area there will be more than one point to be selected. However, points in this small area are likely to offer the same information, and thus there is no need to select all of them. More importantly, it will cause data redundancy which decreases the classification accuracy. To overcome this shortcoming, we apply a manifold-preserving graph reduction algorithm to the original active learning method.

3.1 MPGR

In machine learning, manifold assumption is an important assumption which indicates that examples in a small area should have similar property and their labels should be also similar. That is to say, similar inputs should have similar outputs. This assumption reflects local smoothness of the decision function which can alleviate the overfitting problems. However, there are several disadvantages in manifold learning methods. First, when constructing manifolds, they usually rely on a high-density data set, which causes high complexity in computation. Second, they are very sensitive to noisy points and outliers. To avoid these two shortcomings, we use a sparse manifold graph, which can be deemed as a discrete representation of the manifold. Sparse manifolds have significant advantages as follows: it can effectively eliminate the influence of outliers and noisy points and simultaneously accelerate the evaluation of predictors learned from the manifolds [20]. Manifold-preserving graph reduction (MPGR) is a simple but efficient graph reduction algorithm based on the manifold assumption.

Through the MPGR algorithm, we can construct a manifold-preserving sparse graph. Manifold-preserving properties mean that an example outside of the sparse graph should have a high space connectivity with an example retained in it. Given a graph composed of all unlabeled examples, the manifold-preserving sparse graphs are those sparse graph candidates which have a high space connectivity with the original graph [20]. The value of space connectivity is:

$$\frac{1}{M-m}\sum_{i=m+1}^{M}\left(\max_{j=1,\ldots,m} W_{ij}\right), \tag{8}$$

where M is the number of all unlabeled points, m is the number of points to be reserved, and W is the weight matrix. It has been proved that maximizing Eq. (8) can obtain a larger lower bound of the expected connectivity between the m points retained in sparse graph and $M-m$ points outside of it by using the McDiarmid's inequality [20]. This provides guarantees to obtain a good space connectivity. However, the problem of directly seeking manifold-preserving sparse graphs is NP-hard, and thus the MPGR algorithm was proposed to seek an approximation to maximizing the above equation.

For a graph, normally speaking, weight can measure the similarity of linked points, and a higher weight means linked examples are more similar. Here we introduce a definition of degree $d(i)$. We denote it to be $d(i)=\sum_{i\sim j} w_{ij}$, where $i \sim j$ means that example i is connected with example j (e.g., defined by the k-nearest-neighbor rule) and w_{ij} is their corresponding weight. If two examples are not linked, their weight is regarded as 0. Due to its simplicity, $d(i)$ is generally used as a criterion of constructing sparse graphs. The bigger $d(i)$ is, the more information the example i has. Equivalently, the example i is more likely to be selected into the sparse graphs.

3.2 Active learning of Gaussian processes with MPGR

In this section, we will introduce our method which applies MPGR to active learning of Gaussian processes.

As mentioned above, 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. We denote the new method as GPMAL. By exploiting aforementioned MPGR, 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. Since these examples are high representative, we can just select them to represent the whole data set to a large extent. Compared with the original GPAL, GPMAL considers the structural space connectivity among the unlabeled data. Moreover, GPMAL can effectively maintain the manifold structure and eliminate the influence of noisy points and outliers which will be excluded due to the low space connectivity. The GPMAL algorithm is represented in Algorithm 2.

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
```

The difference between Algorithm 1 and Algorithm 2 is the scale of unlabeled examples to be queried. The former queries all the unlabeled examples, while the latter is just querying a subset. By using the subset constructed by the MPGR algorithm, we select m examples with important information from all unlabeled examples. It means that only a portion of unlabeled examples are needed to evaluate kernel functions. Moreover, a classifier learned from these examples will generalize well to the unselected examples. This owns to the property of manifold-preserving sparse graphs: examples outside of the sparse graph have similar features and labels with the points in the sparse graph.

It can be seen that our method is a refinement of the original active learning method of Gaussian processes. Essentially, it consists of two steps. First, we construct a sparse subset by MPGR. Due to the property of high space connectivity, the subset is high representative and preserving the global structure of the original distribution of training set. Then we use the active learning procedure to reselect unlabeled points based on the subset. Thus it can not only reduce the number of unlabeled points to be queried, but also provide a guarantee that the selected points are important and valuable. Further it can remove the noisy points and outliers from the candidate points.

4 Experiments

We evaluate our method on one artificial data set and five real data sets. The five real data sets are separately the Blood Transfusion Service Center (BTSC) data set [23], the Monk's Problems (MP) data set, the Vertebral Column (VC) data set, the Balance Scale (BS) data set, and the Concrete and Cardiotoco graphic (CCG) data set. All the real data sets are public which can be downloaded from the UCI Machine Learning Repository [1]. To demonstrate the generality of our method, these data sets include both binary classification and multi-class classification.

In our experiments we set the noise model variance $\sigma = 10^{-5}$ (experiments show that the final results are not sensitive to this value). The GPML toolbox [2] is used to construct Gaussian processes for MPGR, GPAL and GPMAL, providing the mean function, covariance function, likelihood function and inference method. For the graph weight matrix in MPGR, we generally use the RBF (radio basis function) kernel as the symmetrical weight matrix. The weight is defined as:

$$W_{ij} = \begin{cases} \exp(-\frac{||x_i - x_j||^2}{c\beta}), & \text{if } x_i,\ x_j \text{ are neighbors },\\ 0, & \text{otherwise }. \end{cases} \tag{9}$$

Here c is a parameter varying in $\{1,5,10\}$, and β is the mean of all the smallest distances between one point and its neighbors. The k-nearest-neighbor rule is used to construct the adjacency graph where k is set to 10 in this paper. We proceed to choose all the parameters by five-fold cross-validation on the training set and conduct experiments ten times on each data set.

To show the effectiveness of our method, three methods are compared in this paper: GPMAL, GPAL and random selection. In our experiments, we focus on the average error rates per class of ten times about six data sets. During each iteration, we select several points to add into the training set and accordingly reduce them from the test data. The difference is that GPMAL selects these points from the subset constructed by MPGR, while GPAL selects them from the whole unlabeled data set. As a baseline, the method of random selection is randomly picking these points. In the following experiments except for the CCG data set, the number of the original training data set and selected data points at each iteration are all five. For the CCG data set, the number of the original training data set and selected points are 50 and 10, respectively. Here

[1] http://archive.ics.uci.edu/ml/

[2] http://gaussianprocess.org/gpml/

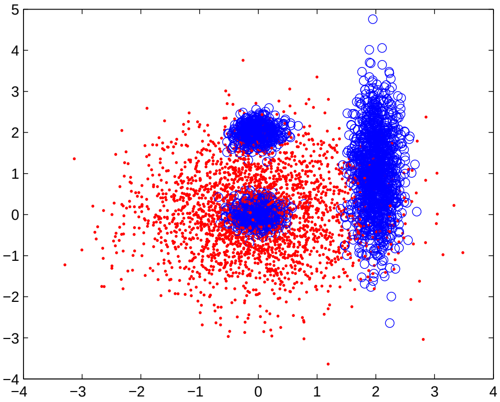


Fig. 1 The distribution of the Clouds data set.

we must consider a fact that the size of the subset constructed by MPGR should not be too large. From [20], we can see that the classification accuracy often first increases and then decreases as the proportion of unlabeled examples retained increases.

4.1 Artificial data

The first data set is an artificial data set (Clouds data set) [3]. Fig. 1 depicts its distribution which is a two-class classification problem. This data set contains 5000 examples, and 2500 examples per class. Each data point has two attributes. For simplification, we randomly choose 600 points to perform the experiment and the number of each class is set to 300. We randomly choose five labeled examples as the training data and the rest unlabeled examples as the test data. In GPMAL, the size of subsets constructed by the MPGR algorithm is 100. Fig. 2 shows the unlabeled points selected by GPMAL and GPAL, respectively. The red points with "+" mean that they are the points selected for active learning. From Fig. 2, it is straightforward to see that the points selected by GPMAL are more representative of the global information than GPAL.

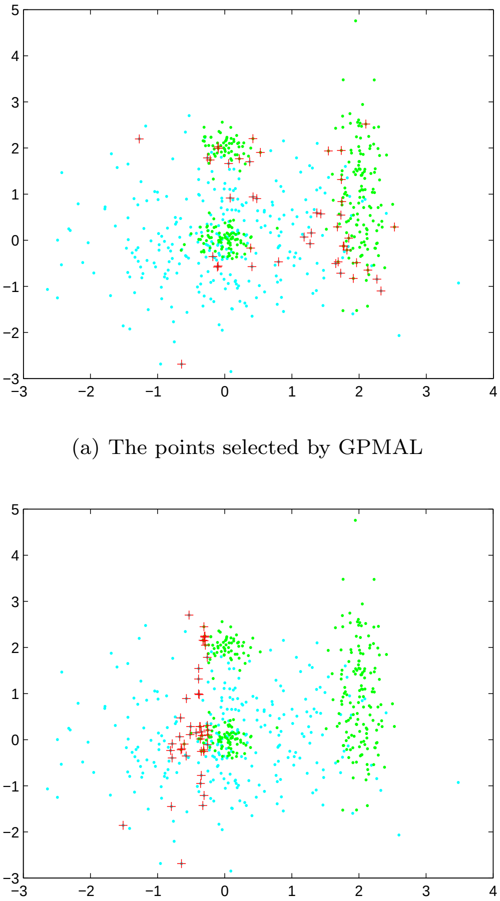
Fig. 3 shows the average classification error rates of three methods: GPMAL, GPAL and Random (the random example selection method). From Fig. 3 we can clearly see that our new approach obtains a lower classification error rate compared with GPAL and random selection.

4.2 UCI data sets

To demonstrate the generality and effectiveness of our approach, we further evaluate it on five real data sets which are benchmark data for machine learning algorithms.

The BTSC data set seeks to predict if a man donated blood or not. It is also a binary classification problem which consists of 748 examples. Each example has four

[3] https://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/databases/



(a) The points selected by GPMAL

(b) The points selected by GPAL

Fig. 2 The distribution of the points selected by GPMAL and GPAL.

attributes which describe some information about donation and a binary output feature representing whether a man donated blood. As mentioned above, the original training data set is five and the subset size is 200.

The MP data set is the basis of the first international comparison of learning algorithms. The data set includes 432 examples where each data point includes 7 features and an output attribute. The size of the subset constructed by MPGR is 100. It is also a two-class classification problem.

The VC data set contains six biomechanical features. It is used to classify orthopaedic patients into two classes (normal or abnormal). There are 310 examples in total and the subset size is 100.

The above experiments are all binary classification problems. Fig. 4∼6 show the experimental results. It is straightforward to see that GPMAL obtains a better classification performance than GPAL and random selection.

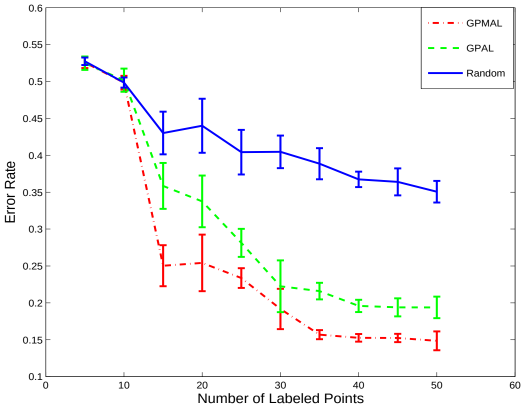


Fig. 3 Experimental results on the Artificial data set.

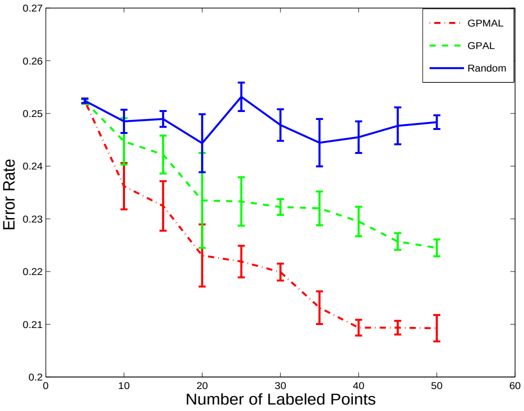


Fig. 4 Experimental results on the BTSC data set.

The BS data set is generated to model psychological experimental results. Each example is classified as: tip to the left (left), tip to the right (right), or to be balanced (balance). There are 625 examples in total. We set the subset size to be 100. It is a multi-class classification problem which has four feature attributes and one predicted attribute. Fig. 7 shows the performance comparison of GPMAL classification with GPAL and random selection on this data set.

The CCG data set consists of measurements of fetal heart rate (FHR) and uterine contraction (UC) features on cardiotocograms classified by expert obstetricians. It consists of 2126 examples. Each data point includes 21 feature attributes and two predicted attributes. It can be a three-class classification or a ten-class classification problem. For simplicity, we predict the NSP (Normal, Suspect, Pathologic) feature, setting it as a three-class classification problem. The original labeled points are 50 and

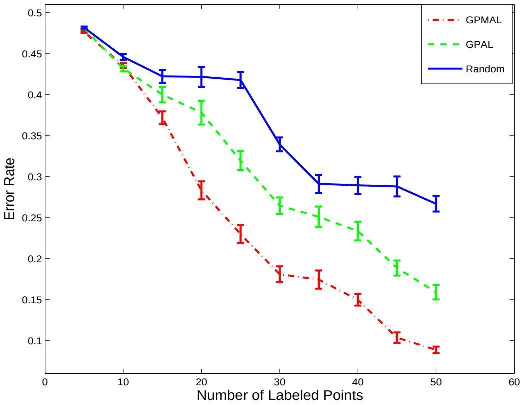


Fig. 5 Experimental results on the MP data set.

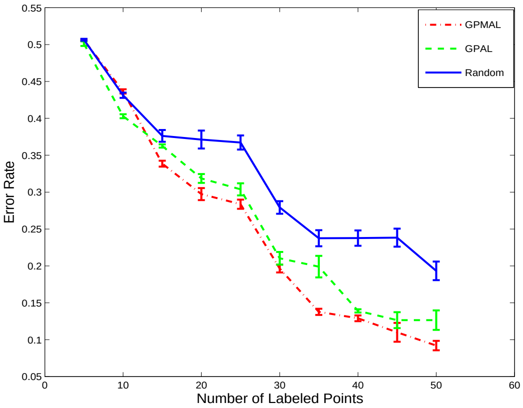


Fig. 6 Experimental results on the VC data set.

Table 1 The error rates for all real data sets with the maximum numbers of labeled points

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

the number of points added into training data is set to 10. The subset size is 500. Fig. 8 shows the performance comparison of GPMAL classification with GPAL and random selection for classification problems on this data set.

The experimental results on the six data sets, which include binary classification and multi-class classification problems, show that our approach GPMAL obtains a lower error rate than GPAL and random selection. Moreover, we list the error rates for all real data sets with the maximum numbers of labeled points (50 or 140) in Table 1. It is straightforward to see the superior performance of GPMAL. This might be due to the following reasons. Firstly, compared with GPAL, MPGR constructs an important and informative subset to represent the global manifold structure and describe local characteristics. Secondly, it can effectively eliminate the influence of noisy points and outliers. Essentially, GPMAL is a refinement of the original GPAL.

4.3 Comparing with feature selection

As mentioned above, the main advantage of GPMAL is that it constructs a subset which well considers the global structural information before labeling points. The structural information of points, to a great extent, relies on their feature representation. Feature selection can thus influence the structural information. To show our method is more competitive, in this section, we apply principal component analysis (PCA) to the data points before GPAL (PCA-GPAL) and further compare it with other methods.

PCA is an unsupervised dimensionality reduction method. It seeks a set of orthogonal projection directions along which the sum of the variances of data is maximized, while retaining as much as possible of the variation present in the data set. In our method, we capture 99% of the data variance to decide the number of bases used. We compare the four methods on three data sets including the aforementioned VC data set, the CCG data set and another new Ionosphere data set which can also be downloaded from the UCI Machine Learning Repository. The Ionosphere data set includes 351 examples where each data point includes 34 features and an output attribute. It is a binary classification problem. The experiment setting of the Ionosphere data set is the same as the VC data set (The number of the original training data set and selected data points at each iteration are all five. The size of the subset constructed by MPGR is 100).

When using PCA-GPAL, we capture 99% of the data variance while reducing the dimensionality from 6 to 4 with PCA before GPAL on the VC data set. Similarly, the dimensionality is reduced from 34 to 21 on the Ionosphere data set and from 21 to 8 on the CCG data set. Fig. 9∼11 show the performance comparison on the three data sets, which further indicate the superiority of GPMAL.

5 Conclusions

In this paper, we applied the MPGR algorithm to an active learning method based on Gaussian processes, and presented our new method GPMAL. As a manifold-preserving graph reduction algorithm, MPGR constructs a subset which well exploits the structural space connectivity and spatial diversity among examples, and thus the subset is high representative and maintains a good global manifold structure. Especially when there are noisy examples and outliers in the training data, the MPGR algorithm can effectively remove them. By using MPGR, an active learner selects the most informative candidates from the subset instead of the whole unlabeled data set when we select unlabeled examples for human labeling. Compared with the original GPAL, GPMAL is a refinement making use of the MPGR algorithm. Moreover, we compare GPMAL with feature selection (PCA-GPAL) and further show the superiority of GPMAL.

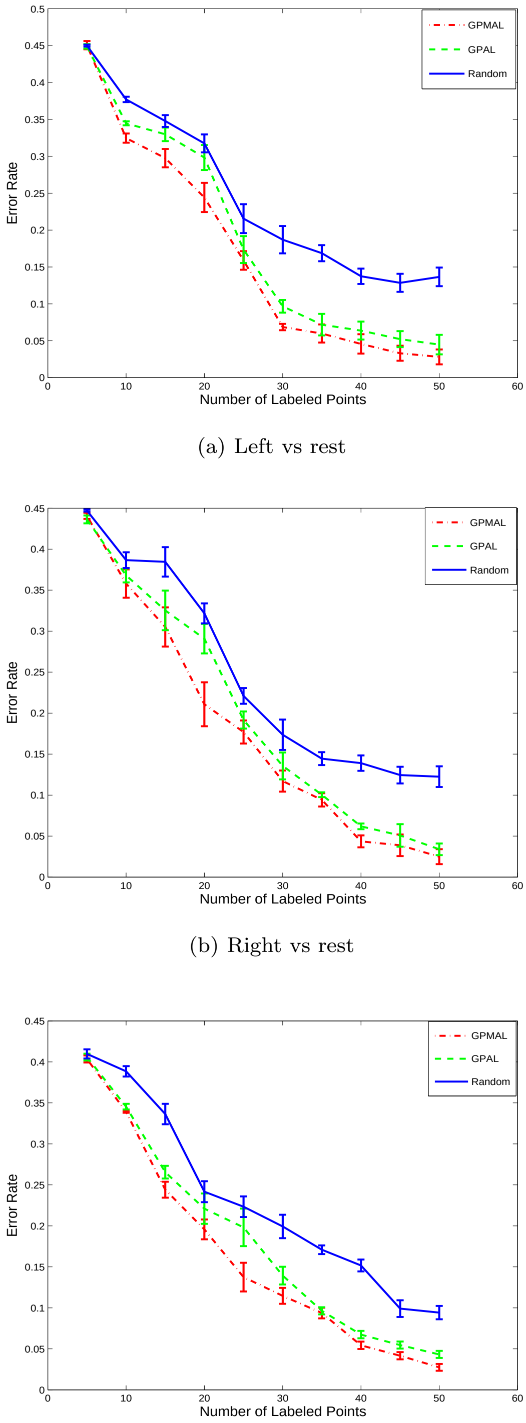
Experimental results on multiple data sets show that our new method GPMAL outperforms random selection and the original GPAL. MPGR for active learning of Gaussian processes has got good performance. In addition, extensions of the MPGR algorithm to other learning contexts will be a topic of interest for future work.

Acknowledgements This work is supported by the National Natural Science Foundation of China under Projects 61370175 and 61075005, and Shanghai Knowledge Service Platform Project (No. ZF1213). Jin Zhou is grateful to Jiang Zhu for helpful discussions on the implementation of the MPGR algorithm.

References

1. M. Belkin, and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation", *Neural Computation*, vol. 15, pp. 1373-1396, 2003.
2. M. Belkin, P. Niyogi and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples", *Journal of Machine Learning Research*, vol. 7, pp. 2399-2434, 2006.
3. C. Campbell, N. Cristianini, and A. Smola, "Query learning with large margin classifiers", *Proceedings of the International Conference on Machine Learning*, pp. 111-118, 2000.
4. D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning", *Machine Learning*, vol. 15, pp. 201-221, 1994.
5. I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers", *Proceedings of the International Conference on Machine Learning*, pp. 150-157, 1995.
6. Y. Freund, H. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm", *Machine Learning*, vol. 28, pp. 133-168, 1997.
7. A. Hauptmann, W. Lin, R. Yan, J. Yang, and M. Chen, "Extreme video retrieval: Joint maximization of human and computer performance", *Proceedings of the ACM Workshop on Multimedia Image Retrieval*, pp. 385-394, 2006.
8. S. Hoi, R. Jin, J. Zhu, and M. Lyu, "Batch mode active learning and its application to medical image classification", *Proceedings of the International Conference on Machine Learning*, pp. 417-424, 2006.
9. D. Lewis and W. Gale, "A sequential algorithm for training text classifiers", *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3-12, 1994.
10. Y. Liu, "Active learning with support vector machine applied to gene expression data for cancer classification", *Journal of Chemical Information and Computer Sciences*, vol. 44, pp. 1936-1941, 2004.
11. A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Gaussian processes for object categorization", *International Journal of Computer Vision*, vol. 88, pp. 169-188, 2010.
12. A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Active learning with Gaussian processed for object categorization", *Proceedings of the International Conference on Computer Vision*, pp. 1-8, 2007.
13. A. Krause, and C. Guestrin, "Nonmyopic active learning of gaussian processes: An exploration-exploitation approach", *Proceedings of the International Conference on Machine Learning*, pp. 449-456, 2007.
14. C. Rasmussen, and C. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge, 2006.
15. N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction", *Proceedings of the International Conference on Machine Learning*, pp. 441-448, 2001.
16. G. Schohn and D. Cohn, "Less is more: Active learning with support vectors machines", *Proceedings of the International Conference on Machine Learning*, pp. 839-846, 2000.
17. B. Settles, M. Craven, and L. Friedland, "Active learning with real annotation costs", *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, pp. 1-10, 2008.
18. S. Sun, "Multi-view Laplacian support vector machines", *Lecture Notes in Artificial Intelligence*, vol. 7121, pp. 209-222, 2011.

19. S. Sun, D. Hardoon, "Active learning with extremely sparse labeled examples", *Neurocomputing*, vol. 73, pp. 2980-2988, 2010.
20. S. Sun, Z. Hussain and J. Shawe-Taylor, "Manifold-preserving graph reduction for sparse semi-supervised learning", *Neurocomputing*, vol. 124, pp. 13-21, 2013.
21. S. Tong, "Active learning: Theory and applications", Ph.D. Thesis, Stanford University, 2001.
22. D. Tuia, F. Ratle, F. Pacifici, M. Kanevski, and W. Emery, "Active learning methods for remote sensing image classification", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, pp. 2218-2232, 2009.
23. I. Yeh, K. Yang, and T. Ting, "Knowledge discovery on RFM model using Bernoulli sequence", *Expert Systems with Applications*, vol. 36, pp. 5866-5871, 2009.
24. Q. Zhang, and S. Sun, "Multiple-view multiple-learner active learning", *Pattern Recognition*, vol. 43, pp. 3113-3119, 2010.
25. C. Zhang and T. Chen, "An active learning framework for content based information retrieval", *IEEE Transactions on Multimedia*, vol. 4, pp. 260-268, 2002.
26. Y. Zhou, and S. Goldman, "Democratic co-learning", *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, pp. 594-602, 2004.
27. J. Zhu, and S. Sun, "Single-task and multitask sparse Gaussian processes", *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 1033-1038, 2013.

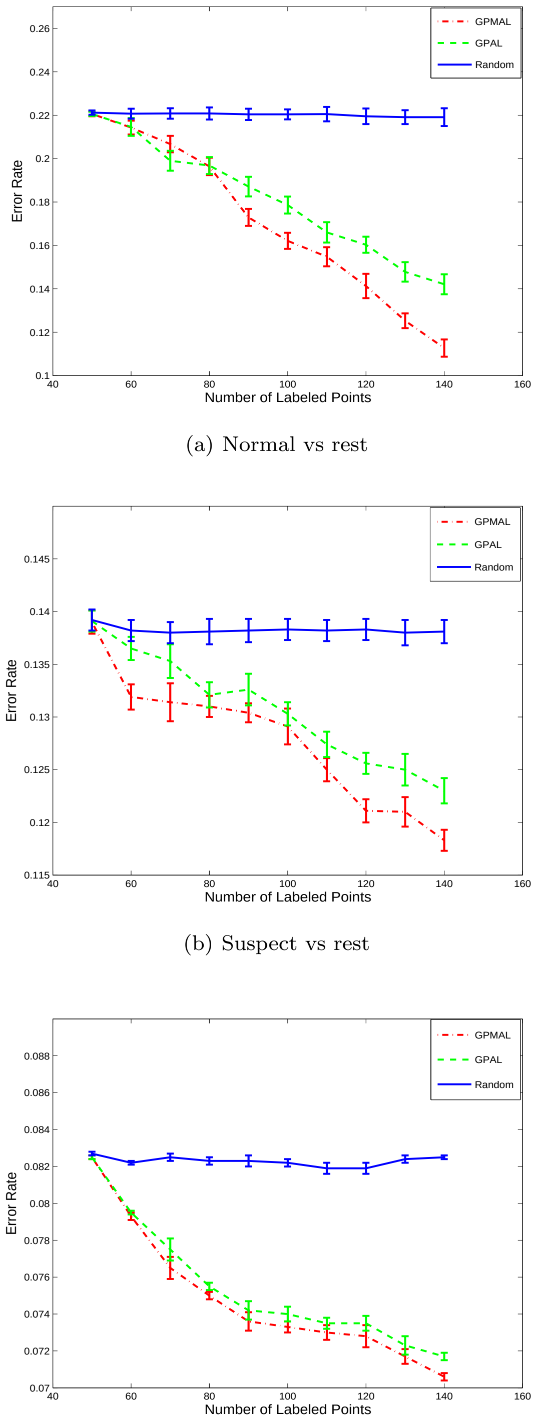


(a) Left vs rest

(b) Right vs rest

(c) Balance vs rest

Fig. 7 Performance comparison of GPMAL, GPAL and random selection on the BS data set.



(a) Normal vs rest

(b) Suspect vs rest

(c) Pathologic vs rest

Fig. 8 Performance comparison of GPMAL, GPAL and random selection on the CCG data set.

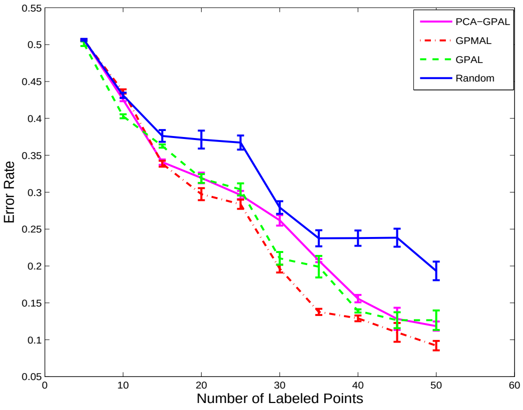


Fig. 9 Performance comparison of PCA-GPAL, GPMAL, GPAL and random selection on the VC data set.

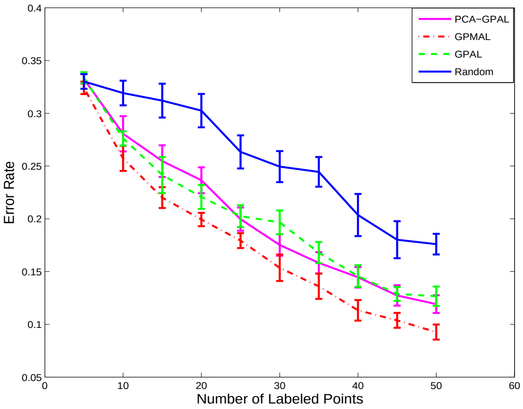
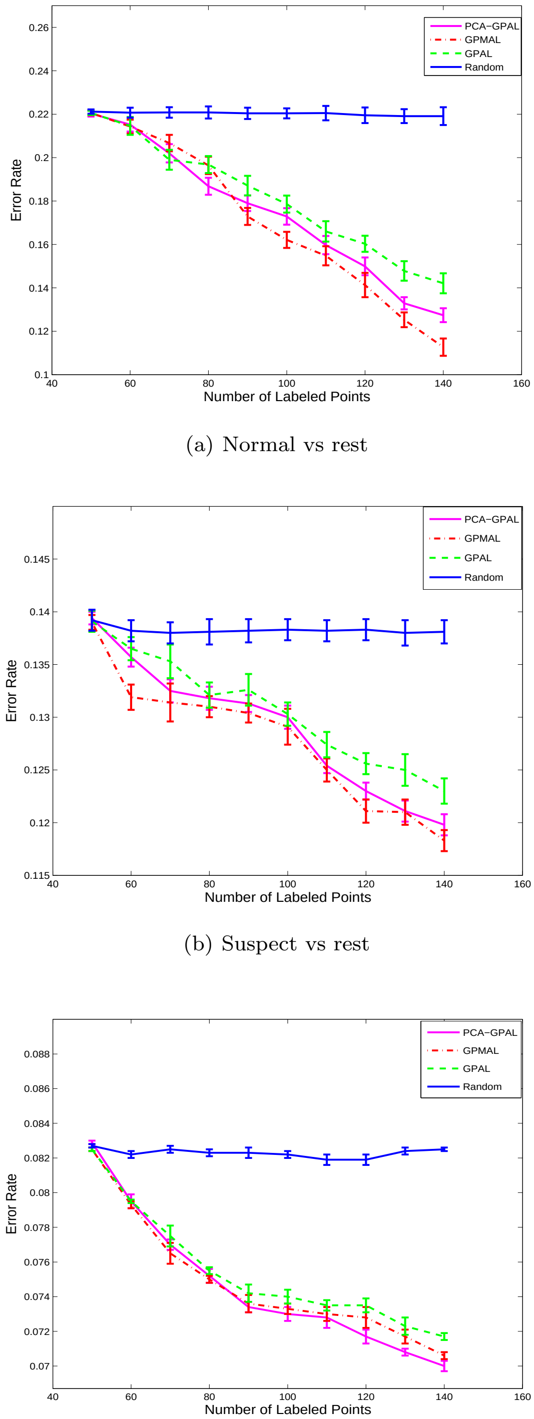


Fig. 10 Performance comparison of PCA-GPAL, GPMAL, GPAL and random selection on the Ionosphere data set.



(a) Normal vs rest

(b) Suspect vs rest

(c) Pathologic vs rest

Fig. 11 Performance comparison of PCA-GPAL, GPMAL, GPAL and random selection on the CCG data set.