SINGLE-TASK AND MULTITASK SPARSE GAUSSIAN PROCESSES

JIANG ZHU, SHILIANG SUN

Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, P. R. China
E-MAIL: jiangzhu11@gmail.com, slsun@cs.ecnu.edu.cn

Abstract:

Gaussian processes are a powerful non-parametic tool for Bayesian inference but are limited by their cubic scaling problem. This paper aims to develop single-task and multitask sparse Gaussian processes for both regression and classification. First, we apply a manifold-preserving graph reduction algorithm to construct single-task sparse Gaussian processes from a sparse graph perspective. Then, we propose a multitask sparsity regularizer to simultaneously sparsify multiple Gaussian processes from related tasks. The regularizer can encourage the global structures of retained points from closely related tasks to be similar, and structures from loosely related tasks to be less similar. Experimental results show that our single-task sparse Gaussian processes are comparable to one state-of-the-art method, and our multitask sparsity regularizer can generate multitask sparse Gaussian processes which are more effective than those obtained from other methods.

Keywords:

Sparse Gaussian processes; Manifold-preserving graph reduction; Multitask sparsity regularizer

1. Introduction

Gaussian processes are a popular and powerful non-parametric tool for Bayesian inference. They are a probabilistic method which can provide estimations of the uncertainty of predictions. For instance, in classification Gaussian processes can estimate the posterior probability of labels, and in regression Gaussian processes can calculate 'error-bar' of estimated values. This property leads to their easy explanation and widespread use. Unfortunately they need $O(n^3)$ time for training where n is the size of the training set. This paper focuses on overcoming the scaling problem of the cubic time complexity with respect to the training size.

Various efforts [1], [2], [3] have been made to overcome this scaling problem, which can be classified into two categories. The first category of methods select a subset of the training points of size d with $d \ll n$. This can bring the scaling down to $O(nd^2)$ time and $O(nd)$ memory. The second category uses approximate matrix-vector multiplication methods. Our work in this paper belongs to the first category. As far as we know, none of the existing sparse Gaussian processes considers the manifold assumption of data. However, learning with the manifold assumption has been successful applied to many machine learning tasks [4], [5], [6], [7], [8]. Manifold-preserving graph reduction (MPGR) is a recent efficient graph sparsification algorithm [9] which can effectively remove outliers and noisy examples. The first contribution of this paper is that we successfully apply it to the data representation of sparse Gaussian processes in a single-task setting.

Multitask learning is an active research direction [10], [11], [12]. Simultaneously learning multiple tasks can be more effective than learning them separately because the relationship between tasks can benefit learning. How to learn and reflect the task relationship in multitask learning is crucial. One common approach is to set some parameters to be shared by all the tasks. For example, the multi-task informative vector machine (MTIVM) [13] shares the kernel matrix among tasks to build multitask sparse Gaussian processes. Different from this kind of methods, here we attempt to construct multitask sparse Gaussian processes by simultaneously considering the global structures of points from different tasks.

The second and main contribution of this paper is that we propose a multitask sparsity regularizer for subset selection among multiple tasks. The regularizer consists of two factors which utilize the distance between task-descriptor features to represent the task relevance and the distance of a data point to its k-nearest neighbors from other tasks to represent its similarity to the structures of other tasks. Here, the task-descriptor feature is defined by the principal variable algorithm [14]. Combining this regularization formula with existing single-

task sparsification criteria will result in multitask sparsification methods. We integrate the regularizer with informative vector machine (IVM) and MPGR to get rMTIVM and rMTMPGR, respectively. Here, r stands for relevance because our method explicitly considers task relevance through the task-descriptor features. Experimental results show that our method significantly promotes the prediction performance.

The rest of the paper is organized as follows. First we introduce some background including Gaussian processes, IVM and MTIVM. We use MPGR to construct single-task sparse Gaussian processes in the next section. Then, we present the multitask sparsity regularizer and apply it to multitask sparse Gaussian processes, after which experimental results on five real data sets are reported. Finally, we make our conclusions.

2. Background

In this section we briefly summarize Gaussian processes and introduce sparse methods IVM and MTIVM that will be used for comparsion in our experiments.

2.1. Gaussian processes

A Gaussian process can be specified by a mean function and a covariance function [15]. Given a data set consisting of N examples $\mathbf{X} = \{\mathbf{x_n}\}_{n=1}^{N}$, corresponding observed labels $\mathbf{y} = \{y_n\}_{n=1}^{N}$ and a set of latent variables $\mathbf{f} = \{f_n\}_{n=1}^{N}$, the prior distribution for these latent variables is assumed to be Gaussian

$$p\left(\mathbf{f}|\mathbf{X},\theta\right) = \mathcal{N}\left(\mathbf{f}|\mathbf{0},\mathbf{K}\right) \tag{1}$$

with a zero mean and a covariance matrix $\mathbf{K}$ which is parameterized by the kernel hyperparameter θ. The joint likelihood can be written as

$$p\left(\mathbf{y},\mathbf{f}|\mathbf{X},\theta\right) = p\left(\mathbf{f}|\mathbf{X},\theta\right)p(\mathbf{y}|\mathbf{f}). \tag{2}$$

For regression, a Gaussian observation likelihood is often used

$$p\left(\mathbf{y}|\mathbf{f}\right) = \mathcal{N}\left(\mathbf{y}|\mathbf{f},\sigma^2\mathbf{I}\right). \tag{3}$$

After integrating out the latent variables, the marginal likelihood will be

$$p\left(\mathbf{y}|\mathbf{X}\right) = \mathcal{N}\left(\mathbf{y}|\mathbf{0},\mathbf{K}+\sigma^2\mathbf{I}\right). \tag{4}$$

The prediction distribution of the label at a new point $\mathbf{x}_*$ is also Gaussian. Its mean is ${\mathbf{k}_*}^T\left(\mathbf{K}+\sigma^2\mathbf{I}\right)^{-1}\mathbf{y}$ and covariance is $\tilde{k}\left(\mathbf{x}_*,\mathbf{x}_*\right) - {\mathbf{k}_*}^T\left(\mathbf{K}+\sigma^2\mathbf{I}\right)^{-1}\mathbf{k}_*$. Here, $\tilde{k}$ is the covariance function and $\mathbf{k}_*$ is the vector of covariances between $\mathbf{x}_*$ and the training points.

Gaussian processes need $O(n^3)$ time complexity because of the inversion of the corresponding covariance matrix. For large data sets, therefore we must seek an effective sparse method to reduce the computation time.

2.2. IVM and MTIVM

IVM tries to resolve the cubic scaling problem of Gaussian processes by seeking a sparse representation of the training data [1]. Based on information theory, it seeks to extract the maximum amount of information with the minimum number of data points. In particularly, points with the most information, namely giving the largest reduction in the posterior process entropy, are selected. The entropy reduction associated with selecting the nth point at the ith selection is given by

$$\Delta H_{in} = -\frac{1}{2}\log\left|\Sigma_{i,n}\right| + \frac{1}{2}\log\left|\Sigma_{i-1}\right|, \tag{5}$$

where Σ_{i-1} is the posterior covariance after the $(i-1)$th selection and $\Sigma_{i,n}$ is the posterior covariance after selecting the nth point at the ith selection.

The IVM approach is later extended to multitask learning [13]. Under the constraint that L tasks are conditionally independent on the kernel hyperparameter θ, which is shared among all the tasks, the distribution model of MTIVM is

$$p\left(\mathbf{Y}|\tilde{\mathbf{X}},\theta\right) = \prod_{\ell=1}^{L} p\left(\mathbf{y}_\ell|\mathbf{X}_\ell,\theta\right), \tag{6}$$

where the columns of $\mathbf{Y}$ and $\tilde{\mathbf{X}}$ are $\mathbf{y}_\ell$ and $\mathbf{X}_\ell$, respectively, and each $p\left(\mathbf{y}_\ell|\mathbf{X}_\ell,\theta\right)$ is a Gaussian process. Assuming vector $\tilde{\mathbf{y}}$ is formed by stacking columns of $\mathbf{Y}$, $\tilde{\mathbf{y}} = \left[\mathbf{y}_1^{\mathrm{T}}\ldots\mathbf{y}_L^{\mathrm{T}}\right]^{\mathrm{T}}$, the covariance matrix is then

$$\tilde{\mathbf{K}} = \begin{bmatrix} \mathbf{K}_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{K}_L \end{bmatrix}, \tag{7}$$

where $\mathbf{K}_\ell$ is the covariance matrix of $\mathbf{X}_\ell(\ell = 1,...,L)$.

The overall likelihood is thus a Gaussian process

$$p\left(\tilde{\mathbf{y}}|\tilde{\mathbf{X}},\theta\right) = \mathcal{N}\left(\mathbf{0},\tilde{\mathbf{K}}\right). \tag{8}$$

The selected subset of the training data for MTIVM is shared among all tasks which can reduce the computation and memory

consumption. The entropy reduction with the nth point for task ℓ at the ith selection is given by

$$\Delta H_{in}^{(\ell)} = -\frac{1}{2}\log\left|\Sigma_{i,n}^{(\ell)}\right| + \frac{1}{2}\log\left|\Sigma_{i-1}^{(\ell)}\right|, \tag{9}$$

where $\Sigma_{i-1}^{(\ell)}$ is the posterior covariance of task ℓ after the $(i-1)$th selection and $\Sigma_{i,n}^{(\ell)}$ is the posterior covariance of task ℓ after selecting the nth point at the ith selection.

3. Single-task sparse Gaussian processes

In this section, we apply MPGR [9] to construct sparse Gaussian processes in the single-task setting.

A sparse graph with manifold-preserving properties means that a point outside of it should have a high connectivity with a point retained. A manifold-preserving sparse graph is the graph that maximizes the quantity

$$\frac{1}{m-t}\sum_{i=t+1}^{m}\left(\max_{j=1,\ldots,t} W_{ij}\right) \tag{10}$$

where m is the number of all vertices, W is the weight matrix and t is the number of vertices to be reserved.

Using the McDiarmid's inequality it has been proved that maximizing (10) can enlarge the lower bound of the expected connectivity between the sparse graph and the $m-t$ vertices outside of the sparse graph. This provides a guarantee of obtaining a good space connectivity. As directly seeking manifold-preserving sparse graphs is NP-hard, the MPGR algorithm seeks an approximation to maximizing (10).

For subset selection of sparse Gaussian processes, a point that is closer to surrounding points should be selected because it is more likely to contain more important information. This is exactly reflected by the MPGR algorithm where points with a large degree will be preferred. Another reason for adopting MPGR to construct sparse Gaussian processes is that the property of manifold-preserving requires a high space connectivity, which tends to select globally representative points.

The high space connectivity among points in the sparse graph returned by MPGR is of great practical significance. For regression, it tends to select points with wide spread rather than points gathered together. This maintains a good global structure which is less prone to overfitting. For example, if the points retained reside in a small area, the regression function will probably overfit these points and generalize badly to other areas. For classification, a classifier learned from the sparse graph obtained by MPGR will generalize well to the unselected points. The reason is that from the definition of manifold-preserving sparse graphs, points outside of the sparse graph have high feature similarities to the vertices in the sparse graph. High similarities of features usually leads to high similarities of labels. This classifier is also likely to generalize well to points not in the training set, as a result of the high space connectivity of the sparse graph.

The MPGR algorithm for constructing sparse Gaussian processes is shown is composed of three procedures: 1) Construct a graph with all training points. In this paper, we adopt the k-nearest-neighbor rule for graph adjacency construction where k is set to 10. 2) Select a subset by the MPGR algorithm. Slightly different from the original algorithm we just retain the selected vertices while ignore the weights which are unnecessary for subsequent training of Gaussian processes. 3) Train a Gaussian process with the subset.

4. Multitask sparse Gaussian processes with multitask sparsity regularization

It is known that learning multiple tasks simultaneously has the potential to improve the generalization performance. Applying this idea to the subset selection of multitask sparse Gaussian processes, we propose a multitask sparsity regularizer which seeks to simultaneously construct multiple sparse Gaussian processes.

Our starting point is that the global structures of retained points from closely related tasks should be similar and structures from loosely related tasks should be less similar. Now we explain the rationality of this intuition through Figure 1. Assume that there are just two closely related tasks, and the top point and the bottom point from left task get an equal value based on their original criterion. From the perspective of keeping the similarity of global structures, the top point is more proper. The multitask sparsity regularizer can be utilized to measure the similarity of global structures. Combined the original criterion with the regularizer, the next point to be selected from the left task should be the top point rather than the bottom right point.

We proceed to transform the above multitask sparsity intuition to mathematical formulations. First of all, we use $\sum_{j=1}^{k'}\frac{1}{\|x_{t_n}-x_i^j\|}$ to measure the similarity of points, where x_{t_n} is a point considered for selection from task t_n and $x_i^j (j = 1, ..., k')$ are k' nearest neighbors of x_{t_n} from already-selected points of another task i. A smaller distance means a closer relationship, and a closer relationship means a high similarity. We use the reciprocal of distance for the sake of maximizing our regularization formula. Considering the possibly wide scatter

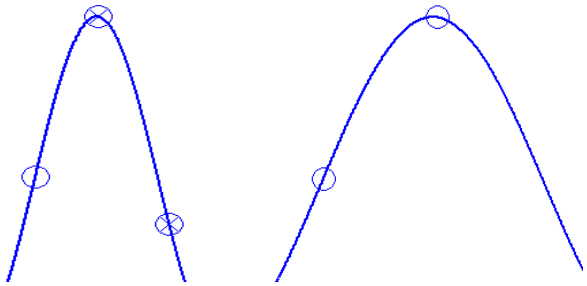

Figure 1. The effect of multitask sparsity regularizer. The left part is the task which is selecting the next data point. The right part is a related task. The lines are the real distribution curves. $\circ$ means a selected point. $\otimes$ means an unselected point.

of retained points of related tasks, we just employ the distance of a point to its k' nearest neighbors from related tasks. For example, suppose that a related task selected four points in group A and four points in group B, and A and B are far apart. If we utilize the distance to all the selected points, the regularizer may help to select a point which is far away from both A and B. When k' is set to be four, points closer to either A or B are encouraged to be selected. k' is three in this paper.

Then, we propose to use $\frac{1}{\|f'_{t_n} - f'_i\|}$ to modulate the similarity, which reflects the relevance between different tasks. Here, f'_{t_n} is the task-descriptor feature of task t_n. E. Bonilla et al. [16] chose eight crucial points and set the mean of their labels to be the the task-descriptor feature. Here, we define the task-descriptor feature in the same way, and the crucial points are selected based on the principal variable algorithm [14]. We use the distance between task-descriptor features to represent the task relevance. A smaller distance between task-descriptor features means tasks are more similar.

We reach the multitask sparsity regularizer by combining the two terms mentioned above. The regularization formula is given as

$$Reg(x_{t_n}) = \sum_{i=1, i\neq t_n}^{n_t} \sum_{j=1}^{k'} \frac{1}{\| f'_{t_n} - f'_i \| \| x_{t_n} - x_i^j \|}, \quad (11)$$

where n_t is the total number of tasks. Maximizing the formula can make the global structures of points of related tasks be similar. Combining the proposed multitask sparsity regularizer with existing single-task sparse criteria in an appropriate way will easily result in our multitask sparse Gaussian processes.

The multitask sparsity regularizer can be applied to IVM almost directly to get a multitask sparse Gaussian process. As mentioned before, IVM maximizes (5). Combined with (11), the sparse criterion is then to maximize

$$-\tfrac{1}{2}\log\left|\Sigma^{(t_n)}_{i,x_{t_n}}\right| + \tfrac{1}{2}\log\left|\Sigma^{(t_n)}_{i-1}\right| + \lambda Reg(x_{t_n}), \quad (12)$$

where λ controls the proportion between IVM formula and our regularizer. We call the multitask sparse Gaussian process underlying (12) relevance multitask informative vector machine (rMTIVM). Here, r stands for relevance since our method explicitly considers task relevance. To make λ easy to set, we split it into two parameters, $\lambda = \alpha \times \beta$. First, we use a normalization parameter α to make the value of our formula be in the same range of IVM formula. Then, we set β to control the relative proportion.

We also integrate our multitask sparsity regularizer with MPGR to induce another multitask sparse Gaussian process which we call rMTMPGR. rMTMPGR maximizes the following objectives to select a point

$$d\left(x_{t_n}\right) + \lambda Reg(x_{t_n}), \quad (13)$$

where $d\left(x_{t_n}\right) = \sum_j w(x_{t_n}, j)$. As mentioned before, IVM is extended to MTIVM by using the same point selection criterion and sharing the kernel parameters and training set among multiple tasks. For comparison, we develop MPGR algorithm for constructing sparse Gaussian processes to MTMPGR in the same way, which selects the point with the largest degree among points of all the tasks.

5. Experiments

We evaluate the proposed single-task and multitask sparse Gaussian processes on five data sets. The GPML toobox [1] and MTIVM toolbox [2] are used to construct Gaussian processes for MPGR, MTMPGR, rMTMPGR, and IVM, MTIVM, rMTIVM, respectively.

We perform experiments for single-task sparse Gaussian processes on three data sets and multitask sparse Gaussian processes on two data sets. All the data sets are publicly available where Haberman's Survival (HS), IRIS, Auto MPG, Concrete Slump (CS) are from the UCI Machine Learning Repository, Landmine data can be found in a website [3].

All the parameters are selected by five-fold cross-validation on training data. We define the graph weight in MPGR using

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

[2] http://www.dcs.shef.ac.uk/∼neil

[3] http://www.ece.duke.edu/lcarin/∼Land-mineData.zip

TABLE 1. Experimental Results on HS.

Method	Error rate (%)	Time (s)
IVM	29.5 ± 2.2	14.4
MPGR	$\mathbf{27.1 \pm 4.4}$	**13.2**

TABLE 2. Experimental Results on IRIS.

Method	Error rate (%)	Time (s)
IVM	5.0 ± 2.1	4.0
MPGR	$\mathbf{4.2 \pm 1.8}$	**3.8**

TABLE 3. Experimental Results on Auto MPG.

Method	MARE	Time (s)
IVM	$\mathbf{13.2 \pm 1.5}$	6.0
MPGR	14.8 ± 1.0	**2.9**

TABLE 4. Experimental Results on LANDMINE.

Method	Error rate (%)	Time (s)
MTIVM	9.8 ± 5.0	**8.7**
rMTIVM	7.8 ± 2.7	22.8
MTMPGR	7.4 ± 2.8	13.2
rMTMPGR	$\mathbf{5.8 \pm 1.5}$	31.6

the Gaussian RBF (radial basis function) kernel

$$w_{i,j} = e^{\frac{-\|x_i - x_j\|}{t \times m_x}} \tag{14}$$

where t is a parameter varying in $\{1, 5, 10\}$ and m_x is the mean of all the smallest distances between one point and its neighbors. The normalization parameter α is set to be the ratio of the maximum values of the formula of IVM or MPGR and our multitask sparsity regularizer. The proportion parameter β is selected from $\{1/2, 2/3, 1, 3/2, 2\}$. For constructing gaussian processes, the type of mean function, covariance function, likelihood function and inference method are selected by the accuracy rate of experiments on test data.

All the results are based on ten random splits of training and test data for each data set. We measure the experimental performance by the accuracy and the average time consumption. For classification problems, accuracy is measured by the error rate (%). For regression problems, accuracy is measured by the mean absolute relative error (MARE), which is defined as

$$MARE = \frac{1}{\ell_x} \sum_i \left| \frac{x_i^* - x_i}{x_i} \right|, \tag{15}$$

where x_i is the real value of the ith point, x_i^* is its predicted value and ℓ_x is the total number of points.

5.1. Single-task sparse Gaussian processes

The HS data set is used to predict if a cancer patient could survive more than five years from the age of patient, the operation year and the number of positive axillary nodes detected. We randomly choose 200 points as training data, the rest points for test and the subset size is 100.

The IRIS data set contains three classes of 50 instances each. It has four feature attributes and one predicted attribute. It is a multi-class classification problem. We randomly choose 100 points as the training data, 50 for test and the subset size is 55.

The Auto MPG data concerns the city-cycle fuel consumption in miles per gallon, which is a regression problem. We modify the original data set by removing the car name attribute due to its uniqueness for each instance, and use seven attributes to predict the continuous attribute MPG. We randomly choose 200 points as the training data, 100 for test and the subset size is 100.

Our experiments include binary classification, multi-class classification and regression problems which can show the wide applicability of our methods. TABLE 1, TABLE 2 and TABLE 3 gives the experimental results. It is straightforward to see that MPGR for sparse Gaussian processes is comparable to IVM. For classification problems MPGR obtains a lower error rate than IVM. For the regression problem MPGR is slightly worse than IVM. MPGR has a less time consumption than IVM.

5.2. Multitask sparse Gaussian processes

The Landmine data set is collected from a real landmine field. It's a binary classification problem that has 19 related tasks with totally 9674 data points and each point is represented by a nine-dimensional feature vector. We randomly choose 320 points as the training data, 80 for test and the subset size is 40.

The CS data set includes 103 data points. There are seven input variables, and three output variables in the data set. We apply this multi-output data to multitask experiments by setting each output as a task. This is a regression problem. We randomly choose 80 points as the training data, 23 for test and the subset size is 50.

The experimental results are shown in TABLE 4 and TABLE 5, which indicate that the multitask sparsity regularizer can help

TABLE 5. Experimental Results on CS.

Method	Error rate (%)	Time (s)
MTIVM	15.3 ± 2.6	$\mathbf{14.1}$
rMTIVM	$\mathbf{14.7 \pm 1.4}$	33.7
MTMPGR	17.1 ± 3.9	15.0
rMTMPGR	16.0 ± 3.4	36.6

to promote the performance. With the time consumption of our methods being less than three times that of contrast methods, both the averaged error rates and the standard deviations of our methods are smaller.

6. Conclusions

In this paper we have applied the MPGR algorithm to construct single-task sparse Gaussian processes. As a graph sparsification algorithm MPGR works well at maintaining the global structure, which is desirable for constructing sparse Gaussian processes. Experimental results show that our method is comparable to IVM.

We have also proposed a multitask sparsity regularizer for subset selection among multiple tasks. The regularization formulation is composed of two factors, which reflect the task relevance and the similarities of the global structures of retained points between related tasks, respectively. It encourages the global structures of retained points from closely related tasks to be similar. Combining the regularizer with existing single-task sparse criteria results in our multitask sparse Gaussian processes. Experimental results show their effectiveness.

In this paper, we utilized the principal variable algorithm to describe task features that are employed to measure the task relevance. Extensions of our multitask sparsity regularizer to including other task relevance measurements can be interesting future work.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Project 61075005, and Shanghai Knowledge Service Platform Project (No. ZF1213).

References

[1] N. Lawrence, M. Seeger and R. Herbrich, "Fast sparse Gaussian process methods: The informative vector machine", Advances in Neural Information Processing Systems, Vol 15, pp. 609-616, 2002.

[2] J. Quiñonero-Candela and Carl. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression", Journal of Machine Learning Research, Vol 6, pp. 1939-1959, 2005.

[3] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs", Advances in Neural Information Processing Systems, Vol 18, pp. 1257-1264, 2006.

[4] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding", Science, Vol 290, pp. 2323-2326, 2000.

[5] J. Tenenbaum, V. de Silva and J. Langford, "A global geometric framework for nonlinear dimensionality reduction", Science, Vol 290, pp. 2319-2323, 2000.

[6] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation", Neural Computation, Vol 15, pp. 1373-1396, 2003.

[7] 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.

[8] S. Sun, "Multi-view Laplacian support vector machines", Lecture Notes in Artificial Intelligence, Vol 7121, pp. 209-222, 2011.

[9] S. Sun, Z. Hussain and J. Shawe-Taylor, "Manifold-preserving graph reduction for sparse semi-supervised learning", Neurocomputing, DOI: 10.1016/j.neucom.2012.08.070, 2013.

[10] R. Caruana, "Multitask learning", Machine Learning, Vol 28, pp. 41-75, 1997.

[11] T. Jebara, "Multitask sparsity via maximum entropy discrimination", Journal of Machine Learning Research, Vol 12, pp. 75-110, 2011.

[12] L. Ungar, P. Dhillon and D. Foster, "Minimum description length penalization for group and multi-task sparse learning", Journal of Machine Learning Research, Vol 12, pp. 525-564, 2011.

[13] N. Lawrence and J. Platt, "Learning to learn with the informative vector machine", Proceeding of ICML 2004, pp. 1-8, 2004.

[14] J. Cumming and D. Wooff, "Dimension reduction via principal variables", Computational Statistics and Data Analysis, Vol 52, pp. 550-565, 2007.

[15] C. Rasmussen and C. Williams, Gaussian Process for Machine Learning, MIT Press, Cambridge, 2006.

[16] E. Bonilla, F. Agakov and C. Williams, "Kernel multi-task learning using task-specific features", Proceeding of ICAIS, pp. 1-8, 2007.