# Multi-task sparse Gaussian processes with improved multi-task sparsity regularization

Jiang Zhu and Shiliang Sun

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

**Abstract.** Gaussian processes are a popular and effective Bayesian method for classification and regression. Generating sparse Gaussian processes is a hot research topic, since Gaussian processes have to face the problem of cubic time complexity with respect to the size of the training set. Inspired by the idea of multi-task learning, we believe that simultaneously selecting subsets of multiple Gaussian processes will be more suitable than selecting them separately. In this paper, we propose an improved multi-task sparsity regularizer which can effectively regularize the subset selection of multiple tasks for multi-task sparse Gaussian processes. In particular, based on the multi-task sparsity regularizer proposed in [12], we perform two improvements: 1) replacing a subset of points with a rough global structure when measuring the global consistency of one point; 2) performing normalization on each dimension of every data set before sparsification. We combine the regularizer with two methods to demonstrate its effectiveness. Experimental results on four real data sets show its superiority.

**Keywords:** Gaussian processes, multi-task learning, sparse representation, regularization

## 1 Introduction

Gaussian processes [1, 2] are a popular and powerful non-parametric tool for probabilistic modeling. However, the scaling problem of the cubic time complexity with respect to the training size $N$ limits their widespread use. Lots of efforts [4–6, 12, 13] have been made to overcome the cubic time complexity problem. A common way to solve this is to select a subset of the training set to get a sparse representation of the original Gaussian process, where the subset size $d$ is much smaller than $N$. The time complexity of the training can be brought down from $O(N^3)$ to $O(d^2N)$. Various criteria exist for selecting the subset. For example, Lawrence et al. [5] selected the points with the biggest entropy reduction based on information theory. Titsias [6] selected the points with the smallest Kullback-Leibler divergence between the variational distribution and the exact posterior distribution over the latent function value.

Multi-task learning is an active research direction [3, 7, 8, 15]. Simultaneously learning multiple tasks can be more effective than learning them separately because the relationship between tasks can be exploited to benefit learning. In this paper, we focus on multi-task sparse Gaussian processes. Some work [9, 10, 12] has been done to sparsify multiple Gaussian processes. For example, the multi-task informative vector machine (MTIVM) [9] which is an extension of [5] shares the kernel matrix and the training set among tasks. Then, all the tasks just train one Gaussian process for prediction. Time and memory consumption are much reduced, but the performance of this method could be unsatisfactory when large differences between tasks exist. Based on the idea that global structures of subsets of multiple tasks should be consistent, recently Zhu and Sun[12] proposed a multi-task sparsity regularizer which regularize the subset selection of multiple Gaussian processes.

In this paper, we propose an improved multi-task sparsity regularizer which makes improvements over the above regularizer. It consists of three steps. First, normalization for each data set on each dimension is performed before the subset selection to make them in the same range. Second, it utilizes manifold-preserving graph reduction (MPGR) [12, 14] to select one rough global structure for each task. Last, it replaces the already-selected points in the multi-task sparsity regularization formula with the rough global structures when calculating the Euclidean distance of one data point to its $k$ nearest neighbors from other tasks. We integrate the regularizer with MPGR and manifold-preserving graph reduction with outputs (MPGRO) [13] to get IrMTMPGR and IrMTMPGRO, respectively. Here, “I” stands for “improved” to distinguish them from the multi-task Gaussian processes built by the previous multi-task sparsity regularizer, and “r” stands for “relevance” because our method explicitly considers the task relevance. Experimental results show its effectiveness.

A preliminary report [12] has been presented. In this paper, we make significant improvements for the multi-task sparsity regularizer, and conduct more experiments.

The rest of the paper is organized as follows. First we introduce related work with the multi-task sparsity regularizer. After that, we will analyze two shortages of the multi-task sparsity regularizer, propose the improved multi-task sparsity regularizer and apply it to construct multi-task Gaussian processes. We make our conclusion after experiments on four real data sets.

## 2 Relate Work

In this section, we briefly introduce the multi-task sparsity regularizer[12].

Based on the idea that the global structures of retained points from closely related tasks should be similar and structures from loosely related tasks should be less similar, the multi-task sparsity regularizer is proposed which regularize the subset selection of multiple Gaussian processes.

It is composed of two parts: 1) One to measure the global consistency of two subsets; 2) One to measure the task relevance. For the first part, the authors use the term $\sum_{j=1}^{k} \frac{1}{\|x_{t_n}-x_i^j\|}$ to evaluate, 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 the already-selected points of another task $i$. The reciprocal of the Euclidean distance is adopted for the sake of maximizing the regularization formula. For the second part, the authors use the $\frac{1}{\|f_{t_n}-f_i\|}$ to modulate the task relevance, where $f_{t_n}$ is the task-descriptor feature of task $t_n$. The task-descriptor feature is utilized to describe tasks. Bonilla et al. [11] chose eight crucial points and set the mean of their labels to be the the task-descriptor feature.

The multi-task sparsity regularizer is then reached by combining the two terms mentioned above. The regularization formula is given as

$$Reg(x_{t_n}) = \sum_{i=1,i\neq t_n}^{n_t} \sum_{j=1}^{k} \frac{1}{\parallel f_{t_n} - f_i \parallel\parallel x_{t_n} - x_i^j \parallel}, \tag{1}$$

where $n_t$ is the total number of tasks. A big value of $Reg$ means that the point to be evaluated is more globally consistent with other tasks for its own task.

In that paper, the authors also integrated the multi-task sparsity regularizer with MPGR to get a multi-task sparse Gaussian process, the relevance multi-task manifold-preserving graph reduction (rMTMPGR). The sparse criterion of rMTMPGR is

$$deg\,(x_{t_n}) + \lambda Reg(x_{t_n}), \tag{2}$$

where $deg\,(x_{t_n}) = \sum_j w(x_{t_n}, j)$, and $\lambda$ controls the proportion between the MPGR formula and the regularizer.

## 3 Multi-task sparse Gaussian processes with improved multi-task sparsity regularization

The above multi-task sparsity regularizer seeks to simultaneously construct multiple sparse Gaussian processes utilizing the consistency of global structures of retained points among sparse subsets of different tasks. Although the starting point is reasonable, two shortages still exist.

The first shortage is that the multi-task sparsity regularizer is prone to overfit the initial points. In formula (1), the multi-task sparsity regularizer uses already-selected points of related tasks. This pushes the initial points to a very important position. Many subsequent points would consider the initial points when evaluating their global consistency. Specially before the $k$th selection, the initial points are used in formula (1) for all the previous iterations. This makes the following selected points prone to be close to the initial points. A straightforward strategy to solve this is to replace the already-selected points with the full data sets. But for every selection, it has to calculate the Euclidean distance from one candidate to all the points in the other tasks. For the efficiency purpose, we replace the already-selected points with a rough global structure. By the rough global structure, we mean that its size should be larger than the targeted sparse number of points $d$ and its points are representative. In this paper, we set the size to $2d$. This can avoid the overfitting shortage, and also improve the generalization ability of the multi-task sparsity regularizer. For the algorithm to select the rough global structure, the MPGR algorithm is suitable because of its superiority on representative-subset selection.

As shown in Fig. 1, we employ the multi-task sparsity regularizer and the improved multi-task sparsity regularizer with the MPGR algorithm, respectively, to choose four points. The top point is set as the initial point. The result demonstrates that the subset selected with the improved multi-task sparsity regularizer is more representative. Fig. 1 is only for illustrative purpose, and more experiments on real data sets will be performed in the next section.

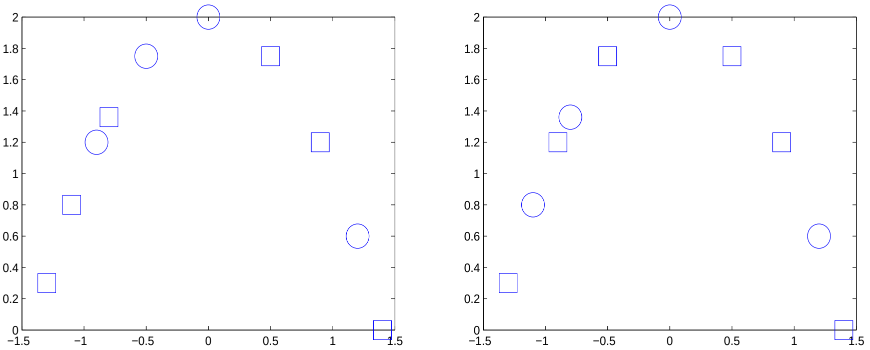

**Fig. 1.** Different selections by the multi-task sparsity regularizer and the improved multi-task sparsity regularizer. Circle points are already-selected points. Square points are candidate points that have not been selected.

The other shortage is that the multi-task sparsity regularizer disregards the situation that a big difference of the range of input values between tasks exists. Since the Euclidean distance is adopted to reflect similarities in formula (1), when the ranges of the input values among tasks are very different, the multi-task sparsity regularizer will possibly have a negative effect on the subset selection. See Fig. 2. Suppose tasks A and B are selecting their sparse subsets, and just three nearest neighbors are considered in formula (1). When evaluating the consistency of points in task A at this selection, points E, F and G are counted in because they are closer to points of task A. In addition, these three points would always be the three nearest neighbors of points of task A. This can lead to a critical overfitting problem.

To avoid the problem of the big difference, a straightforward approach is that when evaluating points of task A we make changes to points of task B and let them be in the same numerical range of task A. For every point of task B, we execute this formula $\frac{x^n+m_A^n-m_B^n}{\sigma_B^n/\sigma_A^n}$ for each dimension, where $x^n$ is the $n$th

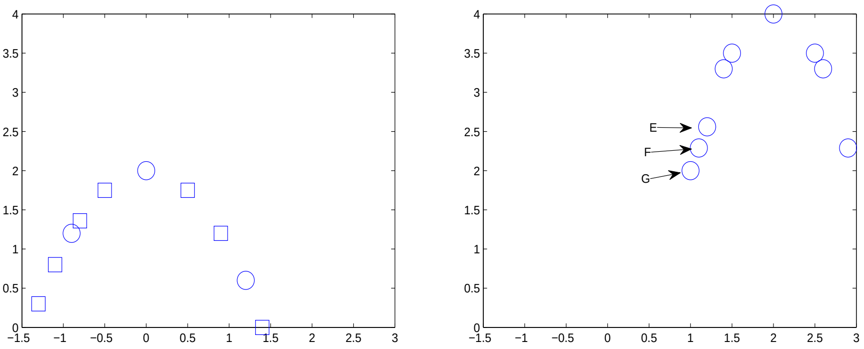

**Fig. 2.** An illustration of the problem of numerical scale differences that the multi-task sparsity regularizer has to face. Task A (left) is selecting the next point with a related task (right). Points E, F and G will always be the three nearest neighbors of points of task A.

dimension of one data point in task B, $m_A^N$ and $m_B^n$ are mean values of the $n$th dimension of data sets of task A and B, respectively, and $\sigma_A^n$ and $\sigma_B^n$ are standard deviations of the $n$th dimension of data sets of task A and B, respectively. The shortage is overcome but this method consumes too much time. For each pair of tasks, the formula would be executed one time. And when the number of tasks is as large as $M$, it would need to execute the formula $M(M-1)$ times. A better way to solve this is to execute a normalization formula $\frac{x^n - m^n}{\sigma^n}$ for each dimension of each task to let them all be in the same numerical range, where $m$ and $\sigma$ are the mean value and the standard deviation, respectively. By this method, each task only needs to execute the formula one time, and the formula would be executed $M$ times when the number of tasks is $M$.

With the two shortages overcome, the obtained method is called the improved multi-task sparsity regularizer. In conclusion, it consists of three steps. First, normalization for each data set on each dimension is performed to make them in the same range. The formula $\frac{x^n - m^n}{\sigma^n}$ is utilized. Second, it utilizes MPGR to select a rough global structure for each task. Last, it replaces the already-selected points with the rough global structures when calculating the distance of a data point to its $k$ nearest neighbors from other tasks. Compared to the previous multi-task sparsity regularizer, it solves the overfitting problem not only to the initial points but also that derived from the big scale difference. At the same time, points that are considered in formula (1) would be more representative. Thus, it improves the generalization ability of the previous regularizer. For comparison, we apply the improved multi-task sparsity regularizer to MPGR and MPGRO to get multi-task sparse Gaussian processes IrMTMPGR and IrMTMPGRO, respectively, in the same way as the previous multi-task sparsity regularizer.

## 4 Experiments

We evaluate our methods on four real data sets, Landmine, Concrete slump, MONK and Energy efficiency. To demonstrate the generalization ability of our methods, these data sets include problems for binary classification, multi-class classification and regression. All the data sets are public, which can be found from the UCI Machine Learning Repository.

We utilize the GPML toolbox [1] to construct Gaussian processes. All the mean functions, covariance functions, likelihood functions and inference methods are selected by the accuracy of one experiment with the whole training set. We conduct experiments ten times on each data set. The training set and the test set are split randomly. The parameters needed for MPGR and MPGRO are set the same as in [13]. To make $\lambda$ easy to set, we split it into two parameters, $\lambda = \alpha \times \beta$. First, we use a normalization parameter $\alpha$ which is set to be the ratio of the maximum values of the formula of MPGR or MPGRO and our regularizers, to make them in the same range. Then, we set $\beta$ to control the relative proportion, which is selected from $\{1/2, 2/3, 1, 3/2, 2\}$. We proceed to choose all the parameters by five-fold cross-validation on the training set. Then we evaluate performance on the test set. We conduct our experiments by a computer with dual 2.53 GHz CPUs and one GB memory.

The error rates and average time are utilized to evaluate the performance of different methods. For classification, the error rate is measured by the rate of misclassification. For regression, the error rate is measured by mean absolute relative error (MARE), which is defined as

$$MARE = \frac{1}{\ell_x} \sum_i \left| \frac{x_i^* - x_i}{x_i} \right|, \tag{3}$$

where $x_i$ is the real value of the $i$th point, $x_i^*$ is its predicted value and $\ell_x$ is the total number of points. The average time is the mean of time on ten experiments, including subset selection and constructing Gaussian processes with the subsets. We choose two kinds of multi-task sparse Gaussian processes as a baseline. As mentioned before, IVM is extended to MTIVM [9] by sharing the kernel parameters and training set among multiple tasks. For comparison purpose in this paper, 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. MPGRO is also extended to MTMPGRO for contrast.

The Landmine data set is collected from a read landmine field. It is a binary classification problem that has 19 related tasks with 9674 data points in total and each point is represented by a nine-dimensional feature vector. Due to the time constraint, we just randomly choose five tasks for experiments, and for each task we randomly choose 320 points as the training data, 80 for test and the subset size is 40. It is necessary to mention that the Landmine data set is an unbalanced data set where the ratio of +1 class is only 6.18% on average. The Concrete slump

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

| Method | Error rate (%) | Time (s) |
|---|---|---|
| MTMPGR | $7.4 \pm 2.5$ | 104.9 |
| rMTMPGR | $6.9 \pm 1.5$ | 105.6 |
| IrMTMPGR | $\mathbf{6.5 \pm 1.4}$ | 172.9 |
| MTMPGRO | $22.5 \pm 7.7$ | 510.4 |
| rMTMPGRO | $50.1 \pm 8.5$ | **46.2** |
| IrMTMPGRO | $47.3 \pm 10.6$ | 238.7 |

**Table 1.** Experimental results on the Landmine data set.

data set including 103 data points concerns a regression problem. The slump flow of concrete is not only determined by the water content, but also influenced by other concrete ingredients. Seven feature attributes are used to predict three output variables. We apply this multi-output data to multi-task experiments by setting each output as a task. In this setting, from the perspective of inputs, all three tasks are closely consistent both locally and globally. We randomly choose 80 points as the training data, 23 for test, and the subset size is 50. The MONK data set is a famous classification problem which is the basis of the first international comparison of learning algorithms. It rely on an artificial robot domain, in which robots are described by six different attributes. Three tasks are included, and all their data are randomly selected from 432 robots. They are different in task size, feature setting, misclassification ratio and noise. Their sizes are 124, 169 and 122, respectively. We randomly choose 120 points as the training data, 80 for test, and the subset size is 30. The Energy efficiency data set is provided by a study which attempts to assess the heating load and cooling load requirements of buildings (that is, energy efficiency) as a function of building parameters. The data set contains eight features to predict the two responses. It is similar to the concrete slump data set which is also a multi-output regression problem. We randomly choose 300 points as the training data, 100 for test, and the subset size is 50. Tables 1∼4 list the experimental results, which show the effectiveness of the improved multi-task regularizer. From the overall prediction performance, just like previous informal analysis, the improved multi-task sparsity regularizer is obviously the best among all the methods for constructing multi-task sparse Gaussian processes. From the perspective of error

| Method | Error rate (%) | Time (s) |
|---|---|---|
| MTMPGR | $16.5 \pm 4.1$ | **1.3** |
| rMTMPGR | $16.4 \pm 2.6$ | 2.3 |
| IrMTMPGR | $14.0 \pm 1.6$ | 4.9 |
| MTMPGRO | $16.4 \pm 4.7$ | 2.4 |
| rMTMPGRO | $14.5 \pm 2.4$ | 1.6 |
| IrMTMPGRO | $\mathbf{13.5 \pm 2.5}$ | 7.1 |

**Table 2.** Experimental results on the Concrete slump data set.

| Method | Error rate (%) | Time (s) |
|---|---|---|
| MTMPGR | $32.1 \pm 6$ | **2.6** |
| rMTMPGR | $\mathbf{23.0 \pm 5.5}$ | 3.0 |
| IrMTMPGR | $34.7 \pm 4.9$ | 5.1 |
| MTMPGRO | $37.6 \pm 4.2$ | 3.0 |
| rMTMPGRO | $26.4 \pm 8.3$ | 2.8 |
| IrMTMPGRO | $24.2 \pm 7.7$ | 7.5 |

**Table 3.** Experimental results on MONK.

| Method | Error rate (%) | Time (s) |
|---|---|---|
| MTMPGR | $12.7 \pm 1.3$ | 14.7 |
| rMTMPGR | $23.1 \pm 5.4$ | 15.7 |
| IrMTMPGR | $14.4 \pm 0.4$ | 25.1 |
| MTMPGRO | $15.3 \pm 2.1$ | 9.4 |
| rMTMPGRO | $11.7 \pm 1.5$ | **6.9** |
| IrMTMPGRO | $\mathbf{11.2 \pm 0.1}$ | 27.9 |

**Table 4.** Experimental results on the Energy efficiency data set.

rates, the best learning algorithms associated with the four data sets almost all utilize the improved multi-task sparsity regularizer. As mentioned before, the Landmine data set is an unbalanced data set. Experimental results in Table 1 show that MTMPGRO works badly on this condition, and the results get worse with the regularizer. This unbalanced case is an open problem worth studying in the future.

## 5 Conclusion

In this paper, we proposed the improved multi-task sparsity regularizer to overcome two shortages of the multi-task sparsity regularizer. We utilized the MPGR algorithm to choose rough global structures and replaced the already-selected points with them in formula (1). Then, we carried out normalization for each data set on each dimension before the subset selection of multiple Gaussian processes. The combined method to get multi-task sparse Gaussian processes is the same as the previous multi-task sparsity regularizer. Experimental results have shown that it indeed improves the performance of the multi-task sparsity regularizer.

As mentioned in the experiment section, the improved multi-task sparsity regularizer combined with MTMPGRO can not perform well on the unbalanced data set. Special considerations on unbalanced data sets will be one of our future research topics. The time consumed by the methods coupled with the improved multitask sparsity regularizer is slightly higher, which will be optimized in the future.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China under Project 61370175 and 61075005, and Shanghai Knowledge Service Platform Project (No.ZF1213).

## References

1. Rasmussen, C., Williams, C.: Gaussian process for machine learning, MIT Press, Cambridge (2006)
2. Sun, S.: Infinite mixtures of multivariate Gaussian processes, in: Proceedings of the International Conference on Machine Learning and Cybernetics, pp. 1011-1016. (2013)
3. Bonilla, E., Chai, K. M., Williams, C. K. I.: Multi-task Gaussian process prediction, in: Proceedings of the Neural Information Processing Systems, pp. 1-8. (2008)
4. Williams, C., Seeger, M.: Using the Nystr*ö*m method to speed up kernel machines, Advances in Neural Information Processing Systems 13, 682-688 (2001)
5. Lawrence, N., Seeger, M., Herbrich, R.: Fast sparse Gaussian process methods: The informative vector machine, Advances in Neural Information Processing Systems 15, 609-616 (2002)
6. Titsias, M.: Variational learning of inducing variables in sparse Gaussian processes, in: Proceedings of the 12th International Workshop on Artificial Intelligence and Statistics, pp. 567-574. (2009)
7. Dhillon, P., Foster, D., Ungar, L.: Minimum description length penalization for group and multi-task sparse learning, Journal of Machine Learning Research 12, 525-564 (2011)
8. Jebara T.: Multitask sparsity via maximum entropy discrimination, Journal of Machine Learning Research 12, 75-110 (2011)
9. Lawrence, N., Platt J.: Learning to learn with the informative vector machine, in: Proceedings of International Conference on Machine Learning, pp. 1-8. (2004)
10. Wang, J., Khardon, R.: Sparse Gaussian processes for multi-task learning, in: Proceedings of Machine Learning and Knowledge Discovery in Databases, pp. 711-727. (2012)
11. Bonilla, E., Agakov, F., Williams, C.: Kernel multi-task learning using task-specific features, in: Proceedings of International Conference on Artificial Intelligence and Statistics, pp. 43-50. (2007)
12. Zhu, J., Sun, S.: Single-task and multitask Gaussian processes, in: Proceedings of the International Conference on Machine Learning and Cybernetics, pp. 1033-1038. (2013)
13. Zhu, J., Sun, S.: Sparse Gaussian processes with manifold-preserving graph reduction, Neurocomputing 138, 99-105 (2014)
14. Sun, S., Hussain, Z., Shawe-Taylor, J.: Manifold-preserving graph reduction for sparse semi-supervised learning, Neurocomputing 124, 13-21 (2014)
15. Sun, S., Multitask learning for EEG-based biometrics, in: Proceedings of the 19th International Conference on Pattern Recognition, pp. 1-4 (2008)