Gaussian Processes Regression with Joint Learning of Precision Matrix

Xiaoyu Miao, Aimin Jiang, and Ning Xu
College of Internet of Things Engineering
Hohai University
Changhzou, China
Emails: mxy@hhu.edu.cn, {jiangam, xuning}@hhuc.edu.cn

***Abstract*—In the traditional Gaussian process regression (GPR), covariance matrix is modeled by a kernel function, which is dominated by a set of hyper-parameters. However, the estimation of such hyper-parameters are generally a highly nonconvex optimization problem, which imposes computational difficulties and undermines the practical performance. To improve the prediction accuracy, we propose in this paper a novel GPR algorithm that introduces the estimate of precision matrix of target values. Covariance and precision matrices are coupled by a regularized approximation error term. In practice, the precision matrix and hyper-parameters are trained by the alternating optimization. Experimental results demonstrate that the performance of the joint-learning formulation is superior to traditional GPR.**

***Index Terms*—Alternating optimization, Bayesian, Gaussian process regression (GPR), joint learning, kernel, precision matrix.**

I. Introduction

Gaussian processes (GPs) are rich distributions over functions, which provide Bayesian kernel-based frameworks for solving supervised learning tasks such as regression and classification [1]. They are widely used in a range of applications, e.g., model predictive control and system analysis [2]-[3], image processing [4], and speech processing [5]. Given training data, GPs generate the posterior distribution of test target values which correspond to test inputs.

In this paper, we focus on Gaussian process regression (GPR), whose objective is to reconstruct the underlying signals by removing contaminating noise [1]. GPR models the similarity of target values by kernel functions and generally assume that data points with similar inputs are likely to have similar target values. A variety of kernel functions, such as *squared exponential* (SE) kernel, *rational quadratic* (RQ) kernel, *periodic kernels* (PE) [1] and *spectral mixture* (SM) kernel [6], have been deployed in GP models. Among them, SE kernel is the most popular one, which describes the relationship of two target values using the Euclidean distance of their corresponding inputs.

This work was supported in part by the National Key Research and Development Program 2018AAA0100800, the National Nature Science Foundation of China under grants 61471157, 61772090, 61701471, and 61801055, the Fundamental Research Funds for the Central Universities of China under grants 2018B23014 and 2018B47114, and the Key Development Program of Jiangsu Province of China under grants BE2017071, BE2017647, and BE2018004-04.

The smoothness and generalization properties of the GP are generally encoded by kernel functions or, essentially, their hyper-parameters. But the optimization of those hyper-parameters is still a challenging task in practice, especially when dealing with some complex kernel functions. To reduce computational complexity of GPR, inducing point methods construct approximate covariance matrices of original data (or inducing points), so that GPR with complex kernels can also be applied in the scenario of big data. Typical approaches of this class include subset of regressors (SoR) [7], fully independent training conditional (FITC) [8], partially independent training conditional (PITC) [9], and structured kernel interpolation (SKI) [10].

In the traditional GPR, hyper-parameters are often learned by Maximum Likelihood Estimate (MLE). But the resulting models are generally nonconvex optimization problems. Thus, unsuitable initial values of hyper-parameters could lead to local points, that are far away from global solutions. As an attempt to overcome this computational difficulty, we introduce in our GPR model a precision matrix, that bridges the kernel covariance matrix and target values in a more efficient way. It is known that a pair of precision and covariance matrices are defined as the inverse of each other. Recovering the structure of an undirected Gaussian graph is equivalent to the recovery of the support of precision matrix [11]. In GPR problems, covariance matrix are dominated by kernel functions. Hence, essentially speaking, the estimation of the corresponding precision matrix is still a challenging problem. In our algorithm, however, these kernel constraints are removed from precision matrix so as to avoid severe performance degradation incurred by inappropriate initial hyper-parameters. Covariance and precision matrices are coupled by a regularization term measuring their approximation error. In practice, covariance and precision matrices can be optimized alternatively.

The paper is organized as follows. In Section II, we first review the traditional GPR. Then, the new GPR problem is formulated. An alternating optimization algorithm is further developed in Section II. Experimental results are presented in Section III. Section IV concludes the paper finally.

II. Proposed Algorithm

A. Traditional GPR

For a GP, any finite number of random variables have a joint Gaussian distribution. Let $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \dots\ \mathbf{x}_N]^T \in \mathbb{R}^{N\times D}$ denote N training inputs of D dimension and $\mathbf{y} \in \mathbb{R}^N$ consists of N target values. We model each target value y_i as the output of an underlying function

$$y_i = g(\mathbf{x}_i) + \varepsilon. \tag{1}$$

For the additive noise ε, one generally supposes that it follows the zero-mean isotropic Gaussian distribution, i.e.,

$$p(\varepsilon) \sim \mathcal{N}\left(0, \sigma^2\right). \tag{2}$$

Then, the PDF of $\mathbf{y}$ is given by

$$p(\mathbf{y}) \sim \mathcal{N}\left(\mu, \mathbf{K} + \sigma^2\mathbf{I}\right) \tag{3}$$

where $\mathbf{K} \in \mathbb{R}^{N\times N}$ denotes the covariance matrix. Entry $K_{i,j}$ of $\mathbf{K}$ represents the correlation of y_i and y_j, which can be expressed in the kernel form as

$$K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j). \tag{4}$$

The task of the GPR is that, given $\mathbf{X}$ and $\mathbf{y}$, one needs to estimate hyper-parameters of (3), such that $\mathbf{y}$ can best fit (3) under some criterion. Let $\mathbf{y}^* \in \mathbb{R}^M$ be target values of test inputs $\mathbf{X}^* \in \mathbb{R}^{M\times D}$. Then, the joint Gaussian process distribution of $\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix}$ is obtained by

$$p\left(\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix}\right) \sim \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu^* \end{bmatrix}, \begin{bmatrix} \mathbf{K}+\sigma^2\mathbf{I} & \mathbf{K}_{N,*} \\ \mathbf{K}_{N,*}^T & \mathbf{K}_{**}+\sigma^2\mathbf{I} \end{bmatrix}\right), \tag{5}$$

where $\mathbf{K}_{N,*}$ denotes the cross-correlation of $\mathbf{y}$ and $\mathbf{y}^*$, and $\mathbf{K}_{**}$ represents the covariance matrix of $\mathbf{y}^*$. For convenience, we consider in this paper the zero mean function, that is, $\mu = \mathbf{0}$ and $\mu^* = \mathbf{0}$. Then, the posterior distribution of $\mathbf{y}^*$ is expressed as

$$p(\mathbf{y}^* \mid \mathbf{y}) \sim \mathcal{N}(\mathbf{m}, \mathbf{F}) \tag{6a}$$

where the mean $\mathbf{m}$ and covariance $\mathbf{F}$ of the posterior distribution of $\mathbf{y}^*$ are given by

$$\mathbf{m} = \mathbf{K}_{N,*}^T\left(\mathbf{K}+\sigma^2\mathbf{I}\right)^{-1}\mathbf{y} \tag{6b}$$

$$\mathbf{F} = \mathbf{K}_{**} + \sigma^2\mathbf{I} - \mathbf{K}_{N,*}^T\left(\mathbf{K}+\sigma^2\mathbf{I}\right)^{-1}\mathbf{K}_{N,*}. \tag{6c}$$

Given $\mathbf{X}$ and $\mathbf{y}$, hyper-parameters can be estimated under different criteria. In this paper, we consider the MLE method. Its corresponding problem is cast as

$$\min_{\mathbf{C},\sigma^2} f\left(\mathbf{C}^{-1}\right) = \mathbf{y}^T\mathbf{C}^{-1}\mathbf{y} + \log\det(\mathbf{C}) \tag{7a}$$

$$\text{s.t. } \mathbf{C} = \mathbf{K} + \sigma^2\mathbf{I}. \tag{7b}$$

As mentioned before, $\mathbf{K}$ is constructed by a specified kernel function, which is dominated by some hyper-parameters. Since problem (7a) is a non-convex, only local solutions are supposed to be obtained. By defining precision matrix $\mathbf{L} = \mathbf{C}^{-1}$ and removing the kernel constraint 7b, problem (7) can be transformed to

$$\min_{\mathbf{L}} f(\mathbf{L}) = \mathbf{y}^T\mathbf{L}\mathbf{y} - \log\det(\mathbf{L}) \tag{8a}$$

$$\text{s.t. } \mathbf{L} = \mathbf{L}^T. \tag{8b}$$

The above problem is convex. Thus, its optimal solution can be reliably obtained by various optimization techniques. A novel algorithm will be developed in the next subsection based on this observation.

B. Problem Formulation

The proposed joint learning model is formulated as

$$\min_{\mathbf{L},\mathbf{C}} f(\mathbf{L}) + f\left(\mathbf{C}^{-1}\right) + \alpha\left\|\mathbf{L}^{\frac{1}{2}}\mathbf{C}\mathbf{L}^{\frac{1}{2}} - \mathbf{I}\right\|_F^2 \tag{9a}$$

$$\text{s.t. } \mathbf{L} = \mathbf{L}^T \tag{9b}$$

$$\mathbf{C}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma^2\cdot\delta(i-j) \tag{9c}$$

where k denotes a specified kernel function and Dirac delta function δ is defined by

$$\delta(x) = \begin{cases} 1, & x = 0 \\ 0, & x \neq 0. \end{cases} \tag{10}$$

We introduce the third term in (9a) to guarantee that $\mathbf{C} \approx \mathbf{L}^{-1}$. The approximation accuracy is controlled by the regularization parameter α. Comparing with (7), model (9) approaches $\mathbf{L}^{-1}$ by $\mathbf{C}$. Thus, (9) can be considered as a graph-regularized GPR problem. The introduction of the graph Laplacian aims to exploit the global structure among data samples to improve the performance of the proposed algorithm.

C. Alternating Optimization

In practice, we adopt the alternating optimization to solve (9). At each iterative step, we fix either $\mathbf{L}$ or $\mathbf{C}$ and solve for the other one. Major steps are described below.

1) Update $\mathbf{C}$*:* Given $\mathbf{L}^{(l-1)}$, the next step is to update $\mathbf{C}$ by solving

$$\mathbf{w}^{(l)} = \arg\min_{\mathbf{w}} f\left(\mathbf{C}^{-1}\right) + \alpha\left\|\left(\mathbf{L}^{(l-1)}\right)^{\frac{1}{2}}\mathbf{C}\left(\mathbf{L}^{(l-1)}\right)^{\frac{1}{2}} - \mathbf{I}\right\|_F^2 \tag{11a}$$

$$\text{s.t. } \mathbf{C}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma^2\cdot\delta(i-j). \tag{11b}$$

Here, $\mathbf{w}$ represents all the hyper-parameters used in the model. It includes parameters of the specified kernel function k, and also the standard deviation σ of additive Gaussian noise. The above problem is non-convex. We adopt to tackle the above problem the nonlinear conjugate gradients (CG) approach, which is the traditional method in the GPR. It is worth noting that, in each iteration, the initial hyper-parameters of $\mathbf{C}$ in the current iteration should be chosen as those obtained in the last iteration in order to ensure that the objective value of problem (11a) monotonically decreases.

2) Update $\mathbf{L}$*:* Given $\mathbf{C}^{(l)}$ obtained in the previous iteration, $\mathbf{L}$ is updated by solving

$$\mathbf{L}^{(l)} = \arg\min_{\mathbf{L}} f(\mathbf{L}) + \alpha \left\| \mathbf{L}^{\frac{1}{2}} \mathbf{C}^{(l)} \mathbf{L}^{\frac{1}{2}} - \mathbf{I} \right\|_F^2 \tag{12a}$$

$$\text{s.t. } \mathbf{L} = \mathbf{L}^T. \tag{12b}$$

For the ease of notation, we ignore the superscript $^{(l-1)}$ in the subsequent presentation. Note that

$$\left\| \mathbf{L}^{\frac{1}{2}} \mathbf{C} \mathbf{L}^{\frac{1}{2}} - \mathbf{I} \right\|_F^2 = \left\| \mathbf{C}^{\frac{1}{2}} \mathbf{L} \mathbf{C}^{\frac{1}{2}} - \mathbf{I} \right\|_F^2. \tag{13}$$

Furthermore, using $\det(\mathbf{AB}) = \det(\mathbf{A}) \det(\mathbf{B})$ for any square matrices $\mathbf{A}$ and $\mathbf{B}$, the objective function of (12) can be rewritten as

$$\begin{aligned} & f(\mathbf{L}) + \alpha \left\| \mathbf{L}^{\frac{1}{2}} \mathbf{C} \mathbf{L}^{\frac{1}{2}} - \mathbf{I} \right\|_F^2 \\ &= \tilde{\mathbf{y}}^T \tilde{\mathbf{L}} \tilde{\mathbf{y}} - \log\det\left(\tilde{\mathbf{L}}\right) + \alpha \left\| \tilde{\mathbf{L}} - \mathbf{I} \right\|_F^2 + \text{const} \\ &= -\log\det\left(\tilde{\mathbf{L}}\right) + \alpha \left\| \tilde{\mathbf{L}} - \tilde{\mathbf{Y}} \right\|_F^2 + \text{const} \end{aligned} \tag{14}$$

where

$$\tilde{\mathbf{L}} = \mathbf{C}^{\frac{1}{2}} \mathbf{L} \mathbf{C}^{\frac{1}{2}} \tag{15}$$

$$\tilde{\mathbf{y}} = \mathbf{C}^{-\frac{1}{2}} \mathbf{y} \tag{16}$$

$$\tilde{\mathbf{Y}} = \mathbf{I} - \frac{1}{2\alpha} \tilde{\mathbf{y}} \tilde{\mathbf{y}}^T. \tag{17}$$

The resulting problem is convex for $\tilde{\mathbf{L}}$. Now, our problem is simplified to

$$\min_{\tilde{\mathbf{L}}} \; -\log\det\left(\tilde{\mathbf{L}}\right) + \alpha \left\| \tilde{\mathbf{L}} - \tilde{\mathbf{Y}} \right\|_F^2 \tag{18a}$$

$$\text{s.t. } \tilde{\mathbf{L}} = \tilde{\mathbf{L}}^T. \tag{18b}$$

The optimal solution $\tilde{\mathbf{L}}^{(l)}$ to (18) can be obtained by making the derivative of its objective function in with respect to $\tilde{\mathbf{L}}$ to $\mathbf{0}$, yielding

$$-\left(\tilde{\mathbf{L}}^{(l)}\right)^{-1} + 2\alpha \tilde{\mathbf{L}}^{(l)} - 2\alpha \tilde{\mathbf{Y}} = \mathbf{0}. \tag{19}$$

Let eigenvalue decompositions of $\tilde{\mathbf{L}}^{(l)}$ and $\tilde{\mathbf{Y}}$ be $\tilde{\mathbf{L}}^{(l)} = \mathbf{U} \Lambda_L \mathbf{U}^T$ and $\tilde{\mathbf{Y}} = \mathbf{V} \Lambda_Y \mathbf{V}^T$, respectively. Then, (19) can be rewritten as

$$\mathbf{U} \left(2\alpha \Lambda_L - \Lambda_L^{-1}\right) \mathbf{U}^T = 2\alpha \mathbf{V} \Lambda_Y \mathbf{V}^T. \tag{20}$$

Obviously, $\mathbf{U} = \mathbf{V}$ makes the equation satisfied and further leads to

$$2\alpha \Lambda_L - \Lambda_L^{-1} = 2\alpha \Lambda_Y. \tag{21}$$

Eq. (21) essentially represents a set of quadratic equations

$$\lambda_i^2 - \eta_i \lambda_i - \frac{1}{2\alpha} = 0, \tag{22}$$

where λ_i and η_i are the ith eigenvalues of $\tilde{\mathbf{L}}$ and $\tilde{\mathbf{Y}}$, respectively. It always has only one positive root

$$\lambda_i = \frac{\eta_i + \sqrt{\eta_i^2 + 2/\alpha}}{2}. \tag{23}$$

Algorithm 1 Alternating optimization algorithm for joint-learning with approximate precision matrix.

1: Set $l = 0$ and initialize the $\mathbf{L}^{(0)}$ and hyper-parameters $\mathbf{w}^{(0)}$ similar to the triditional GPR;
2: **while** not converged **do**
3: Set $l = l + 1$;
4: Given $\mathbf{L}^{(l-1)}$, obtain the $\mathbf{w}^{(l)}$ by solving (11a);
5: Given $\mathbf{h}^{(l)}$, construct $\mathbf{C}^{(l)}$, then use (24) to obtain $\mathbf{L}^{(l)}$;
6: **end while**

TABLE I: Performance of GPR Algorithms Using CO2 Data

N	SEiso		Poly	
	[1]	Proposed	[1]	Proposed
100	-31.562	-31.562	$\mathbf{-32.980}$	$\mathbf{-32.980}$
150	-32.391	-32.391	-33.474	$\mathbf{-33.475}$
200	$\mathbf{-43.980}$	$\mathbf{-43.980}$	-43.534	-43.535

Once obtaining $\tilde{\mathbf{L}}^{(l)}$, the optimal solution to problem (12) is given by

$$\mathbf{L}^{(l)} = \mathbf{C}^{-\frac{1}{2}} \tilde{\mathbf{L}}^{(l)} \mathbf{C}^{-\frac{1}{2}}. \tag{24}$$

The major steps of the alternating optimization algorithm are summarized in Algorithm 1. In practice, this procedure continues until both $\mathbf{L}$ and $\mathbf{C}$ are not significantly updated, i.e.,

$$\max\left\{ \frac{\left\| \mathbf{L}^{(l+1)} - \mathbf{L}^{(l)} \right\|_F}{\left\| \mathbf{L}^{(l)} \right\|_F}, \frac{\left\| \mathbf{C}^{(l+1)} - \mathbf{C}^{(l)} \right\|_F}{\left\| \mathbf{C}^{(l)} \right\|_F} \right\} \le \epsilon, \tag{25}$$

or the iteration number l is larger than a specific value T (e.g., $T = 100$ in our experiments).

III. Experimental Results

A. Experimental Setup

Several sets of real-world data are used to evaluate the prediction performance of the proposed GPR algorithm. The normalized mean square error (NMSE) defined below is adopted to measure the prediction accuracy:

$$\text{NMSE} = 10 \log_{10} \left(\frac{\|\mathbf{m} - \mathbf{y}^*\|_2^2}{\|\mathbf{y}^*\|_F^2} \right). \tag{26}$$

In our experiments, the traditional GPR is also employed for comparison. Three kernel functions are considered by GRP algorithms, which are applied in the (11b).

1) Isotropic SE kernel (SEiso in short):

$$k(\mathbf{x}_i, \mathbf{x}_j) = \delta_f^2 \cdot \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2d^2} \right). \tag{27}$$

TABLE II: Performance of GPR Algorithms Using Boston House Price Data

N	SEiso kernel		SEard kernel		Poly kernel	
	[1]	Proposed	[1]	Proposed	[1]	Proposed
150	-1.197	-1.188	-6.512	$\mathbf{-10.064}$	-6.256	-5.488
300	7.768	7.777	-0.052	$\mathbf{-10.761}$	-4.514	-0.269
455	-6.909	-6.905	-8.252	$\mathbf{-13.627}$	-9.805	-13.474

TABLE III: Performance of GPR Algorithms Using SARCOS Data

N	SEiso kernel		SEard kernel		Poly kernel	
	[1]	Proposed	[1]	Proposed	[1]	Proposed
100	-1.110	$\mathbf{-7.365}$	0.497	-5.805	-3.440	-3.440
200	-1.726	-1.726	-1.524	$\mathbf{-4.143}$	-0.779	-0.844
300	-2.075	-2.074	-3.983	$\mathbf{-4.840}$	-2.611	-2.611
400	-3.672	-3.671	$\mathbf{-4.344}$	$\mathbf{-4.344}$	-2.827	-2.846

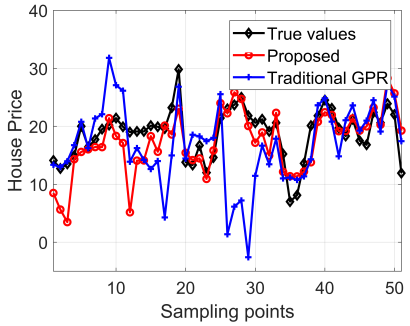

Fig. 1: Prediction results obtained by traditional GPR and the proposed algorithm using SEard kernel for Boston House Price Data ($N = 455$).

Hyper-parameters $\mathbf{w}$ to be estimated include δ_f and d used in (27), and standard deviation σ of additive Gaussian noise ε. It is also known as the radial basis function (RBF) kernel.

2) SE kernel with automatic relevance determination distance measure (SEard in short):

$$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \delta_f^2 \cdot \exp\left\{-\frac{1}{2}\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \mathbf{\Lambda}^{-1}\left(\mathbf{x}_i - \mathbf{x}_j\right)\right\} \tag{28}$$

where Λ is a diagonal matrix. Hyper-parameters $\mathbf{w}$ are δ_f, diagonal elements of Λ, and σ. Actually, the SEiso kernel can be viewed as a special case of the SEard kernel.

3) Polynomial kernel (Poly in short):

$$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \delta_f^2 \cdot \left(c^2 + \mathbf{x}_i^T \mathbf{\Lambda}^{-1} \mathbf{x}_j\right)^p \tag{29}$$

where Λ is a diagonal matrix. Hyper-parameters $\mathbf{w}$ include δ_f, c, diagonal elements of Λ, and σ. Exponent p is generally selected empirically. The Poly kernel essentially belongs to dot product kernel functions, which are invariant to the rotation of coordinates about the origin, but not translations.

The threshold used in the stopping criterion (25) is chosen equal to $\epsilon = 10^{-6}$ and the initial $\mathbf{L}$ in Algorithm 1 is always set to the identity matrix. In the polynomial kernel, we always choose p equal to 2.

B. Prediction for CO2 Data

We first consider CO2 data, which consist of average atmospheric CO2 concentrations at the Mauna Loa Observatory, Hawaii recorded monthly [12]. The data recorded during the first 200 months are used for training, while the data during the remaining 301 months are used for evaluation. Three segments of training data are used in our evaluation. Each of them contains, respectively, 100, 150, and 200 data points. Since the inputs of CO2 data are a series of time instants, i.e., $D = 1$, the SEiso kernel and SEard kernel are essentially identical to each other. In our experiments, we choose α equal to 10^{-4}, 10^{-3}, and 10^{-2} for training data of different lengths, respectively.

Table I lists prediction accuracies of two GPR algorithms. The best results are highlighted in bold font. It can be observed that both the traditional GPR and the proposed algorithm can achieve accurate prediction in this example, if the sufficient data are given for training.

C. Prediction for Boston House Price Data

Boston-housing data containing 455 training data and 51 testing data are used in this example for performance evaluation [13]. In this dataset, 13-dimensional feature data are employed as inputs to GPR models, each representing one factor that can affect target values (i.e., House price). Also, three segments of training data, composed by 150, 300, and 455 data points, are extracted from the original training dataset. Regularization parameter α is chosen, respectively, as 10^{-3}, 10^{-1}, and 10^{-1} in these three cases.

Table II shows the regression accuracies of GPR algorithms when considering different kernel functions. The best results are also marked in bold font. It can be observed that the more complicated SE model can lead to sufficient improvement over the simpler one in this example. Compared to the traditional GPR, our framework achieves significantly improvement by adopting SEard kernel. Fig. 1 shows the prediction results obtained by traditional GPR and the proposed algorithm using SEard kernel for Boston House Price Data.

D. Prediction for SARCOS Data

SARCOS data relate to an inverse dynamics problem for a seven-degrees-of-freedom SARCOS anthropomorphic robot arm [14], which aims to map from a 21-dimensional input space (7 joint positions, 7 joint velocities, 7 joint accelerations) to the corresponding 7 joint torques. We extract, respectively, 100, 200, 300, and 400 data from the original training dataset for the purpose of training. Another 100 data points are chosen for evaluation in all the cases. We set α equal to 10^{-2}, 1, 10^{-3}, and 10^{-4} for SEiso kernel and SEard kernel in different cases. For Poly kernel, another group of α (i.e., $10^{-4}, 10^{-1}, 10^{-5}, 10^{-5}$) are employed.

Table III demonstrates the performance of each algorithm with different kernel functions. The best results are marked

in bold font. Obviously, the proposed algorithm still achieves the best prediction accuracies in all the cases. But, when N is small in this example, the SEiso kernel function leads to better results than SEard and Poly kernels.

IV. Conclusions

A novel algorithm with the framework of joint learning precision and covariance matrices has been presented for the task of GPR. These two matrices are coupled by the approximation error, which introduces a regularization term in our GPR model. The alternating optimization scheme is adopted to update covariance and precision matrices. The closed-form solution is provided in the step of updating precision matrix. Experimental results using public datasets have demonstrated the effectiveness of the proposed algorithm.

References

[1] C. E. Rasmussen, C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[2] G. Cao, E. M. -. Lai and F. Alam, "Gaussian process model predictive control of unknown non-linear systems," *IET Control Theory & Applications*, vol. 11, no. 5, pp. 703-713, 2017.

[3] G. Chowdhary, H. A. Kingravi, J. P. How and P. A. Vela, "Bayesian Nonparametric Adaptive Control Using Gaussian Processes," *IEEE Trans. Neural Netw Learn Syst*, vol. 26, no. 3, pp. 537-550, March 2015.

[4] H. Wang, X. Gao, K. Zhang and J. Li, "Single-Image Super-Resolution Using Active-Sampling Gaussian Process Regression," *IEEE Trans. Image Process*., vol. 25, no. 2, pp. 935-948, Feb. 2016.

[5] D. Moungsri, T. Koriyama and T. Kobayashi, "Duration prediction using multiple Gaussian process experts for GPR-based speech synthesis," *in Proc. of 19th IEEE Int. Conf. Acoust. Speech Signal Process*., New Orleans, LA, 2017, pp. 5495-5499.

[6] A. G. Wilson, R. P. Adams, "Gaussian process covariance kernels for pattern discovery and extrapolation," in *Proc. of 30th Int. Conf. Machine Learn*., pp. 1067-1075, 2013.

[7] B. W. Silverman, "Some aspects of the spline smoothing approach to non-parametric regression curve fitting," *J. Roy. Statistical Soc.*, vol. 74, no. 1, pp. 1-52, 1985.

[8] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 18, pp. 1257-1264, 2006.

[9] J. Quinonero-Candela, C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *J. Mach. Learn. Res.*, vol. 6, pp. 1939-1959, Dec. 2005.

[10] A. G. Wilson, H. Nickisch, "Kernel interpolation for scalable structured gaussian processes (kiss-gp)," *in Proc. of 32nd Int. Conf. Machine Learn.*, pp. 1775-1784, 2015.

[11] T. T. Cai, W. Liu, H. H. Zhou, "Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation," *Annals of Statistics,* vol. 44, no. 2, pp. 455-488, 2016.

[12] C. D. Keeling, T. P. Whorf, "Atmospheric CO2 records from sites in the SIO air sampling network," *Trends: A Compendium of Data on Global Change. Carbon Dioxide Information Analysis Center*, 2004.

[13] D. J. Harrison and D. L. Rubinfeld, "Hedonic housing prices and the demand for clean air," *J. Environ. Economics and Management*, vol. 5, no. 1, pp. 81-102, March 1978.

[14] S. Vijayakumar, A. D'souza, and S. Schaal, "Incremental online learning in high dimensions," *Neural computat*., vol. 17, no. 12, pp. 2602-2634, 2005.