Sparse Gaussian Processes for Multi-task Learning

Yuyang Wang and Roni Khardon

Tufts University, Medford, MA USA {ywang02|roni}@cs.tufts.edu

Abstract. Multi-task learning models using Gaussian processes (GP) have been recently developed and successfully applied in various applications. The main difficulty with this approach is the computational cost of inference using the union of examples from all tasks. The paper investigates this problem for the grouped mixed-effect GP model where each individual response is given by a fixed-effect, taken from one of a set of unknown groups, plus a random individual effect function that captures variations among individuals. Such models have been widely used in previous work but no sparse solutions have been developed. The paper presents the first sparse solution for such problems, showing how the sparse approximation can be obtained by maximizing a variational lower bound on the marginal likelihood, generalizing ideas from single-task Gaussian processes to handle the mixed-effect model as well as grouping. Experiments using artificial and real data validate the approach showing that it can recover the performance of inference with the full sample, that it outperforms baseline methods, and that it outperforms state of the art sparse solutions for other multi-task GP formulations.

1 Introduction

In multi-task learning one learns multiple related tasks simultaneously, with the intention of getting improved predictive performance for all tasks by taking advantage of the common aspects of the tasks. In this paper we explore Bayesian models especially using Gaussian Processes (GP) where sharing the prior and its parameters among the tasks can be seen to implement multi-task learning [3, 4, 18, 7]. Our focus is on the functional grouped mixed-effect model [5, 17] where each task is modeled as a sum of a group-specific fixed-effect (or mean effect, group effect) shared by all the tasks in the group and a random effect that can be interpreted as representing task specific deviations. In particular, all effects are realizations of zero-mean Gaussian processes. Thus, in this model, tasks share structure through hyper-parameters of the prior and through the group-specific fixed-effect portion. This model and its single center counterpart [5, 7] (the classical mixed-effect GP model) have shown success in a wide range of applications, including geophysics [5], medicine [7] and astrophysics [17]. One of the main difficulties with this model, however, is the computational cost, because while the number of samples per task N_j is small, the total sample size $\sum_j N_j$ can be large, and the typical cubic complexity of GP inference can

be prohibitively large [18]. Some improvement can be obtained when all the tasks share the same sampling points, or when different tasks share many of the input points [6, 7]. However, if the number of distinct sampling points is large the complexity remains high. For example, this is the case in [17] where sample points are clipped to a fine grid to avoid the high cardinality of the example set.

The same problem, handling large samples, has been extensively studied in single task formalizations of GP, where several approaches for so-called *sparse solutions* have been developed [11–13, 15]. These methods approximate the GP with $m \ll N$ support variables (also called inducing variables or pseudo inputs) $\mathcal{X}_m$ and their corresponding function values $\mathbf{f}_m$ and perform inference using this set. In the multi-task GP literature, sparse solutions have been proposed in [4] and [1] for a multi-task GP formulation that is different from the one considered in this paper. A more detailed discussion is given in Section 5.

In this paper, we develop a sparse solution for multi-task learning with GP in the context of the functional grouped mixed-effect model. Specifically, we extend the approach of [15] and develop a variational approximation that allows us to efficiently learn the shared hyper-parameters and choose the support variables. In addition, we show how the variational approximation can be used to perform prediction efficiently once learning has been performed. Our approach is particularly useful *when individual tasks have a small number of samples, different tasks do not share sampling points, and there is a large number of tasks.* Our experiments, using artificial and real data, validate the approach showing that it can recover the performance of inference with the full sample, and that it performs better than simple baseline sparse approaches as well as the sparse convolved multiple output GP [1].

To summarize, our contribution is threefold. First we propose the first sparse learning algorithm for multi-task GP in the context of the functional grouped mixed-effect model. Second, we develop a variational model selection approach for the proposed sparse model. Finally we evaluate the algorithm and several baseline approaches for multi-task GP, showing that the proposed method performs well against state of the art sparse solutions for other multi-task GP formulations.

2 Nonparametric Bayesian Grouped Mixed-effect Model

We start by presenting the model which is closely related to the one in [17]. Consider a set of M tasks where the data for the j-th task is given by $\mathcal{D}^j = \{(\boldsymbol{x}_i^j, y_i^j)\}, i = 1, 2, \cdots, N_j$. Given data $\mathcal{D} = \{\mathcal{D}^j\}$, we are interested in learning the nonparametric Bayesian grouped mixed-effect model and using the model to perform inference. The model captures each task f^j as a sum of a mean effect function chosen from a predefined set of K groups and an individual variation (random effect) specific to the j-th task. More precisely,

Assumption 1 *For each j and $\boldsymbol{x} \in \mathcal{X}$, $f^j(\boldsymbol{x}) = \bar{f}_{z_j}(\boldsymbol{x}) + \tilde{f}^j(\boldsymbol{x}), \quad j = 1, \cdots, M$ where $\{\bar{f}_k\}, k = 1, \cdots, K$ and $\tilde{f}^j$ are zero-mean Gaussian processes with covari-*

ance function $\mathcal{K}_k$ and $\widetilde{\mathcal{K}}$, and $z_j \in \{1, \cdots, K\}$. In addition, $\{\bar{f}_k\}$ and $\tilde{f}^j$ are assumed to be mutually independent.

The generative process is as follows, where **Dir** and **Multi** denote the Dirichlet and the Multinominal distribution respectively.

1. Draw the processes of the mean effect: $\bar{f}_k(\cdot)|\boldsymbol{\theta}_k \sim \mathcal{GP}(0, \mathcal{K}_k(\cdot,\cdot)), \quad k = 1, 2, \cdots, K$;
2. Draw $\boldsymbol{\pi}|\boldsymbol{\alpha}_0 \sim \mathbf{Dir}(\boldsymbol{\alpha}_0)$;
3. For the j-th task (time series);
   - Draw $z_j|\boldsymbol{\pi} \sim \mathbf{Multi}(\boldsymbol{\pi})$;
   - Draw the random effect: $\tilde{f}^j(\cdot)|\tilde{\boldsymbol{\theta}} \sim \mathcal{GP}(0, \widetilde{\mathcal{K}}(\cdot,\cdot))$;
   - Draw $\boldsymbol{y}^j|z_j, f^j, \boldsymbol{x}^j, \sigma_j^2 \sim \mathcal{N}\left(f^j(\boldsymbol{x}^j), \sigma_j^2 \cdot \mathbb{I}_j\right)$, where $f^j = \bar{f}_{z_j} + \tilde{f}^j$ and where to simplify the notation $\mathbb{I}_j$ stands for $\mathbb{I}_{N_j}$.

When $K = 1$, our model reduces to the classical mixed-effect GP model [7, 5]. Due to the analogy to clustering we sometimes refer to the latent fixed-effect functions as "centers". Let $\breve{\boldsymbol{x}}$ be the concatenation of the examples from all tasks $\breve{\boldsymbol{x}} = (\boldsymbol{x}_i^j)$, and similarly let $\breve{\boldsymbol{y}} = (\boldsymbol{y}_i^j)$, where $i = 1, 2, \cdots, N_j, j = 1, 2, \cdots, M$ and $N = \sum_j N_j$. When the assignment of tasks into groups is known, the likelihood decomposes into separate terms and the predictive distribution can be obtained directly.

$$\begin{aligned} \mathbb{E}(f^j(\boldsymbol{x}^*)|\mathbf{Z}) &= \mathbf{C}^\dagger(\boldsymbol{x}^*, \breve{\boldsymbol{x}})(\mathbf{C}^\dagger(\breve{\boldsymbol{x}}, \breve{\boldsymbol{x}}) + \mathcal{I})^{-1}\breve{\boldsymbol{y}} \\ \mathbf{Cov}(f^j(\boldsymbol{x}^*)|\mathbf{Z}) &= \mathbf{C}^\dagger(\boldsymbol{x}^*, \boldsymbol{x}^*) - \mathbf{C}^\dagger(\boldsymbol{x}^*, \breve{\boldsymbol{x}})(\mathbf{C}^\dagger(\breve{\boldsymbol{x}}, \breve{\boldsymbol{x}}) + \mathcal{I})^{-1}\mathbf{C}^\dagger(\breve{\boldsymbol{x}}, \boldsymbol{x}^*), \end{aligned} \tag{1}$$

where the covariance matrix $\mathbf{C}^\dagger$ is given by $\mathbf{C}^\dagger((\boldsymbol{x}_i^j), (\boldsymbol{x}_k^l)) = \delta_{z_j, z_l}\mathcal{K}_{z_j}(\boldsymbol{x}_i^j, \boldsymbol{x}_k^l) + \delta_{jl} \cdot \widetilde{\mathcal{K}}(\boldsymbol{x}_i^j, \boldsymbol{x}_k^l)$, and $\mathcal{I} = \bigoplus_j \sigma_j^2 \mathbb{I}_j$ ($\bigoplus$ denotes the matrix direct sum). The marginal distribution is $\Pr(\breve{\boldsymbol{y}}|\breve{\boldsymbol{x}}) \sim \mathcal{N}(\mathbf{0}, \mathbf{C}^\dagger(\breve{\boldsymbol{x}}, \breve{\boldsymbol{x}}) + \mathcal{I})$, which can be used for model selection when group membership is known or if our model only allows for one group. This model works well in that sharing the information improves predictive performance but, as the number of tasks grows, the dimension N increases leading to slow inference scaling as $\mathcal{O}(N^3)$. In other words, even though each task may have a very small sample, the multi-task inference problem becomes infeasible when the number of tasks is large. This holds even if we have the group structure.

For single task GP regression, in order to reduce the computational cost, several sparse GP approaches have been proposed [11–13, 15]. In general, these methods approximate the GP with a small number $m \ll N$ of support variables and perform inference using this subset and the corresponding function values $\mathbf{f}_m$. Different approaches differ in how they choose the support variables and the simplest approach is to choose a random subset of the given data points. Recently, Titsias [15] introduced a sparse method based on variational inference using a set $\mathcal{X}_m$ of support variables, which are independent from the training points. In this approach, the support variables $\mathcal{X}_m$ are chosen to maximize a variational lower bound on the marginal likelihood, therefore providing a clear

methodology for the choice of the support set. Later, [2] extended this idea to derive variational approximation for the sparse convolved multiple output GPs.

Developing a sparse solution for our model is significantly more complex than the single task case because of the need to perform inference over multiple tasks, and even more so because the group structure is not known in advance. In this paper, we propose a variational method to solve the learning problem for the mixture model (both full and sparse) as well as choosing the optimal support variables for the sparse model. As in the case of sparse methods for single task GP, we want to introduce a small set of m auxiliary support variables $\mathcal{X}_m$ and base the learning and inference on these points. For the multi-task case, each $\tilde{f}^j(\cdot)$ is specific to the j-th task. Therefore, it makes sense to induce values only for the fixed-effect portion. Our sparse model picks a separate set $\mathcal{X}_m^k$ for each group and uses the fixed-effect portion $\boldsymbol{\eta}_k = \bar{f}_k(\mathcal{X}_m^k)$ for inference. The details of this construction for learning and for prediction are developed in the next two sections.

3 Learning the Sparse Model

In this section we show how to perform the learning via variational approximation. As mentioned above, for the k-th mixed-effect (or center), we introduce m_k auxiliary inducing support variables $\mathcal{X}_m^k$ and the hidden variable $\boldsymbol{\eta}_k = \bar{f}_k(\mathcal{X}_m^k)$, which is the value of k-th fixed-effect function evaluated at $\mathcal{X}_m^k$.

Let $\mathfrak{f}_k = \bar{f}_k(\breve{\boldsymbol{x}}) \in \mathbb{R}^N$ denote the function values of the k-th mean effect so that $\mathbf{f}_k^j = \bar{f}_k(\boldsymbol{x}^j) \in \mathbb{R}^{N_j}$ is the sub-vector of $\mathfrak{f}_k$ corresponding to the j-th task. Let $\tilde{\mathbf{f}}^j = \tilde{f}(\boldsymbol{x}^j) \in \mathbb{R}^{N_j}$ be the values of the random effect at $\boldsymbol{x}^j$. Denote the collection of the hidden variables as $\mathfrak{F} = \{\mathfrak{f}_k\}, \widetilde{\mathcal{F}} = \{\tilde{\mathbf{f}}^j\}, \mathbf{H} = \{\boldsymbol{\eta}_k\}$, $\mathbf{Z} = \{z_j\}$, and $\boldsymbol{\pi}$. In addition let $\mathbf{c}_{*j}^k = \mathcal{K}_k(\boldsymbol{x}^*, \boldsymbol{x}^j)$, $\mathbf{C}_{jj}^k = \mathcal{K}_k(\boldsymbol{x}^j, \boldsymbol{x}^j), \mathbf{C}_{jk} = \mathcal{K}_k(\boldsymbol{x}^j, \mathcal{X}_m^k)$ and $\mathbf{C}_{kk} = \mathcal{K}_k(\mathcal{X}_m^k, \mathcal{X}_m^k)$, and similarly $\widetilde{\mathbf{c}}_{*j} = \widetilde{\mathcal{K}}(\boldsymbol{x}^*, \boldsymbol{x}^j), \widetilde{\mathbf{C}}_{jj} = \widetilde{\mathcal{K}}(\boldsymbol{x}^j, \boldsymbol{x}^j)$ and $\widehat{\mathbf{C}}_{jj} = \widetilde{\mathbf{C}}_{jj} + \sigma_j^2 \mathbb{I}_j$ where $\mathbb{I}_j$ stands for $\mathbb{I}_{N_j}$.

To learn the sparse model we need to maximize the marginal likelihood $\Pr(\breve{\boldsymbol{y}}|\breve{\boldsymbol{x}})$, which cannot be evaluated directly. In the following we develop a variational lower bound for this quantity. To this end, we need the complete data likelihood and the variational distribution. The complete data likelihood is given by

$$\Pr(\breve{\boldsymbol{y}}, \mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}, \mathbf{Z}, \boldsymbol{\pi}) = \Pr(\breve{\boldsymbol{y}}|\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{Z}) \Pr(\mathfrak{F}|\mathbf{H}) \Pr(\mathbf{Z}|\boldsymbol{\pi}) \Pr(\boldsymbol{\pi}) \Pr(\widetilde{\mathcal{F}}) \Pr(\mathbf{H}), \quad (2)$$

$$\Pr(\mathbf{H}) = \prod_{k=1}^{K} \Pr(\boldsymbol{\eta}_k),\ \Pr(\widetilde{\mathcal{F}}) = \prod_{j=1}^{M} \Pr(\tilde{\mathbf{f}}^j), \Pr(\boldsymbol{\pi}) = \mathbf{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0),\ \Pr(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{j=1}^{M}\prod_{k=1}^{K} \pi_k^{z_{jk}}$$

$$\Pr(\mathfrak{F}|\mathbf{H}) = \prod_{k=1}^{K} \Pr(\mathfrak{f}_k|\boldsymbol{\eta}_k),\ \Pr(\breve{\boldsymbol{y}}|\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{Z}) = \prod_{j=1}^{M}\prod_{k=1}^{K} \left[\Pr(\boldsymbol{y}^j|\tilde{\mathbf{f}}^j, \mathfrak{f}_k)\right]^{z_{jk}}$$

where, as usual, $\{z_{jk}\}$ represent z_j as a unit vector.

Next we approximate the true posterior $\Pr(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}, \mathbf{Z}, \boldsymbol{\pi}|\breve{\boldsymbol{y}})$ on the hidden variables using the following variational distribution

$$q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}, \mathbf{Z}, \boldsymbol{\pi}) = q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}|\mathbf{Z})q(\mathbf{Z})q(\boldsymbol{\pi}) \tag{3}$$

where $q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}|\mathbf{Z}) = \Pr(\widetilde{\mathcal{F}}|\mathfrak{F}, \mathbf{Z}, \breve{\boldsymbol{y}}) \Pr(\mathfrak{F}|\mathbf{H})\boldsymbol{\Phi}(\mathbf{H})$, which equals

$$\prod_{j=1}^{M}\prod_{k=1}^{K}\left[\Pr(\tilde{\mathbf{f}}^j|\mathfrak{f}_k, \boldsymbol{y}^j)\right]^{z_{jk}} \prod_{k=1}^{K}\Pr(\mathfrak{f}_k|\boldsymbol{\eta}_k)\phi(\boldsymbol{\eta}_k).$$

This generalizes the variational form used by [15] to handle the multiple tasks, their grouping and the individual variations of each task. One can see that the variational distribution is not completely factorized (i.e., some dependencies are preserved) but also not completely in free form in that the value of some of the factors is already determined. In particular, $q(\cdot)$ preserves the exact form of $\Pr(\tilde{\mathbf{f}}^j|\mathfrak{f}_k, \boldsymbol{y}^j)$ and in using $\Pr(\mathfrak{f}_k|\boldsymbol{\eta}_k)$ it preserves some information but implicitly assumes that $\boldsymbol{\eta}_k$ is a sufficient statistic for $\mathfrak{f}_k$. The free form $\phi(\boldsymbol{\eta}_k)$ corresponds to $\Pr(\boldsymbol{\eta}_k|\mathcal{D})$ but allows it to diverge from this value to compensate for the assumption that $\boldsymbol{\eta}_k$ is sufficient. Notice that we are not making any assumption about the sufficiency of $\boldsymbol{\eta}_k$ in the generative model and the approximation is entirely due to the variational distribution. An additional assumption is needed in the next section to derive a simplified form of the predictive distribution.

The variational lower bound, denoted as F_V, is given by:

$$\begin{aligned}
\Pr(\breve{\boldsymbol{y}}|\breve{\boldsymbol{x}}) \geqslant F_V &= \int q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}, \mathbf{Z}, \boldsymbol{\pi}) \times \log\left[\frac{\Pr(\breve{\boldsymbol{y}}, \mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}, \mathbf{Z}, \boldsymbol{\pi})}{q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}, \mathbf{Z}, \boldsymbol{\pi})}\right] d\mathfrak{F}\, d\widetilde{\mathcal{F}}\, d\mathbf{H}\, d\mathbf{Z}\, d\boldsymbol{\pi} \\
&= \int q(\mathbf{Z})q(\boldsymbol{\pi}) \log\left[\frac{\Pr(\boldsymbol{\pi})\Pr(\mathbf{Z}|\boldsymbol{\pi})}{q(\mathbf{Z})q(\boldsymbol{\pi})}\right] d\boldsymbol{\pi} d\mathbf{Z} \\
&\quad + \int q(\mathbf{Z})q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}|\mathbf{Z}) \log\left[\frac{\Pr(\breve{\boldsymbol{y}}|\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{Z})\Pr(\mathfrak{F}|\mathbf{H})\Pr(\widetilde{\mathcal{F}})\Pr(\mathbf{H})}{q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}|\mathbf{Z})}\right] d\mathfrak{F} d\widetilde{\mathcal{F}} d\mathbf{H} d\mathbf{Z}
\end{aligned}$$

After some algebraic manipulation, the variational lower bound can be rewritten as follows.

$$\begin{aligned}
F_V = &\int q(\mathbf{Z})q(\pi) \log\left[\frac{\Pr(\boldsymbol{\pi})\Pr(\mathbf{Z}|\boldsymbol{\pi})}{q(\mathbf{Z})q(\boldsymbol{\pi})}\right] d\boldsymbol{\pi} d\mathbf{Z} \\
&+ \int q(\mathbf{Z})\left[\int \prod_{k=1}^{K}\phi(\boldsymbol{\eta}_k)\left\{\log G(\mathbf{Z}, \mathbf{H}, \breve{\boldsymbol{y}}) + \sum_{k=1}^{K}\log\left[\frac{\Pr(\boldsymbol{\eta}_k)}{\phi(\boldsymbol{\eta}_k)}\right]\right\} d\mathbf{H}\right] d\mathbf{Z}
\end{aligned} \tag{4}$$

where $\log G(\mathbf{Z}, \mathbf{H}, \breve{\boldsymbol{y}})$ equals

$$\int \Pr(\widetilde{\mathcal{F}}|\mathfrak{F}, \mathbf{Z}, \breve{\boldsymbol{y}}) \Pr(\mathfrak{F}|\mathbf{H}) \log\left[\prod_{j=1}^{M}\prod_{k=1}^{K}\left[\frac{\Pr(\boldsymbol{y}^j|\mathfrak{f}_k, \tilde{\mathbf{f}}^j)\Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j|\mathfrak{f}_k, \boldsymbol{y}^j)}\right]^{z_{jk}}\right] d\mathfrak{F} d\widetilde{\mathcal{F}}.$$

In Section 3.1, we show that $\log G(\mathbf{Z}, \mathbf{H}, \breve{\boldsymbol{y}})$ can be decomposed as $\log G(\mathbf{Z}, \mathbf{H}, \breve{\boldsymbol{y}}) = \sum_{j=1}^{M}\sum_{k=1}^{K} z_{jk} \log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j)$, where

$$\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j) = \log\left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{C}}_{jj})\right] - \frac{1}{2}\mathbf{Tr}\left[(\mathbf{C}_{jj}^k - \boldsymbol{Q}_{jj}^k)\widehat{\mathbf{C}}_{jj}^{-1}\right], \tag{5}$$

where $\boldsymbol{\alpha}_j^k = \mathbf{C}_{jk}\mathbf{C}_{kk}^{-1}\boldsymbol{\eta}_k$ and $\boldsymbol{Q}_{jj}^k = \mathbf{C}_{jk}\mathbf{C}_{kk}^{-1}\mathbf{C}_{kj}$.

To optimize the parameters we use the variational EM algorithm. In the **Variational E-Step**, we estimate $q^*(\mathbf{Z}), q^*(\boldsymbol{\pi})$ and $\{\phi^*(\boldsymbol{\eta}_k)\}$.

To get the variational distribution $q^*(\mathbf{Z})$, we take derivative of F_V w.r.t. $q(\mathbf{Z})$ and set it to 0. Solving for $q(\mathbf{Z})$, we get

$$q^*(\mathbf{Z}) = \prod_{j=1}^{M}\prod_{k=1}^{K} r_{jk}^{z_{jk}}, \qquad r_{jk} = \frac{\rho_{jk}}{\sum_{k=1}^{K}\rho_{jk}}$$
$$\log \rho_{jk} = \mathbb{E}_{q(\boldsymbol{\pi})}[\log \pi_k] + \mathbb{E}_{\phi(\boldsymbol{\eta}_k)}[\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j)],$$

where $\mathbb{E}_{q(\boldsymbol{\pi})}[\log \pi_k] = \Psi(\alpha_k) - \Psi(\sum_k \alpha_k)$ where Ψ is the digamma function, α_k is defined below, and $\mathbb{E}_{\phi(\boldsymbol{\eta}_k)}[\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j)]$ is given below in (16).

Similarly, $q^*(\boldsymbol{\pi})$ can be obtained as $q^*(\boldsymbol{\pi}) = \mathbf{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha})$ where $\alpha_k = \alpha_0 + N_k$ and $N_k = \sum_{j=1}^{K} r_{jk}$.

The final step is to get the variational distribution of $\phi^*(\boldsymbol{\eta}_k), k = 1, \cdots, K$. Notice that only the second term of F_V is a function of $\phi(\boldsymbol{\eta}_k)$ and it can be rewritten as

$$\sum_{k=1}^{K}\int \phi(\boldsymbol{\eta}_k)\left\{\left[\sum_{j=1}^{M}\mathbb{E}_{q(\mathbf{Z})}[z_{jk}]\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j)\right] + \log\left[\frac{\Pr(\boldsymbol{\eta}_k)}{\phi(\boldsymbol{\eta}_k)}\right]\right\} d\boldsymbol{\eta}_k. \tag{6}$$

Thus, our task reduces to find each $\phi^*(\boldsymbol{\eta}_k)$ separately. Taking the derivative of (6) w.r.t. $\phi(\boldsymbol{\eta}_k)$ and setting it to 0, we obtain

$$\phi^*(\boldsymbol{\eta}_k) \propto \prod_{j=1}^{M}\left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{C}}_{jj})\right]^{\mathbb{E}_{q(\mathbf{Z})}[z_{jk}]}\Pr(\boldsymbol{\eta}_k). \tag{7}$$

Thus, we have

$$\phi^*(\boldsymbol{\eta}_k) \propto \exp\left\{-\frac{1}{2}\boldsymbol{\eta}_k^T\left(\mathbf{C}_{kk}^{-1}\boldsymbol{\Phi}\mathbf{C}_{kk}^{-1}\right)\boldsymbol{\eta}_k + \boldsymbol{\eta}_k^T\left(\mathbf{C}_{kk}^{-1}\sum_{j=1}^{M} r_{jk}\mathbf{C}_{kj}[\widehat{\mathbf{C}}_{jj}]^{-1}\boldsymbol{y}_j\right)\right\},$$

where $\boldsymbol{\Phi} = \mathbf{C}_{kk} + \sum_{j=1}^{M} r_{jk}\mathbf{C}_{kj}[\widehat{\mathbf{C}}_{jj}]^{-1}\mathbf{C}_{jk}$. Completing the square yields the Gaussian distribution $\phi^*(\boldsymbol{\eta}_k) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where

$$\boldsymbol{\mu}_k = \mathbf{C}_{kk}\boldsymbol{\Phi}^{-1}\sum_{j=1}^{M} r_{jk}\mathbf{C}_{kj}[\widehat{\mathbf{C}}_{jj}]^{-1}\boldsymbol{y}_j, \quad \boldsymbol{\Sigma}_k = \mathbf{C}_{kk}\boldsymbol{\Phi}^{-1}\mathbf{C}_{kk}. \tag{8}$$

In the **Variational M-Step**, based on the previous estimated variational distribution, we wish to find hyperparameters that maximize the variational lower bound F_V. The terms that depend on the hyperparameters $\boldsymbol{\Theta}$ and the inducing variables $\mathcal{X}_m = \{\mathcal{X}_m^k\}$ are given in (6). Therefore, using (5) again, we can express $F_V(\mathcal{X}_m, \boldsymbol{\Theta})$ as

$$\sum_{k=1}^{K}\mathbb{E}_{\phi^*(\boldsymbol{\eta}_k)}\left\{\log\left[\frac{\prod_j\left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{C}}_{jj})\right]^{r_{jk}}\Pr(\boldsymbol{\eta}_k)}{\phi^*(\boldsymbol{\eta}_k)}\right]\right\} - \frac{1}{2}\sum_{k,j}^{m} r_{jk}\mathbf{Tr}\left[(\mathbf{C}_{jj}^k - \boldsymbol{Q}_{jj})\widehat{\mathbf{C}}_{jj}^{-1}\right].$$

From (7), we know that the term inside the log is constant, and therefore, extracting the log from the integral and cancelling the $\phi^*(\boldsymbol{\eta}_k)$ terms we see that the k'th element of first term is equal to the logarithm of

$$\int \prod_{j=1}^{M} \left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{C}}_{jj})\right]^{r_{jk}} \Pr(\boldsymbol{\eta}_k) d\boldsymbol{\eta}_k. \tag{9}$$

We next show how this multivariate integral can be evaluated. First we can write $\left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{C}}_{jj})\right]^{r_{jk}} = A_{jk}\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, r_{jk}^{-1}\widehat{\mathbf{C}}_{jj})$, where $A_{jk} = (r_{jk})^{\frac{N_j}{2}}(2\pi)^{\frac{N_j(1-r_{jk})}{2}}$ $|\widehat{\mathbf{C}}_{jj}|^{\frac{1-r_{jk}}{2}}$. Thus, we have $\prod_j \left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{C}}_{jj})\right]^{r_{jk}} = \left[\prod_j A_{jk}\right] \prod_j \mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, r_{jk}^{-1}\widehat{\mathbf{C}}_{jj})$. As the first part is not a function of $\boldsymbol{\eta}_k$, for the integration we are only interested in the second part. Since $\breve{\boldsymbol{y}}$ is the concatenation of all $\boldsymbol{y}^j$'s, we can write

$$\prod_{j=1}^{M} \mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, r_{jk}^{-1}\widehat{\mathbf{C}}_{jj}) = \mathcal{N}(\breve{\boldsymbol{y}}|\boldsymbol{\Lambda}_k\mathbf{C}_{kk}^{-1}\boldsymbol{\eta}_k, \widehat{\mathbf{C}}^k), \tag{10}$$

where $\boldsymbol{\Lambda}_k = [\mathbf{C}_{1k}^T, \mathbf{C}_{2k}^T, \cdots, \mathbf{C}_{Mk}^T]^T \in \mathbb{R}^{N,m_k}$ and $\widehat{\mathbf{C}}^k = \bigoplus_{j=1}^{M} r_{jk}^{-1}\widehat{\mathbf{C}}_{jj}^k \in \mathbb{R}^{N,N}$, which is the block diagonal matrix with element $r_{jk}^{-1}\widehat{\mathbf{C}}_{jj}^k$. Therefore, the integral can the written as the following marginal distribution of $\Pr(\breve{\boldsymbol{y}}|k)$,

$$\int \prod_{j=1}^{M} \mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, r_{jk}^{-1}\widehat{\mathbf{C}}_{jj}) \Pr(\boldsymbol{\eta}_k) d\boldsymbol{\eta}_k = \int \mathcal{N}(\breve{\boldsymbol{y}}|\boldsymbol{\Lambda}_k\mathbf{C}_{kk}^{-1}\boldsymbol{\eta}_k, \widehat{\mathbf{C}}^k) \Pr(\boldsymbol{\eta}_k) d\boldsymbol{\eta}_k. \tag{11}$$

Using the fact that $\Pr(\boldsymbol{\eta}_k) = \mathcal{N}(\mathbf{0}, \mathbf{C}_{kk})$ and observing that (10) is a conditional Gaussian, we have $\Pr(\breve{\boldsymbol{y}}|k) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}_k\mathbf{C}_{kk}^{-1}\boldsymbol{\Lambda}_k^T + \widehat{\mathbf{C}}^k)$. Using this form and the portion of A_{jk} that depends on the parameters we get the variational lower bound $F_V(\mathcal{X}_m, \boldsymbol{\Theta})$, which equals

$$\sum_{k=1}^{K} \log \Pr(\breve{\boldsymbol{y}}|k) + \frac{K-1}{2} \sum_{j=1}^{M} \log |\widehat{\mathbf{C}}_{jj}| - \frac{1}{2} \sum_{j,k} r_{jk} \mathbf{Tr}\left[(\mathbf{C}_{jj}^k - \boldsymbol{Q}_{jj})\widehat{\mathbf{C}}_{jj}^{-1}\right]. \tag{12}$$

Notice that when the number of tasks and the number of centers are both 1, we recover the results in [15] provided that the random effect is independent white noise.

Using the ideas in the previous derivation, the direct inference for the full model can also be obtained where $\boldsymbol{\eta}_k$ is substituted with $\mathfrak{f}_k$ and the variational lower bound becomes

$$F_V(\mathcal{X}_m, \boldsymbol{\Theta}) = \sum_{k=1}^{K} \log \mathcal{N}(\breve{\boldsymbol{y}}|\mathbf{0}, \mathbf{C}_{kk} + \widehat{\mathbf{C}}^k) + \frac{K-1}{2} \sum_{j=1}^{M} \log |\widehat{\mathbf{C}}_{jj}|. \tag{13}$$

We have explicitly written the parameters that can be chosen to further optimize the lower bound (12), namely the support inputs $\{\mathcal{X}_m^k\}$, and the set of hyper-parameters $\boldsymbol{\Theta}$ which is composed of $\{\boldsymbol{\theta}_k\}$ and $\{\tilde{\boldsymbol{\theta}}\}$ in $\mathcal{K}_k$ and $\widetilde{\mathcal{K}}$ respectively.

By calculating derivatives of (12) we can optimize the lower bound using a gradient based method. This can be done by making use of the special form of the covariance matrix $\mathbf{\Lambda}_k \mathbf{C}_{kk}^{-1} \mathbf{\Lambda}_k^T + \widehat{\mathbf{C}}^k$, the matrix inversion formula, the chain rule for derivatives, and sequencing the matrix operations appropriately (details omitted due to space constraints). The complexity of evaluating the derivative of (12) is $\mathcal{O}(N \sum_k m_k^2 + \sum_k m_k^3 + \sum_j N_j^3)$. In our implementation, we use stochastic coordinate descent, where at each iteration, one coordinate (parameter) is chosen at random and we perform gradient descent on that coordinate.

3.1 Evaluating $\log G(\mathbf{Z}, \mathbf{H}, \breve{\boldsymbol{y}})$

In this section, we wish to evaluate $\log G(\mathbf{Z}, \mathbf{H}, \breve{\boldsymbol{y}})$, which equals

$$\begin{aligned}
&\int \prod_{l=1}^{M} \prod_{p=1}^{K} \left[\Pr(\tilde{\mathbf{f}}^l | \mathfrak{f}^p, \boldsymbol{y}^l) \right]^{z_{lp}} \prod_{v=1}^{K} \Pr(\mathfrak{f}_v | \boldsymbol{\eta}_v) \times \sum_{j=1}^{M} \sum_{k=1}^{K} z_{jk} \log \left[\frac{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j)} \right] d\mathfrak{F} d\widetilde{\mathcal{F}} \\
&= \sum_{j=1}^{M} \sum_{k=1}^{K} z_{jk} \left[\int \Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j) \Pr(\mathfrak{f}_k | \boldsymbol{\eta}_k) \times \log \left[\frac{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j)} \right] d\mathfrak{f}_k d\tilde{\mathbf{f}}^j \right],
\end{aligned} \tag{14}$$

where the second line holds because in the sum indexed by l and p all the product measures $\prod_{l \neq j, p \neq k} \left[\Pr(\tilde{\mathbf{f}}^l | \mathfrak{f}^p, \boldsymbol{y}^l) \right]^{z_{lp}}$ are integrated to 1, leaving only the $\Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}^k, \boldsymbol{y}^j)$. Denote the term inside the brackets by $\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j)$; this term can be evaluated as

$$\begin{aligned}
&\int \Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j) \Pr(\mathfrak{f}_k | \boldsymbol{\eta}_k) \times \log \left[\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j) \cdot \frac{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k)}{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k)} \right] d\mathfrak{f}_k d\tilde{\mathbf{f}}^j \\
&= \int \Pr(\mathfrak{f}_k | \boldsymbol{\eta}_k) \log \left[\Pr(\boldsymbol{y}^j | \mathfrak{f}_k) \right] d\mathfrak{f}_k = \int \Pr(\mathbf{f}_k^j | \boldsymbol{\eta}_k) \log \left[\Pr(\boldsymbol{y}^j | \mathbf{f}_k^j) \right] d\mathbf{f}_k^j
\end{aligned} \tag{15}$$

where the last line holds because of the independence between $\tilde{\mathbf{f}}^j$ and $\mathfrak{f}_k$. Noticing that evaluating (15) involves marginalization over Gaussians, after some algebraic manipulation, we obtain (5).

Furthermore, marginalizing out $\boldsymbol{\eta}_k$, we get that $\mathbb{E}_{\phi^*(\boldsymbol{\eta}_k)} \log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j)$ equals

$$\log \left[\mathcal{N}(\boldsymbol{y}^j | \boldsymbol{\mu}_k, \widehat{\mathbf{C}}_{jj}) \right] - \frac{1}{2} \mathbf{Tr} \left[\mathbf{C}_{jk} \mathbf{C}_{kk}^{-1} (\mathbf{\Sigma}_k - \mathbf{C}_{kk}) \mathbf{C}_{kk}^{-1} \mathbf{C}_{jk} \widehat{\mathbf{C}}_{jj}^{-1} \right]. \tag{16}$$

4 Prediction Using the Sparse Model

The proposed sparse model can be used for two types of problems. Prediction for existing tasks and prediction for a newly added task. We start with deriving the predictive distribution for existing tasks. Given any task j, our goal is to calculate the predictive distribution $\Pr(f^j(\boldsymbol{x}^*) | \mathcal{D})$ at new input point $\boldsymbol{x}^*$, which can be written as

$$\sum_{k=1}^{K} \Pr(f^j(\boldsymbol{x}^*)|z_{jk}=1,\mathcal{D}) \Pr(z_{jk}=1|\mathcal{D}) = \sum_{k=1}^{K} r_{jk} \Pr(f^j(\boldsymbol{x}^*)|z_{jk}=1,\mathcal{D}). \quad (17)$$

That is, because z_{jk} form a partition we can focus on calculating $\Pr(f^j(\boldsymbol{x}^*)|z_{jk}=1,\mathcal{D})$ and then combine the results using the partial labels. Instead of calculating the full Bayesian prediction, one can use *Maximum A Posteriori* (MAP) by assigning the j-th task to the center c such that $c = \operatorname{argmax} \Pr(z_{jk}=1|\mathcal{D})$. Preliminary experiments (not shown here) show that the full Bayesian approach gives better performance. In the following, we will show how to calculate $\Pr(f^j(\boldsymbol{x}^*)|z_{jk}=1,\mathcal{D})$, i.e. the predictive distribution when $f^j = \bar{f}_k + \tilde{f}_j$.

As described before, the full inference is expensive and therefore we wish to use the variational approximation for the prediction as well. The key idea is that $\boldsymbol{\eta}_k$ contains as much information as $\mathcal{D}$ in terms of making prediction for $\bar{f}_k$. To start with, for each k, it is easy to see that the predictive distribution is Gaussian (conditioned on $z_{jk}=1$) and that it satisfies

$$\begin{aligned} \mathbb{E}[f^j(\boldsymbol{x}^*)|\mathcal{D}] &= \mathbb{E}[\bar{f}_k(\boldsymbol{x}^*)|\mathcal{D}] + \mathbb{E}[\tilde{f}^j(\boldsymbol{x}^*)|\mathcal{D}] \\ \mathbf{Var}[f^j(\boldsymbol{x}^*)|\mathcal{D}] &= \mathbf{Var}[\bar{f}_k(\boldsymbol{x}^*)|\mathcal{D}] + \mathbf{Var}[\tilde{f}^j(\boldsymbol{x}^*)|\mathcal{D}] + 2\mathbf{Cov}[\bar{f}_k(\boldsymbol{x}^*)\tilde{f}^j(\boldsymbol{x}^*)|\mathcal{D}]. \end{aligned} \quad (18)$$

The above equation is more complex than the predictive distribution for single-task sparse GP because of the coupling induced by $\bar{f}_k(\boldsymbol{x}^*)\tilde{f}^j(\boldsymbol{x}^*)|\mathcal{D}$. We next show how this can be calculated via conditioning.

The calculation of the terms in (18) consists of three parts, i.e. $\Pr(\bar{f}_k(\boldsymbol{x}^*)|\mathcal{D})$, $\Pr(\tilde{f}(\boldsymbol{x}^*)|\mathcal{D})$ and $\mathbf{Cov}[\bar{f}_k(\boldsymbol{x}^*)\tilde{f}_j(\boldsymbol{x}^*)|\mathcal{D}]$. Using the approximation of the variational form given in (3), we have the following facts:

1. $\boldsymbol{\eta}_k|\mathcal{D} \sim \phi^*(\boldsymbol{\eta}_k) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are given in (8).

2. $\boldsymbol{\eta}_k$ is sufficient for $\mathfrak{f}_k$, i.e. $\Pr(\mathfrak{f}_k|\boldsymbol{\eta}_k,\mathcal{D}) = \Pr(\mathfrak{f}_k|\boldsymbol{\eta}_k)$. Since we are interested in prediction for each task separately, by marginalizing out the tasks other than j, we also have $\Pr(\mathbf{f}_k^j|\boldsymbol{\eta}_k,\mathcal{D}) = \Pr(\mathbf{f}_k^j|\boldsymbol{\eta}_k)$ and

$$\mathbf{f}_k^j|\boldsymbol{\eta}_k,\mathcal{D} \sim \mathcal{N}\left(\mathbf{C}_{jk}\mathbf{C}_{kk}^{-1}\boldsymbol{\eta}_k, \mathbf{C}_{jj}^k - \mathbf{C}_{jk}\mathbf{C}_{kk}^{-1}\mathbf{C}_{kj}\right). \quad (19)$$

3. For $\tilde{f}^j(\boldsymbol{x}^*)$ we can view $\boldsymbol{y}^j - \mathbf{f}_k^j$ as noisy realizations from the same GP as $\tilde{f}^j(\boldsymbol{x}^j)$.

$$\tilde{f}^j(\boldsymbol{x}^*)|\mathbf{f}_k^j,\mathcal{D} \sim \mathcal{N}\left(\widetilde{\mathbf{c}}_{*j}\left[\widetilde{\mathbf{C}}_{jj}+\sigma_j^2\mathbb{I}_j\right]^{-1}(\boldsymbol{y}^j-\mathbf{f}_k^j), \widetilde{\mathbf{c}}_{**}-\widetilde{\mathbf{c}}_{*j}\left[\widetilde{\mathbf{C}}_{jj}+\sigma_j^2\mathbb{I}_j\right]^{-1}\widetilde{\mathbf{c}}_{j*}\right). \quad (20)$$

In order to obtain a sparse form of the predictive distribution we need to make an additional assumption beyond the variational approximation used for training the model. Specifically, we assume that $\boldsymbol{\eta}_k$ is sufficient for $\bar{f}_k(\boldsymbol{x}^*)$, i.e., $\Pr(\bar{f}_k(\boldsymbol{x}^*)|\boldsymbol{\eta}_k,\mathcal{D}) = \Pr(\bar{f}_k(\boldsymbol{x}^*)|\boldsymbol{\eta}_k)$, implying that

$$\bar{f}(\boldsymbol{x}^*)|\boldsymbol{\eta}_k,\mathcal{D} \sim \mathcal{N}\left(\mathbf{c}_{*k}^k\mathbf{C}_{kk}^{-1}\boldsymbol{\eta}_k, \mathbf{c}_{**}^k - \mathbf{c}_{*k}\mathbf{C}_{kk}^{-1}\mathbf{c}_{k*}\right). \quad (21)$$

The above set of conditional distributions also imply that $\bar{f}_k(\boldsymbol{x}^*)$ and $\tilde{f}^j(\boldsymbol{x}^*)$ are independent given $\boldsymbol{\eta}_k$ and $\mathcal{D}$. Next, we can easily get $\Pr(\bar{f}_k(\boldsymbol{x}^*)|\mathcal{D})$ by marginalizing out $\boldsymbol{\eta}_k|\mathcal{D}$ in (21).

Similarly, we can obtain $\Pr(\tilde{f}(\boldsymbol{x}^*)|\mathcal{D})$ by first calculating $\Pr(\mathbf{f}_k^j|\mathcal{D})$ by marginalizing out $\boldsymbol{\eta}_k|\mathcal{D}$ in (19) and then marginalizing out $\mathbf{f}_k^j|\mathcal{D}$ in (20). Finally, for the remaining term we have $\mathbf{Cov}[\bar{f}_k(\boldsymbol{x}^*)\tilde{f}^j(\boldsymbol{x}^*)|\mathcal{D}] = \mathbb{E}\left[\bar{f}_k(\boldsymbol{x}^*)\tilde{f}^j(\boldsymbol{x}^*)|\mathcal{D}\right] - \mathbb{E}[\bar{f}_k(\boldsymbol{x}^*)|\mathcal{D}]\mathbb{E}[\tilde{f}^j(\boldsymbol{x}^*)|\mathcal{D}]$ where

$$\begin{aligned}\mathbb{E}\left[\bar{f}_k(\boldsymbol{x}^*)\cdot\tilde{f}^j(\boldsymbol{x}^*)|\mathcal{D}\right] &= \mathbb{E}_{\boldsymbol{\eta}_k|\mathcal{D}}\mathbb{E}\left[\bar{f}_k(\boldsymbol{x}^*)\cdot\tilde{f}^j(\boldsymbol{x}^*)|\boldsymbol{\eta}_k,\mathcal{D}\right]\\ &= \mathbb{E}_{\boldsymbol{\eta}_k|\mathcal{D}}\left[\mathbb{E}\left[\bar{f}_k(\boldsymbol{x}^*)|\boldsymbol{\eta}_k\right]\cdot\mathbb{E}[\tilde{f}^j(\boldsymbol{x}^*)|\boldsymbol{\eta}_k,\boldsymbol{y}^j]\right]\end{aligned} \tag{22}$$

where the second line holds because, as observed above, the terms are conditionally independent. The first term $\mathbb{E}\left[\bar{f}^j(\boldsymbol{x}^*)|\boldsymbol{\eta}_k\right]$ can be obtained directly from (21). By marginalizing out $\mathbf{f}_k^j|\boldsymbol{\eta}_k$ in (20) we can get the second term. Finally taking expectation w.r.t. $\phi^*(\boldsymbol{\eta}_k|\mathcal{D})$ can be calculated via properties of the multivariate normal distribution.

We have therefore shown how to calculate the predictive distribution in (18). The complexity of these computations is $\mathcal{O}(K(N_j^3+m^3))$ which is a significant improvement over $\mathcal{O}(KN^3)$ where $N=\sum_j N_j$.

Our model is also useful for making prediction for newly added tasks. Suppose we are given $\{\boldsymbol{x}^{M+1},\boldsymbol{y}^{M+1}\}$ and we are interested in predicting $f^{M+1}(\boldsymbol{x}^*)$. We use the variational procedure to estimate its partial labels w.r.t. different centers $\Pr(z_{M+1,k}=1|\mathcal{D})$ and then (17) can be applied for making the prediction. In the variational procedure we update the parameters for Z_{M+1} but keep all other parameters fixed. Since each task has small number of samples, we expect this step to be computationally cheap.

5 Related Work

Our work is related to [15] particularly in terms of the form of the variational distribution of the inducing variables. However, our model is much more complex than the basic GP regression model. With the mixture model and an additional random effect per task, we must take into account the coupling of the random effect and group specific fixed-effect functions. The technical difficulty that the coupling introduces is addressed in our paper, yielding a generalization that is consistent with single-task solution.

The other related thread comes from the area of GP for multi-task learning. Bonilla et al. proposed a model that learns a shared covariance matrix on features and a covariance matrix for tasks that explicitly models the dependency between tasks [4]. They also presented techniques to speed up the inference by using Nystrom approximation of the kernel matrix and incomplete Cholesky decomposition of the task correlations matrix. Their model, which is known as the linear coregionalization model (LCM) is subsumed by the framework of convolved multiple output Gaussian process [1]. The work of [1] also derives sparse

solutions which are extensions of different single task sparse GP [13, 9]. Our work differs from the above models in that we allow a random effect for each individual task. As we show in the experimental section, this is important in modeling various applications. If the random effect is replaced with independent white noise, then our model is similar to LCM. To see this, from (17), we recognize that the posterior GP is a convex combination of K independent GPs (mean effect). However, our model is capable of prediction for newly added tasks while the models in [4] and [1] cannot. Further, the proposed model can naturally handle *heterotopic* inputs, where different tasks do not necessarily share the same inputs. In [4], each task is required to have same number of samples so that one can use the property of Kronecker product to derive the EM algorithm.

6 Experimental Evaluation

Our implementation of the algorithm makes use of the gpml package [10] and extends it to implement the required functions. For performance criteria we use the standardized mean square error (SMSE) and the mean standardized log loss (MSLL) that are defined in [11]. We compare the following methods. The first four methods use the same variational inference as described in Section 3. They differ in the form of the variational lower bound they choose to optimize.

1. **Direct Inference**: use full samples as the support variables and optimize (13). When $K = 1$, the marginal likelihood is described in Section 2 and the predictive distribution is (1).
2. **Variational Sparse GP for MTL (MT-VAR)**: the proposed approach.
3. **MTL Subset of Datapoints (MT-SD)**: a subset $\mathcal{X}_m^k$ of size m_k is chosen uniformly from the input points from all tasks $\breve{\boldsymbol{x}}$ for each center. The hyper-parameters are selected using $\mathcal{X}_m^k$ (the support variables are fixed in advance) and their corresponding observations by maximizing the variational lower bound. We call this MT-SD as a multi-task version of SD [11], because in the single center case we can use the marginal likelihood and (1) where the subset $\mathcal{X}_m, \mathcal{Y}_m$ and $\boldsymbol{x}^j, \boldsymbol{y}^j$ serve as the full sample (thus discarding other samples).
4. **MTL Projected Process Approximation (MT-PP)**: the variational lower bound of MT-PP is given by the first two terms of (12) ignoring the trace term, and therefore the optimization chooses different support variables and hyper-parameters. We call it MT-PP because in the single center case it corresponds to a multi-task version of PP [11].
5. **Convolved Multiple Output GP (MGP-FITC, MGP-PITC)**: the approaches proposed in [1]. For all experiments, we use code from [1] with the following setting. The kernel type is set to be `gg`. The hyperparameters parameters and the position of inducing variables are obtained via optimizing the marginal likelihood using a scaled conjugated gradient algorithm. The support variables are initialized as equally spaced points over the range of the inputs. We set the $R_q = 1$, which means that the latent functions share the same covariance function. Whenever possible, we set Q which, roughly

speaking, corresponds to the number of centers in our approach, to agree with the number of centers. The number of maximum iterations allowed in the optimization procedure is set to be 200. The number of support variables is controlled in the experiments as in our methods.

Three datasets are used to demonstrate the empirical performance of the proposed approach. The first synthetic dataset contains data sampled according to our model. The second dataset is also synthetic but it is generated from differential equations describing glucose concentration in biological experiments, a problem that has been previously used to evaluate multi-task GP [7]. Finally, we apply the proposed method on a real astrophysics dataset. For all experiments, the kernels for different centers are assumed to be the same. The hyperparameter for the Dirichlet distribution is set to be $\alpha_0 = 1/K$. The inducing variables are initialized to be equally spaced points over the range of the inputs. To initialize, tasks are randomly assigned into groups. We run the conjugate gradient algorithm (`minimize.m`) on a small subset of tasks (100 tasks each having 5 samples) to get the starting values of hyperparameters of the $\widetilde{\mathcal{K}}$ and $\mathcal{K}$, and then follow with the full optimization as above. Finally, we repeat the entire procedure 5 times and choose the one that achieves best variational lower bound. The maximum number of iterations for the stochastic coordinate descent is set to be 50 and the maximum number of iterations for the variational inference is set to be 30. The entire experiment is repeated 10 times to obtain the average performance and error bars.

6.1 Synthetic data

In the first experiment, we demonstrate the performance of our algorithm on a regression task with artificial data. More precisely, we generated 1000 single-center tasks where each $f^j(x) = \bar{f}(x) + \tilde{f}^j(x)$ is generated on the interval $x \in [-10, 10]$. Each task has 5 samples. The fixed-effect function is sampled from a GP with covariance function $\mathbf{Cov}[\bar{f}(t_1), \bar{f}(t_2)] = e^{-(t_1-t_2)^2/2}$. The individual effect $\tilde{f}^j$ is sampled via a GP with the covariance function $\mathbf{Cov}[\tilde{f}^j(t_1), \tilde{f}^j(t_2)] = 0.25e^{-(t_1-t_2)^2/2}$. The noise level σ^2 is set to be 0.1. The sample points $\boldsymbol{x}^j$ for each task are sampled uniformly in the interval $[-10, 10]$ and the 100 test samples are chosen equally spaced in the same interval. The fixed-effect curve is generated by drawing a single realization from the distribution of $\bar{\mathbf{f}}$ while the $\{\mathbf{f}^j\}$ are sampled i.i.d. from their common prior. We set the number of latent functions $Q = 1$ for MGP. The results are shown in Fig. (1). The top row shows qualitative results for one run using 20 support variables. We restrict the initial support variables to be in $[-7, 7]$ on purpose to show that the proposed method is capable of finding the optimal inducing variables. It is clear that the predictive distribution of the proposed method is much closer to the results of direct inference. The bottom row gives quantitative results for SMSE and MSLL showing the same, as well as showing that with 40 pseudo inputs the proposed method recovers the performance of full inference. The MGP performs poorly on this dataset, indicating that it is not sufficient to capture the random effect. We also see a large

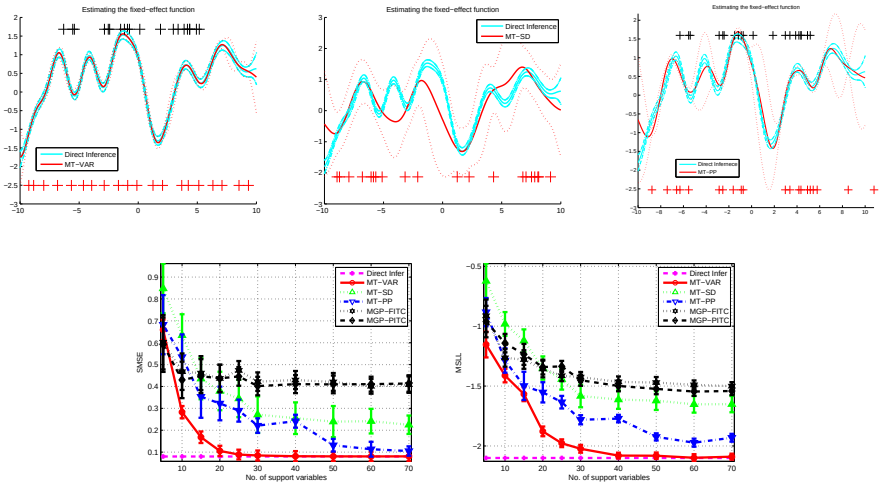

Fig. 1. Synthetic Data. Top row: Predictive distribution for the fixed-effect. The solid line denotes the predictive mean and the corresponding dotted line is the predictive variance. The black crosses (at the top) are the initial value of the support variables and the red ones (at the bottom) are their values after learning process. Bottom row: The average SMSE and MSLL for all the tasks.

computational advantage over MGP in this experiment. When the number of inducing variables is 20, the training time for FITC (the time for constructing the sparse model plus the time for optimization) is 1515.19 sec. while the proposed approach is about 7 times faster (201.81 sec.).

6.2 Simulated Glucose Data

We evaluate our method to reconstruct the glucose profiles in an intravenous glucose tolerance test (IVGTT) [16, 7] where [7] developed an online multi-task GP solution for the case where sample points are frequently shared among tasks. This provides a more realistic test of our algorithm because data is not generated explicitly by our model. We follow previous work and generate the data using minimal models of glucose which is commonly used to analyze glucose and insulin IVGTT data [16], as follows

$$\begin{aligned} \dot{G}(t) &= -[S_G + X(t)]G(t) + S_G \cdot G_b + \delta(t) \cdot D/V \\ \dot{X}(t) &= -p_2 \cdot X(t) + p_2 \cdot S_I \cdot [I(t) - I_b] \\ G(0) &= G_b, \quad X(0) = 0 \end{aligned} \tag{23}$$

where D denotes the glucose dose, $G(t)$ is plasma glucose concentration and $I(t)$ is the plasma insulin concentration which is assumed to be known. G_b and I_b are the glucose and insulin base values. $X(t)$ is the insulin action and $\delta(t)$ is the Dirac delta function. S_G, S_I, p_2, V are four parameters of this model.

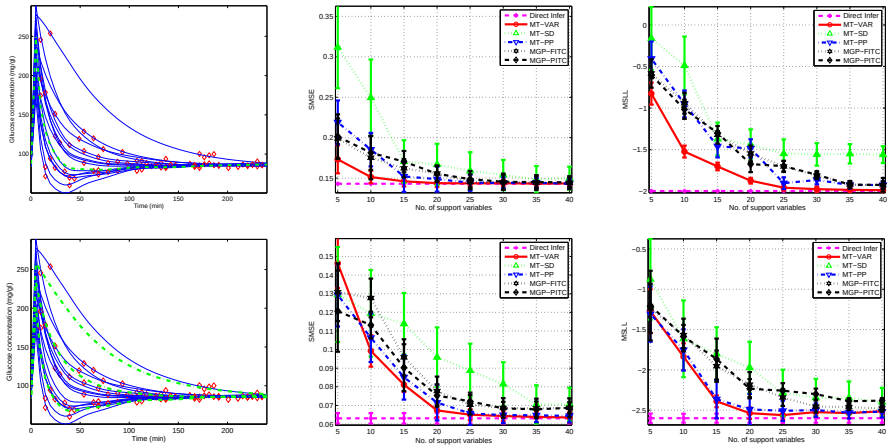

Fig. 2. Simulated Glucose Data. Left: 15 tasks (Blue) with observations (Red Diamonds) and estimated fixed-effect curve (Green) obtained from 1000 IVGTT responses. Center: The average SMSE for all tasks; Right: The average MSLL for all tasks.

We generate 1000 synthetic subjects (tasks) following the setup in previous work: 1) the four parameters are sampled from a multivariate Gaussian with the results from the normal group in Table 1 of [16], 2) $I(t)$ is obtained via spline interpolation using the real data in [16]; 3) G_b is fixed to be 84 and D is set to be 300; 4) $\delta(t)$ is simulated using a Gaussian profile with support on the positive axis and the standard deviation (SD) randomly drawn from a uniform distribution on the interval $[0, 1]$; 5) Noise is added to the observations with $\sigma^2 = 1$. Each task has 5 measurements chosen uniformly from the interval $[1, 240]$ and an additional 10 measurements are used for testing. Notice that the approach in [7] cannot deal with this situation efficiently since the inputs do not share samples often.

The experiments were done under both the single center and the multi center setting. The plots of task distribution on the left of Fig. 2 suggest that one can get more accurate estimation by using multiple centers. For the multiple center case, the number of centers for the proposed method is (arbitrarily) set to be 3 ($K = 3$) and the number of latent function of MGP is set to be 2 ($Q = 2$) (We were not able of obtain reasonable results using MGP when $Q = 3$). The experimental results (top/bottom for single/multi center) are shown in Fig. 2. First, we observe that the multi-center version performs better than the single center one, indicating that the group-based generalization of the traditional mixed-effect model is beneficial. Second, we can see that all the methods achieve reasonably good performance, but that the proposed method significantly outperforms the other methods.

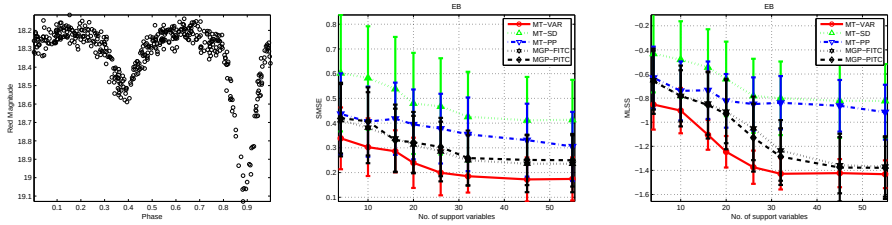

Fig. 3. OGLEII: Left: time series for EB star. Middle and Right show SMSE and MSLL respectively for EB type.

6.3 Real Astrophysics Data

We evaluate our method using the astronomy dataset of [17], where a generative model was developed to capture and classify different types of stars. The dataset, extracted from the OGLEII survey [14], includes stars of 3 types (RRL, CEPH, EB) which constitute 3 datasets in our context. One example of EB is shown in Fig. 3. This star is densely sampled but some stars have less samples and we simulate the sparse case by sub-sampling in our experiments. In [17], we developed a grouped mixed-effect multi-task model that in addition allowed for phase shift of the light measurements. As shown in [17], stars of the same type have a spread of different shapes and the group structure is useful in modeling this domain. However, for inference, [17] used a simple approach clipping sample points to a fine grid of 200 equally spaced points, due to the high dimensionality of the full sample (over 18000 points).

Here we use a random subset of 700 stars (tasks) for each type and preprocess the data normalizing each star to have mean 0 and *SD* 1, and using universal phasing [8] to phase each time series to align the maximum of a sliding window of 5% of the original points. For each time series, we randomly sample 10 examples for training and 10 examples for testing per evaluation of SMSE and MSLL. The number of centers is set to be 3 for the proposed approach and for MGP we set $Q = 1$ (We were not able to use $Q > 1$). The results for EBs are shown in Fig. (3). We can see that the proposed model outperforms all other methods. For Cepheid and RRL (results not shown due to space limit), the performance of the proposed model and MGP is very close and they outperform the other methods.

7 Conclusion

The paper develops an efficient variational learning algorithm for the grouped mixed-effect GP for multi-task learning, which compresses the information of all tasks into an optimal set of support variables for each mean effect. Experimental evaluation demonstrates the effectiveness of the proposed method. In future, it will be interesting to derive an online sparse learning algorithm for this model.

Acknowledgement

We would like to thank the authors of [1] who kindly made their code available online. This research was partly supported by NSF grant IIS-0803409. The experiments in this paper were performed on the the Tufts Linux Research Cluster supported by Tufts UIT Research Computing.

References

1. M. Álvarez and N. Lawrence. Computationally efficient convolved multiple output Gaussian processes. *JMLR*, 12:1425–1466, 2011.
2. M. Álvarez, D. Luengo, M. Titsias, and N. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In *AISTATS*, 2010.
3. M. Álvarez, L. Rosasco, and N. Lawrence. Kernels for vector-valued functions: a review. *Arxiv preprint arXiv:1106.6251*, 2011.
4. E. Bonilla, K. Chai, and C. Williams. Multi-task Gaussian process prediction. *NIPS*, 20:153–160, 2008.
5. Z. Lu, T. Leen, Y. Huang, and D. Erdogmus. A reproducing kernel Hilbert space framework for pairwise time series distances. In *ICML*, pages 624–631, 2008.
6. G. Pillonetto, G. De Nicolao, M. Chierici, and C. Cobelli. Fast algorithms for nonparametric population modeling of large data sets. *Automatica*, 45(1):173–179, 2009.
7. G. Pillonetto, F. Dinuzzo, and G. De Nicolao. Bayesian Online Multitask Learning of Gaussian Processes. *IEEE T-PAMI*, 32(2):193–205, 2010.
8. P. Protopapas, J. M. Giammarco, L. Faccioli, M. F. Struble, R. Dave, and C. Alcock. Finding outlier light curves in catalogues of periodic variable stars. *Monthly Notices of the Royal Astronomical Society*, 369:677–696, 2006.
9. J. Quiñonero-Candela and C. Rasmussen. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.
10. C. Rasmussen and H. Nickisch. Gaussian Processes for Machine Learning (GPML) Toolbox. *JMLR*, 11:3011–3015, 2010.
11. C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.
12. C. Seeger, M. Williams and N. Lawrence. Fast forward selection to speed up sparse gaussian process regression. In *AISTATS 9*. 2003.
13. G. Z. Snelson, M. Sparse Gaussian processes using pseudo-inputs. In *NIPS 18*, pages 1257–1264. 2006.
14. I. Soszynski, A. Udalski, and M. Szymanski. The Optical Gravitational Lensing Experiment. Catalog of RR Lyr Stars in the Large Magellanic Cloud 06. *Acta Astronomica*, 53:93–116, 2003.
15. M. Titsias. Variational learning of inducing variables in sparse gaussian processes. *AISTATS*, 2009.
16. P. Vicini and C. Cobelli. The iterative two-stage population approach to ivgtt minimal modeling: improved precision with reduced sampling. *American Journal of Physiology-Endocrinology and Metabolism*, 280(1):E179, 2001.
17. Y. Wang, R. Khardon, and P. Protopapas. Shift-invariant grouped multi-task learning for Gaussian processes. *ECML*, pages 418–434, 2010.
18. K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. In *ICML*, pages 1012–1019, 2005.