Shift-Invariant Grouped Multi-task Learning for Gaussian Processes

Yuyang Wang[1], Roni Khardon[1], and Pavlos Protopapas[2]

[1] Tufts University, Medford, MA USA
{ywang02,roni}@cs.tufts.edu
[2] Harvard-Smithsonian Center for Astrophysics, Cambridge, MA USA
pprotopapas@cfa.harvard.edu

Abstract. Multi-task learning leverages shared information among data sets to improve the learning performance of individual tasks. The paper applies this framework for data where each task is a phase-shifted periodic time series. In particular, we develop a novel Bayesian nonparametric model capturing a mixture of Gaussian processes where each task is a sum of a group-specific function and a component capturing individual variation, in addition to each task being phase shifted. We develop an efficient EM algorithm to learn the parameters of the model. As a special case we obtain the Gaussian mixture model and EM algorithm for phased-shifted periodic time series. Experiments in regression, classification and class discovery demonstrate the performance of the proposed model using both synthetic data and real-world time series data from astrophysics. Our methods are particularly useful when the time series are sparsely and non-synchronously sampled.

1 Introduction

In many real world problems we are interested in learning multiple related tasks where the training set for each task is quite small. For example, in pharmacological studies, we may be attempting to predict the concentration of some drug at different times across multiple patients. Finding a good regression function for an individual patient based only on his or her measurements can be difficult due to insufficient training data for the patient. Instead, by using measurements across all the patients, we may be able to leverage common patterns across patients to obtain better estimates for the population and for each patient individually. Multi-task learning captures this intuition aiming to learn multiple correlated tasks simultaneously. This idea has attracted much interest in the literature and several approaches have been applied to a wide range of domains, including medical diagnosis [1], recommendation systems [2] and HIV Therapy Screening [3]. Building on the theoretical framework for single-task learning, multi-task learning has recently been formulated as a multi-task regularization problem in vector-valued Reproducing Kernel Hilbert space (RKHS) [4].

Several approaches to multi-task learning exist in the context of Bayesian statistics. Considering hierarchical Bayesian models, one can view the parameter

J.L. Balcázar et al. (Eds.): ECML PKDD 2010, Part III, LNAI 6323, pp. 418–434, 2010.

sharing of the prior among tasks as a form of multi-task learning [5]. Recently, Bayesian models for multi-task learning were formalized using Gaussian processes [6,7,8]. In this nonparametric mixed-effect model, information is shared among tasks by having each task combine a common (fixed effect) portion and a task specific portion, each of which is generated by an independent Gaussian process. Our work builds on this formulation extending it and the associated algorithms in several ways.

We introduce a multi-task learning model with two novel aspects. First, we allow the fixed effect to be multi-modal so that each task may draw its fixed effect from a different cluster. Second, we extend the model so that each task may be an arbitrarily phase-shifted image of the original time series. This yields our model the *Shift-invariant Grouped Mixed-effect model.*

Our main technical contribution is the inference algorithm for the proposed model. We develop details for the EM algorithm optimizing the *Maximum A Posteriori* (MAP) estimates for the parameters of the model. Technically, the two main insights are in estimating the expectation for the coupled hidden variables (the cluster identities and the task specific portion of the time series) and in solving the regularized least squares problem for a set of phase-shifted observations. As a special case our algorithm yields the Gaussian mixture model (GMM) for phase shifted time series.

Seen from this perspective the paper provides a probabilistic extension of the Phased K-means algorithm [9] that performs clustering for phase-shifted time series data, and a nonparametric Bayesian extension of mixtures of random effects regressions [10] that was recently used for for curve clustering.

Our model primarily captures regression of time series but because it is a generative model it can be used for class discovery, clustering and classification. We demonstrate the utility of the model for such applications with both synthetic data and real-world time series data from astrophysics. The experiments show that our model can yield superior results when compared to single task learning and Gaussian mixture models, especially when each individual task is sparsely and non-synchronously sampled.

2 Shift-Invariant Grouped Mixed-Effect Model

2.1 Preliminaries

Throughout the paper, scalars are denoted using italics, as in $x, y \in \mathbb{R}$; vectors use bold typeface, as in $\mathbf{x}, \mathbf{y}$, and x_i denotes the ith entry of $\mathbf{x}$. For a vector $\mathbf{x}$ and real valued function $f : \mathbb{R} \to \mathbb{R}$, we extend the notation for f to vectors so that $f(\mathbf{x}) = [f(x_1), \cdots, f(x_n)]^{\mathrm{T}}$ where superscript the T stands for transposition (and the result is a column vector). $\mathbb{I}$ is the identity matrix.

A Gaussian process (GP) is a functional extension for Multivariate Gaussian distributions. In the Bayesian literature, it has been widely used in statistical models by substituting a parametric latent function with a stochastic process with Gaussian prior [11]. More precisely, under the single-task setting a simple

Gaussian regression model is given by $y = f(\mathbf{x}) + \epsilon_i$, where f's prior is a zero-mean GP with covariance function $\mathcal{K}$ and ϵ_i is independent zero mean white noise with variance σ^2. Given data set $\mathcal{D} = \{\mathbf{x}_i, y_i\}, i = 1, \cdots, N$, let $\mathbf{K} = (\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j))_{i,j}$, then $\mathbf{f} \triangleq [f(\mathbf{x}_1), \cdots, f(\mathbf{x}_N)]^{\mathrm{T}} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$ where $\mathcal{N}$ denotes the normal distribution and the posterior on $\mathbf{f}$ is given by

$$\Pr(\mathbf{f}|\mathcal{D}) = \mathcal{N}(\mathbf{K}(\sigma^2\mathbb{I} + \mathbf{K})^{-1}\mathbf{y}, \sigma^2(\sigma^2\mathbb{I} + \mathbf{K})^{-1}\mathbf{K}).$$

The predictive distribution for some test point $\mathbf{x}_*$ is

$$\begin{aligned}\Pr(f(\mathbf{x}_*)|\mathbf{x}_*, \mathcal{D}) &= \int \Pr(f(\mathbf{x}_*)|\mathbf{x}_*, f) \Pr(f|\mathcal{D}) df \\ &= \mathcal{N}\left(\mathbf{k}(\mathbf{x}_*)^{\mathrm{T}}(\sigma^2\mathbb{I} + \mathbf{K})^{-1}\mathbf{y}, \mathcal{K}(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}(\mathbf{x}_*)^{\mathrm{T}}(\sigma^2\mathbb{I} + \mathbf{K})^{-1}\mathbf{k}(\mathbf{x}_*)\right)\end{aligned}$$

where $\mathbf{k}(\mathbf{x}_*) = [\mathcal{K}(\mathbf{x}_1, \mathbf{x}_*), \cdots, \mathcal{K}(\mathbf{x}_N, \mathbf{x}_*)]^{\mathrm{T}}$. Furthermore, a Gaussian process f corresponds to a RKHS $\mathcal{H}$ with kernel $\mathcal{K}$ such that $\mathrm{cov}[f(\mathbf{x}), f(\mathbf{y})] = \mathcal{K}(\mathbf{x}, \mathbf{y})$ for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. In this way, we can express a zero mean Gaussian process as a distribution on functions f with the following probability [12][1]

$$f \sim \exp\left\{-\frac{1}{2}\|f\|^2_{\mathcal{H}}\right\}. \tag{1}$$

2.2 The GMT Model

Recently, GPs have been used for multi-task learning [6,8,12]. Given data $\{\mathcal{D}^j\}$, the Bayesian nonparametric mixed-effect model captures each task f^j with respect to $\mathcal{D}^j$ using a sum of an average effect function and an individual variation for each task, $f^j(x) = \bar{f}(x) + \tilde{f}^j(x), \quad j = 1, \cdots, M$ where $\bar{f}$ and $\{\tilde{f}^j\}$ are zero mean Gaussian processes. This assumes that the fixed-effect (mean function) $\bar{f}$ is sufficient to capture the behavior of the data, an assumption that is problematic for distributions with several modes. The model is also not suitable for shifted time series. To address these deficiencies we model our data as follows. For each j and $x \in [0, T)$,

$$f^j(x) = [\bar{f}_{z_j} * \delta_{t_j}](x) + \tilde{f}^j(x), j = 1, \cdots, M \tag{2}$$

where $z_j \in \{1, \cdots, k\}$, $\{\bar{f}_s\}, s = 1, \cdots, k$ and $\tilde{f}^j$ are zero mean Gaussian processes. We use $*$ to denote circular convolution, and δ_{t_j} is the Dirac δ function with support at $t_j \in [0, T)$. Therefore, $[\bar{f}_{z_j} * \delta_{t_j}](x) = \bar{f}_{z_j}(x - t_j \mod T)$. The underlying GPs, i.e. $\{\bar{f}_s\}, \tilde{f}^j$, are assumed to be mutually independent.

We next define the generative model which we call *Shift-invariant Grouped Mixed-effect model* (GMT). In this model, k group effect functions are assumed to share the same Gaussian prior characterized by $\mathcal{K}_0$. The individual effect

[1] In general, a Gaussian process does not induce a probability distribution on the corresponding RKHS but such use has been previously advocated; for further details see [13].

functions are Gaussian processes with covariance function $\mathcal{K}$. The model is characterized by parameter set $\{\mathcal{K}_0, \mathcal{K}, \boldsymbol{\alpha}, \{t_j\}, \sigma^2\}$ and summarized as follows:

$$\begin{aligned}
\bar{f}_s|\mathcal{K}_0 &\sim \exp\left\{-\frac{1}{2}\|\bar{f}_s\|^2_{\mathcal{H}_0}\right\}, \quad s = 1, 2, \cdots, k \\
\tilde{f}^j|\mathcal{K} &\sim \exp\left\{-\frac{1}{2}\|\tilde{f}^j\|^2_{\mathcal{H}}\right\}, \\
z_j|\boldsymbol{\alpha} &\sim \text{Discrete}(\boldsymbol{\alpha}), \\
\mathbf{y}^j &\sim \mathcal{N}\left(f^j(\mathbf{x}^j), \sigma^2\mathbb{I}\right), \quad \text{where } f^j = \bar{f}_{z_j} * \delta_{t_j} + \tilde{f}^j, \quad j = 1, 2, \cdots, M
\end{aligned} \tag{3}$$

where $\boldsymbol{\alpha}$ specifies the mixture proportions. Let $\mathcal{X} = \{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^M\}$ and $\mathcal{Y} = \{\mathbf{y}^1, \mathbf{y}^2, \cdots, \mathbf{y}^M\}$, where $\mathbf{x}^j$ are the time points when each time series is sampled and $\mathbf{y}^j$ are the corresponding observations.

We assume that the group effect kernel $\mathcal{K}_0$ and the number of centers k are known. The assumption on $\mathcal{K}_0$ is reasonable, in that normally we can get more information on the shape of the mean waveforms, thereby making it possible to design kernel for $\mathcal{H}_0$ but the individual variations are more arbitrary. The assumption that k is known requires model selection. An extension using a non-parametric model like the *Dirichlet process* that does not limit k is possible but we leave this to future work. The group effect $\{\bar{f}_s\}$, individual shifts $\{t_j\}$, noise variance σ^2 and the kernel for individual variations $\mathcal{K}$ are unknown and need to be estimated.

The model above is a standard model for regression. We propose to use it for classification by learning a mixture model for each class and using the MAP probability for the class for classification. In particular, consider a training set that has L classes, where the jth instance is given by $\mathcal{D}^j = (\mathbf{x}^j, \mathbf{y}^j, o^j) \in \mathbb{R}^{n_j} \times \mathbb{R}^{n_j} \times \{1, 2, \cdots, L\}$. Each observation $(\mathbf{x}^j, \mathbf{y}^j)$ is given a label from $\{1, 2, \cdots, L\}$. The problem is to learn the model M_ℓ for each class (L in total) separately and the classification rule for a new instance $(\mathbf{x}, \mathbf{y})$ is given by $o = \operatorname{argmax}[\Pr(\mathbf{y}|\mathbf{x}; \mathrm{M}_\ell)\Pr(\ell)]$. As we show in our experiments, the proposed generative model can provide explanatory power for the application while giving excellent classification performance.

2.3 Parameter Learning

Given data set $\mathcal{D} = \{\mathbf{x}^j, \mathbf{y}^j\} = \{x_i^j, y_i^j\}$, $i = 1, \cdots, n_j$, $j = 1, \cdots, M$, the learning process aims to find the MAP estimates of the parameter set $\mathcal{M}^* = \{\boldsymbol{\alpha}, \{\bar{f}_s\}, \{t_j\}, \sigma^2, \mathcal{K}\}$ such that

$$\mathcal{M}^* = \underset{\mathcal{M}}{\operatorname{argmax}} \left(\Pr(\mathcal{Y}|\mathcal{X}; \mathcal{M}) \times \Pr[\{\bar{f}_s\}; \mathcal{K}_0]\right). \tag{4}$$

The direct optimization of Eq. (4) is analytically intractable because of coupled sums that come from the mixture distribution. To solve this problem, we resort to the EM algorithm. The EM algorithm is an iterative method for optimizing the maximum likelihood (ML) or MAP estimates in the context of hidden variables.

In our case, the hidden variables are $\mathbf{z} = \{z_j\}$ (which is the same as in standard GMM), and $\mathbf{f} = \{\mathbf{f}_j \triangleq \tilde{f}^j(\mathbf{x}^j)\}, j = 1, \cdots, M$. The algorithm iterates between the following expectation and maximization steps until it converges to a local maximum.

2.4 Expectation Step

In the **E**-step, we calculate

$$Q(\mathcal{M}, \mathcal{M}^g) = \mathbb{E}_{\{\mathbf{z},\mathbf{f}|\mathcal{X},\mathcal{Y};\mathcal{M}^g\}} \left[\log \left\{\Pr(\mathcal{Y}, \mathbf{f}, \mathbf{z}|\mathcal{X}; \mathcal{M}) \times \Pr[\{\bar{f}_s\}; \mathcal{K}_0]\right\}\right] \quad (5)$$

where $\mathcal{M}^g$ stands for estimated parameters from the last iteration. For our model, the difficulty comes from estimating the expectation with respect to the coupled latent variables $\{\mathbf{z}, \mathbf{f}\}$. We next show how this can be done. First notice that, $\Pr(\mathbf{z}, \mathbf{f}|\mathcal{X}, \mathcal{Y}; \mathcal{M}^g) = \prod_j \Pr(z_j, \mathbf{f}_j|\mathcal{X}, \mathcal{Y}; \mathcal{M}^g)$ and further that

$$\Pr(z_j, \mathbf{f}_j|\mathcal{X}, \mathcal{Y}; \mathcal{M}^g) = \Pr(z_j|\mathbf{x}^j, \mathbf{y}^j; \mathcal{M}^g) \times \Pr(\mathbf{f}_j|z_j, \mathbf{x}^j, \mathbf{y}^j; \mathcal{M}^g). \quad (6)$$

The first term in Eq. (6) can be further written as

$$\Pr(z_j|\mathbf{x}^j, \mathbf{y}^j; \mathcal{M}^g) \propto \Pr(z_j; \mathcal{M}^g) \Pr(\mathbf{y}^j|z_j, \mathbf{x}^j; \mathcal{M}^g) \quad (7)$$

where $\Pr(z_j; \mathcal{M}^g)$ is specified by the parameters estimated from last iteration. Since z_j is given, $\Pr(\mathbf{y}^j|z_j, \mathbf{x}^j; \mathcal{M}^g)$ is the marginal distribution that can be calculated using a GP regression model, that is, $\mathbf{y}^j|z_j \sim \mathcal{N}([\bar{f}_{z_j} * \delta_{t_j}](\mathbf{x}^j), \mathbf{K}_j^g + \sigma^2\mathbb{I})$. As a result $\Pr(z_j|\mathbf{x}^j, \mathbf{y}^j; \mathcal{M}^g)$ can be calculated explicitly. Next consider the second term in Eq. (6). Given z_j, we know that $f^j = \bar{f}_{z_j} + \tilde{f}^j$, i.e. there is no uncertainty about the identity of $\bar{f}_{z_j}$ and therefore the calculation amounts to estimating the posterior distribution under standard GP regression. The conditional distribution is given by $\mathbf{f}_j|z_j, \mathbf{x}^j, \mathbf{y}^j \sim \mathcal{N}(\mu_j^g, \mathbf{C}_j^g)$ where μ_j^g and $\mathbf{C}_j^g$ are the posterior mean and covariance matrix given by

$$\mu_j^g = \mathbf{K}_j^g(\mathbf{K}_j^g + \sigma^2\mathbb{I})^{-1}(\mathbf{y}^j - \bar{\mathbf{f}}^j), \quad \mathbf{C}_j^g = \mathbf{K}_j^g - \mathbf{K}_j^g(\mathbf{K}_j^g + \sigma^2\mathbb{I})^{-1}\mathbf{K}_j^g \quad (8)$$

where $\mathbf{K}_j^g$ is the kernel matrix for the jth task using parameters from the last iteration. To derive the concrete form of $Q(\mathcal{M}, \mathcal{M}^g)$, denote $z_{il} = 1$ iff $z_i = l$. Then the complete data likelihood can be reformulated as

$$\mathcal{L} = \Pr(\mathcal{Y}, \mathbf{f}, \mathbf{z}; \mathcal{X}, \mathcal{M}) = \prod_j \prod_s \left[\alpha_s \Pr(\mathbf{y}^j|\mathbf{f}_j, z_j = s; \mathcal{M}) \Pr(\mathbf{f}_j; \mathcal{M})\right]^{z_{js}} \quad (9)$$

where we have used the fact that exactly one z_{js} is 1 for each j and thus included the last term inside the product over s. Then Eq. (5) can be written as

$$Q(\mathcal{M}, \mathcal{M}^g) = -\frac{1}{2}\sum_s \|f_s\|_{\mathcal{H}_0}^2 + \mathbb{E}_{\{\mathbf{z},\mathbf{f}|\mathcal{X},\mathcal{Y};\mathcal{M}^g\}} [\log \mathcal{L}]. \quad (10)$$

Denote the second term by $\widetilde{Q}$. By a version of Fubini's theorem we have

$$\begin{aligned}\widetilde{Q} &= \mathbb{E}_{\{\mathbf{z}|\mathcal{X},\mathcal{Y};\mathcal{M}^g\}}\mathbb{E}_{\{\mathbf{f}|\mathbf{z},\mathcal{X},\mathcal{Y};\mathcal{M}^g\}}\left[\log\mathcal{L}\right] \\ &= \sum_{\mathbf{z}}\Pr(\mathbf{z}|\mathcal{X},\mathcal{Y};\mathcal{M}^g)\Bigg\{\sum_j\sum_s z_{js} \\ &\qquad \times\int d\Pr(\mathbf{f}_j|z_j=s)\log\left[\alpha_s\Pr(\mathbf{y}^j|\mathbf{f}_j,z_j=s;\mathcal{M})\Pr(\mathbf{f}_j;\mathcal{M})\right]\Bigg\}.\end{aligned} \tag{11}$$

Since the last term in Eq. (11) does not include any z_i, the equation can be further decomposed as

$$\begin{aligned}\widetilde{Q} &= \sum_j\sum_s\left(\sum_{\mathbf{z}}\Pr(\mathbf{z}|\mathcal{X},\mathcal{Y};\mathcal{M}^g)z_{js}\right) \\ &\qquad \times\left\{\int d\Pr(\mathbf{f}_j|z_j=s)\log[\alpha_s\Pr(\mathbf{y}^j|\mathbf{f}_j,z_j=s;\mathcal{M})\Pr(\mathbf{f}_j;\mathcal{M})]\right\} \\ &= \sum_j\sum_s\gamma_{js}\int d\Pr(\mathbf{f}_j|z_j=s)\log\left[\alpha_s\Pr(\mathbf{y}^j|\mathbf{f}_j,z_j=s;\mathcal{M})\Pr(\mathbf{f}_j;\mathcal{M})\right]\end{aligned} \tag{12}$$

where $\gamma_{js}=\mathbb{E}[z_{js}|\mathbf{y}^j,\mathbf{x}^j;\mathcal{M}^g]$ can be calculated from Eq. (7) and it can be viewed as a fractional label for the jth task in the sth group.

Recall that $\Pr(\mathbf{y}^j|\mathbf{f}_j,z_j=s)$ is a normal distribution given by $\mathcal{N}(\bar{f}_{z_j}*\delta_{t_j}(\mathbf{x}^j)+\mathbf{f}_j,\sigma^2\mathbb{I})$ and $\Pr(\mathbf{f}_j;\mathcal{M})$ is a standard multivariate Gaussian distribution determined by its prior, that is

$$\Pr(\mathbf{f}_j;\mathcal{M})=\frac{1}{\sqrt{(2\pi)^{n_j}|\mathbf{K}_j|}}\exp\left\{-\frac{1}{2}\mathbf{f}_j^T\mathbf{K}_j^{-1}\mathbf{f}_j\right\}.$$

Using these facts and after some algebraic manipulation, $Q(\mathcal{M},\mathcal{M}^g)$ can be rewritten as

$$\begin{aligned}Q(\mathcal{M},\mathcal{M}^g) &= -\frac{1}{2}\sum_s\|\bar{f}_s\|^2_{\mathcal{H}_0}-\sum_j n_j\log\sigma+\sum_j\sum_s\gamma_{js}\log\alpha_s \\ &\quad -\frac{1}{2\sigma^2}\sum_j\sum_s\gamma_{js}\mathbb{E}_{\{\mathbf{f}_j|z_j=s,\mathbf{x}^j,\mathbf{y}^j;\mathcal{M}^g\}}\left[\|\mathbf{y}^j-[\bar{f}_s*\delta_{t_j}](\mathbf{x}^j)-\mathbf{f}_j\|^2\right] \\ &\quad +\sum_j\sum_s\gamma_{js}\mathbb{E}_{\{\mathbf{f}_j|\mathbf{x}^j,\mathbf{y}^j;\mathcal{M}^g\}}\left[\log\Pr(\mathbf{f}_j;\mathcal{M})\right].\end{aligned} \tag{13}$$

2.5 Maximization Step

In this step, we aim to find

$$\mathcal{M}^*=\operatorname*{argmax}_{\mathcal{M}}Q(\mathcal{M},\mathcal{M}^g) \tag{14}$$

and use $\mathcal{M}^*$ to update the model parameters. As we show next, this can be decomposed into three separate optimization problems as follows

$$\mathcal{M}^* = \underset{\mathcal{M}}{\operatorname{argmax}}\, Q_1((\{\bar{f}_s\}, \{\delta_{t_j}\}, \sigma)) + Q_2(\mathcal{K}) + \left\{ \sum_j \sum_s \gamma_{js} \log \alpha_s \right\}.$$

That is, $\boldsymbol{\alpha}$ can be estimated easily using its separate term, Q_1 is only a function of $(\{f_s\}, \{t_j\}, \sigma)$ and Q_2 depends only on $\mathcal{K}$. We have

$$\begin{aligned} Q_1(\{\bar{f}_s\}, \{t_j\}, \sigma^2) = \frac{1}{2} \sum_s \|\bar{f}_s\|^2_{\mathcal{H}_0} + \sum_j n_j \log \sigma + \frac{1}{2\sigma^2} \sum_j \sum_s \gamma_{js} \\ \times \mathbb{E}_{\{\mathbf{f}_j | z_j = s, \mathbf{x}^j, \mathbf{y}^j; \mathcal{M}^g\}} \left[\|\mathbf{y}^j - [f_s * \delta_{t_j}](\mathbf{x}^j) - \mathbf{f}_j\|^2 \right]. \end{aligned} \tag{15}$$

The remaining part, Q_2 is

$$\begin{aligned} Q_2(\mathcal{K}) &= \sum_j \sum_s \gamma_{js} \mathbb{E}_{\{\mathbf{f}_j | z_j = s, \mathbf{x}^j, \mathbf{y}^j; \mathcal{M}^g\}} \left[\log \Pr(\mathbf{f}_j; \mathcal{M}) \right]. \\ &= -\frac{1}{2} \sum_j \log |\mathbf{K}_j| - \frac{1}{2} \sum_j \sum_s \gamma_{js} \operatorname{Tr} \left(\mathbf{K}_j^{-1} (\mathbf{C}_j^g + \mu_j^g (\mu_j^g)^{\mathrm{T}}) \right). \end{aligned} \tag{16}$$

Learning $\{\bar{f}_s\}, \{t_j\}, \sigma^2$: To optimize Eq. (15), denote the residual $\tilde{\mathbf{y}}^j = \mathbf{y}^j - \mu_{js}$, where $\mu_{js} = \mathbb{E}[\mathbf{f}_j | \mathbf{y}^j, z_j = s]$. Given σ, optimizing $(\{\bar{f}_s\}, \{t_j\})$ decouples into k independent sub-problems, where finding the sth group effect $\bar{f}_s$ and its corresponding shift $\{t_j\}$ amounts to solving

$$\underset{f \in \mathcal{H}_0, t_1, \cdots, t_M \in [0,T)}{\operatorname{argmin}} \left\{ \frac{1}{2\sigma^2} \sum_j \gamma_{js} \sum_{i=1}^{n_j} (\tilde{\mathbf{y}}_i^j - [f * \delta_{t_j}](\mathbf{x}_i^j))^2 + \frac{1}{2} \|f\|^2_{\mathcal{H}_0} \right\}. \tag{17}$$

Note that different $\mathbf{x}^j, \mathbf{y}^j$ have different dimensions n_j and they are not assumed to be sampled at regular intervals. For further development, following [8], it is useful to introduce the closure vector $\breve{\mathbf{x}} \in \mathbb{R}^{\mathbb{N}}$ whose components are the distinct elements of $\mathcal{X}$. For example if $\mathbf{x}^1 = [1, 2, 3]^T, \mathbf{x}^2 = [2, 3, 4, 5]^T$, then $\breve{\mathbf{x}} = [1, 2, 3, 4, 5]^T$. For the jth task, let the binary matrix C^k be such that $\mathbf{x}^j = C^j \cdot \breve{\mathbf{x}}$ and $f(\mathbf{x}^j) = C^j \cdot f(\breve{\mathbf{x}})$. That is, C^j extracts the values corresponding to the jth task from the closure vector.

If $\{t_j\}$ are fixed, then the optimization in Eq. (17) is standard and the representer theorem gives the form of the solution as $f(\cdot) = \sum_{i=1}^{\mathbb{N}} c_i \mathcal{K}_0(\breve{x}_i, \cdot)$. Denoting the kernel matrix as $\mathfrak{K} = \mathcal{K}_0(\breve{x}_i, \breve{x}_j), i, j = 1, \cdots, \mathbb{N}$, and $\mathbf{c} = [c_1, \cdots, c_{\mathbb{N}}]^T$ we get $f(\breve{\mathbf{x}}) = \mathfrak{K}\mathbf{c}$. To simplify the optimization we assume that $\{t_j\}$ can only take values in the discrete space $\{\tilde{t}_1, \cdots, \tilde{t}_L\}$, that is, $t_j = \tilde{t}_i$, for some $i \in 1, 2, \cdots, L$ (e.g., a fixed finite fine grid), where we always choose $\tilde{t}_1 = 0$. Therefore, we can

write $\left[f * \delta_{t_j}\right](\check{\mathbf{x}}) = \widetilde{\mathcal{K}}_{t_j}^{\mathrm{T}}\mathbf{c}$, where $\widetilde{\mathcal{K}}_{t_j}$ is $\mathcal{K}_0(\check{\mathbf{x}}, [(\check{\mathbf{x}} - \tilde{t}_j) \mod T])$. Accordingly, Eq. (17) is reduced to

$$\underset{\mathbf{c}\in\mathbb{R}^{\mathbb{N}}, t_1, \cdots, t_j \in \{\tilde{t}_i\}}{\operatorname{argmin}} \left\{ \sum_j \gamma_{js} \|\tilde{\mathbf{y}}^j - C^j \cdot \widetilde{\mathcal{K}}_{t_j}^{\mathrm{T}}\mathbf{c}\|^2 + \frac{1}{2}\mathbf{c}^{\mathrm{T}}\mathfrak{K}\mathbf{c} \right\}. \tag{18}$$

To solve this optimization, we follow a cyclic optimization approach where we alternate between steps of optimizing f and $\{t_j\}$ respectively.

At step ℓ, we optimize equation (18) with respect to $\{t_j\}$ given $\mathbf{c}^{(\ell)}$. Since $\mathbf{c}^{(\ell)}$ is known, it follows immediately that Eq. (18) decomposes to M independent problems, where for the jth we need to find $t_j^{(\ell)}$ such that $C^j \widetilde{\mathcal{K}}_{t_j^{(\ell)}}^{\mathrm{T}}\mathbf{c}$ is closest to $\tilde{\mathbf{y}}^j$ under Euclidean distance. A brute force search with time complexity $\mathbf{\Theta}(\mathbb{N}L)$ yields the optimal solution. If the time series are synchronously sampled (i.e. $C^j = \mathbb{I}, j = 1, \cdots, M$) and the allowed time shifts are at sample points, this reduces to finding the shift τ corresponding the *cross-correlation*. In this case, one can use the convolution theorem to find the same value in $\mathbf{\Theta}(\mathbb{N}\log\mathbb{N})$ time [14].

At step $\ell+1$, we optimize equation (18) with respect to $\mathbf{c}^{(\ell+1)}$ given $t_1^{(\ell)}, \cdots, t_M^{(\ell)}$. For the jth task, since $t_j^{(\ell)}$ is known, denote $C^j \widetilde{\mathcal{K}}_{t_j^{(\ell)}}^{\mathrm{T}}$ as $\mathfrak{M}_j^{(\ell)}$. Therefore, $\mathbf{c}^{(\ell+1)}$ can be calculated by solving the following regularized least square problem

$$\underset{\mathbf{c}\in\mathbb{R}^{\mathbb{N}}}{\operatorname{argmin}} \left\{ \sum_j \gamma_{js} \|\tilde{\mathbf{y}}^j - \mathfrak{M}_j^{(\ell)}\mathbf{c}\|^2 + \frac{1}{2}\mathbf{c}^T\mathfrak{K}\mathbf{c} \right\}. \tag{19}$$

Obviously, each step decreases the value of the objective function and therefore the algorithm will converge.

Given $\{\bar{f}_s\}, \{t_j\}$, the value of σ^2 is optimized by $(\sigma^*)^2 = R/\sum_j n_j$, where $R = \sum_j \mathrm{Tr}(\mathbf{C}_j^g) + \sum_j \sum_s \gamma_{js} \left(\|\mathbf{y}^j - [\bar{f}_s^* * \delta_{t_j}](\mathbf{x}^j) - \boldsymbol{\mu}_{js}\|^2\right)$. We use alternating optimization steps to optimize over the three parameters together.

Learning the kernel for individual effects: The work of [12] has already shown how to optimize the kernel in a similar context. Here we provide some of the details for completeness. If the kernel function $\mathcal{K}$ admits a parametric form with parameter θ, for example the RBF kernel

$$\mathcal{K}(x, y) = a \exp\left\{ -\frac{\|x - y\|^2}{2s^2} \right\} \tag{20}$$

then the optimization of $\mathcal{K}$ amounts to finding θ^* such that

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \left\{ -\frac{1}{2} \sum_j \sum_s \gamma_{js} \mathrm{Tr}\left((\mathbf{K}_j; \theta)^{-1} (\mathbf{C}_j^g + \boldsymbol{\mu}_{js}^g (\boldsymbol{\mu}_{js}^g)^{\mathrm{T}})\right) \right\}.$$

The parametric form of the kernel is a prerequisite to perform the regression task when examples are not sampled synchronously as in our development above. If

the data is synchronously sampled, for classification tasks we only need to find the kernel matrix $\mathbf{K}$ for the given sample points and the optimization problem can be rewritten as

$$\mathbf{K}^* = \underset{\mathbf{K}}{\operatorname{argmax}} \left\{ -\frac{1}{2}\sum_j \log|\mathbf{K}| - \frac{1}{2}\sum_j \sum_s \gamma_{js} \operatorname{Tr}\left(\mathbf{K}^{-1}(\mathbf{C}_j^g + \boldsymbol{\mu}_{js}^g(\boldsymbol{\mu}_{js}^g)^{\mathrm{T}})\right) \right\}. \tag{21}$$

Similar to maximum likelihood estimation for the multivariate Gaussian distribution, the solution is $\mathbf{K}^* = \frac{1}{M}\sum_j \sum_s \gamma_{js}(\mathbf{C}_j^g + \boldsymbol{\mu}_{js}^g(\boldsymbol{\mu}_{js}^g)^{\mathrm{T}})$. In our experiments, we use both approaches. For the parametric form we use Eq. (20).

3 Experiments

Our implementation of the algorithm makes use of the gpml package[2] [11] and extends it to implement the required functions. The EM algorithm is restarted 5 times and the function that best fits the data is chosen. The EM algorithm stops when the difference of the log-likelihood is less than 10e-5 or at a maximum of 200 iterations.

3.1 Regression on Synthetic Data

In the first experiment, we demonstrate the performance of our algorithm on a regression task with artificial data. We generated data that is not phase shifted under a mixture of two Gaussian processes. More precisely, each $\bar{f}_s(x), s = 1, 2$ is generated on interval $[-50, 50]$ from a Gaussian process with covariance function $\operatorname{cov}[\bar{f}_s(t_1), \bar{f}_s(t_2)] = e^{-(t_1-t_2)^2/25}, \quad s = 1, 2$. The individual effect $\tilde{f}_j$ is sampled via a Gaussian process with the covariance function $\operatorname{cov}[\tilde{f}_j(t_1), \tilde{f}_j(t_2)] = 0.2e^{-(t_1-t_2)^2/16}$. Then the hidden label z_j is sampled from a discrete distribution with $\alpha = [0.5, 0.5]$. The vector $\breve{\mathbf{x}}$ consists of 100 samples on $[-50, 50]$. We fix a sample size N and each $\mathbf{x}^j$ includes N randomly chosen points from $\{\breve{x}_1, \cdots, \breve{x}_{100}\}$. The observation $f^j(\mathbf{x}^j)$ is obtained as $(f_{z_j} + \tilde{f}_j)(\mathbf{x}^j)$. In the experiment, we vary the individual sample length N from 5 to 50. Finally, we generated 50 random tasks with the observation $\mathbf{y}^j$ for task j given by $\mathbf{y}^j \sim \mathcal{N}(f^j(\mathbf{x}^j), 0.01 \times \mathbb{I}), \quad j = 1, \cdots, 50$. The methods compared here include:

1. **Single-task learning procedure (ST)**, where each $\bar{f}^j$ is estimated only using $\{\mathbf{x}_i^j, \mathbf{y}_i^j\}, i = 1, 2, \cdots, N$.
2. **Single center mixed-effect multi-task learning (SCMT)**, amounts to the mixed-effect model [8] where one average function $\bar{f}$ is learned from $\{\mathbf{x}^j, \mathbf{y}^j\}, j = 1, \cdots, 50$ and $f^j = \bar{f} + \tilde{f}^j, j = 1, \cdots, 50$.
3. **Grouped mixed-effect model (GMT)**, the proposed method.

[2] Available at `http://www.gaussianprocess.org/gpml/`

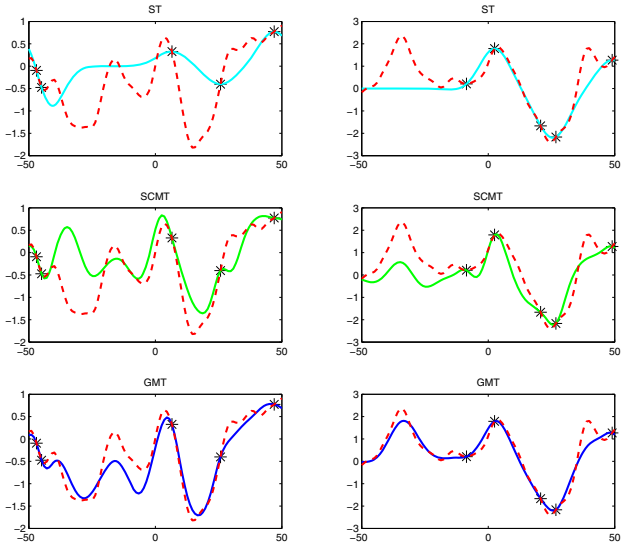

Fig. 1. Simulated data: Comparison of the estimated function between single, multi-task and grouped multi-task. The red dotted line is the reference true function.

4. **"Cheating" grouped fixed-effect model (CGMT)**, which follows the same algorithm as the GMT but uses the true label z_j instead of their expectation for each task j. This serves as an upper bound for the performance of the proposed algorithm.

Except for ST, the other three algorithms use the same method to learn the kernel of the individual effects, which is assumed to be RBF. The Root Mean Square Error (RMSE) for the four approaches is reported. For task j, the RMSE is defined as $\text{RMSE}_j = \sqrt{\|f(\check{\mathbf{x}}) - f^j(\check{\mathbf{x}})\|^2/100}$, where f is the learned function and RMSE for the data set is the mean of $\{\text{RMSE}_j\}, j = 1, \cdots, 50$.

To illustrate the results qualitatively, we first plot in Fig. (1) the true and learned functions in one trial. The left column illustrates some task that is sampled from group effect $\bar{f}_1$ and the right column is for $\bar{f}_2$. It it easy to see that, as expected, the tasks are poorly estimated under ST due the sparse sampling. The SCMT performs better than ST but its estimate is poor in areas where two centers disagree. The estimates of GMT are much closer to the true function.

The left plot of Fig. (2) shows a comparison of the algorithms for 50 random data sets under the above setting when N equals 5. We see that most of the time GMT performs as well as its upper bound, illustrating that it recovers the correct membership of each task. On a few data sets, our algorithm is trapped in a local maximum yielding performance similar to SCMT and ST. The right part of Fig. (2) shows the RMSE for increasing values of N. From the plot, we can draw the conclusion that the proposed method works much better than SCMT and ST when the number of samples is less than 30. As the number of samples

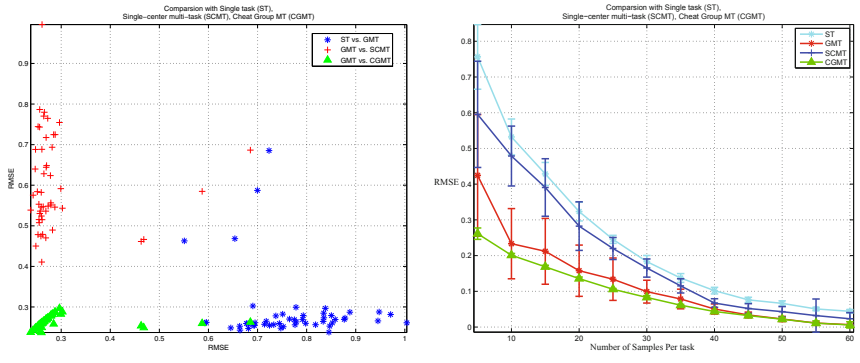

Fig. 2. Comparison between single, multi-task, and grouped multi-task learning on simulated data. Left: the figure gives 3 pairwise comparisons when sample size is 5. We can see that the GMT is better than ST since the blue stars are concentrated on the lower right. Similarly, the plot of red pluses demonstrates the advantage of GMT over SCMT and the plot of green triangles shows that the algorithm behaves almost as well as its upper bound. Right: the figure shows RMSE as a function of sample size.

for each task increases, all methods are improving, but the proposed method always outperforms SCMT and ST in our experiments. Finally, all algorithms converge to almost the same performance level where observations in each task are sufficient to recover the underlying function.

3.2 Classification on Astrophysics Data

The concrete application motivating this research is the classification of stars into several meaningful categories from the astronomy literature. Classification is an important step within astrophysics research, as evidenced by published catalogs such as OGLE [15] and MACHO [16,17]. However, the number of stars in such surveys is increasing dramatically. For example Pan-STARRS [18] will collect data on the order of hundreds of billions of stars. Therefore, it is desirable to apply state-of-art machine learning techniques to enable automatic processing for astrophysics data classification. The data from star surveys is normally represented by time series of brightness measurements, based on which they are classified into categories. Stars whose behavior is periodic are especially of interest in such studies. Fig. (3) shows several examples of such time series generated from the three major types of periodic variable stars: Cepheid, RR Lyrae, and Eclipsing Binary. In our experiments only stars of the types in Fig. (3) are present in the data, and the period of each star is given.

From Fig. (3), it can be noticed that there are two main characteristics of this domain. The time series are not phase aligned, meaning that the light curves in the same category share a similar shape but with some unknown shift. The time series are non-synchronously sampled and each light curve has a different number of samples and sampling times. We run our experiment on the OGLEII data set [19]. This data set consists of 14087 time series (light curves) with

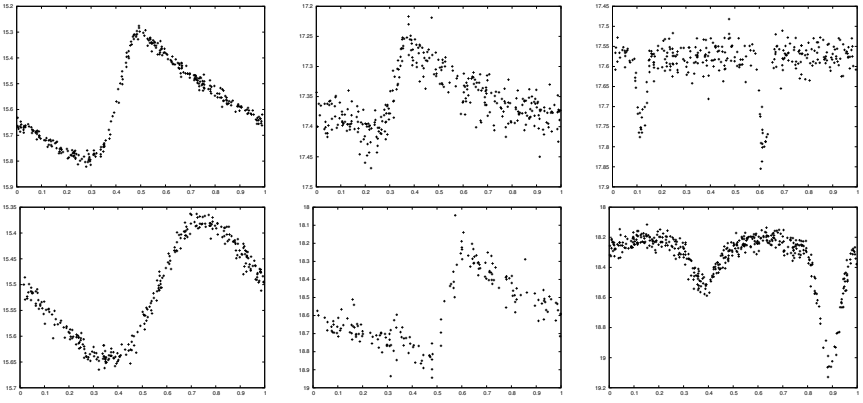

Fig. 3. Examples of light curves of periodic variable stars. Each column shows two stars of the same type. Left: Cepheid, middle: RR Lyrae, right: Eclipsing Binary.

(3425,3390,7272) in the categories (CEPH, RRL, EB) respectively. We perform several experiments with this data set to explore the potential of the proposed method. Previous work using this data [20] developed a kernel for periodic time series and used it with SVM to obtained good classification performance. We use the results of [20] as our baseline.[3]

Classification using dense-sampled time series: In the first experiment, the time series are smoothed using a simple average filter, re-sampled to 50 points via linear-interpolation and normalized to have mean 0 and standard deviation of 1. Therefore, the time series are synchronously sampled in the pre-processing stage. We compare our method to GMM and 1-Nearest Neighbor (1-NN). These two approaches are performed on the time series processed by Universal phasing (UP), which uses the method from [14] to phase each time series according to the sliding window on the time series with the maximum mean. We use a sliding window size of 5% of the number of original points; the phasing takes place after the pre-processing explained above. We learn a model for each class separately and for each class the model order for the GMM and the GMT is set to 15.

We run 10-fold cross-validation (CV) over the entire data set and the results are shown in Tab. (1). We see that when the data is densely and synchronously sampled, the proposed method performs similar to the GMM, and they both outperform the kernel based results from [20]. The poor performance of SCMT shows that a single center is not sufficient for this data. The similarity of the GMM and the proposed method under these experimental conditions is not surprising. The reason is that when the time series are synchronously sampled,

[3] [20] used additional features, in addition to time series itself, to improve the classification performance. Here we focus on results using the time series only. Extensions to add such features to our model are orthogonal to the theme of the paper and we therefore leave them to future work in the context of the application.

Table 1. Accuracies with standard deviations reported on OGLEII dataset

	UP + GMM	SCMT	GMT	UP + 1-NN	K + SVM[20]
RESULTS	0.956 ± 0.006	0.874 ± 0.008	0.952 ± 0.005	0.865 ± 0.006	0.947 ± 0.005

aside from the difference of phasing, finding the group effect functions is reduced to estimating the mean vectors of the GMM. In addition, learning the kernel in the nonparametric approach is the same as estimating the covariance matrix of the GMM. More precisely, assuming all time series are phased and sampled at the same time points, the following results hold:

1. By placing a flat prior on the group effect function $\bar{f}_s, s = 1, \cdots, k$, or equivalently setting $\|\bar{f}_s\|^2_{\mathcal{H}_0} = 0$, Eq. (17) is reduced to finding a vector $\mu_s \in \mathbb{N}$ that minimizes $\sum_j \gamma_{js} \|\tilde{\mathbf{y}}^j - \mu_s\|^2$. Therefore, we obtain $\bar{f}_s = \mu_s = \sum_j \gamma_{js} \tilde{\mathbf{y}}_j / \sum_j \gamma_{js}$, which is exactly the mean of the sth cluster during the iteration of EM algorithm under the GMM setting.

2. The kernel $\mathbf{K}$ is learned in a nonparametric way. Instead of estimating $\mathbf{K}$ and σ^2, it is convenient to put the two terms together, forming $\widehat{\mathbf{K}} = \mathbf{K} + \sigma^2 \mathbb{I}$, that is $\mathbf{y}^j$ is a deterministic function of $f^j(\mathbf{x}^j)$ which in turn has an extra σ^2 term on the diagonal of its covariance matrix. One can show that in this case the estimate of the kernel is $\widehat{\mathbf{K}} = \frac{1}{M} \sum_j \sum_s \gamma_{js} (\mathbf{y}^j - \mu_s)(\mathbf{y}^j - \mu_s)^{\mathrm{T}}$. In the standard EM algorithm for GMM, this is equal to the estimated covariance matrix when all k clusters are assumed to have the same covariance.

Accordingly, when time series are synchronously sampled, the proposed model can be viewed as an extension of the Phased K-means [9] and a variant of [21]. In experiments below, we extend the resulting Phased EM by allowing each cluster to have a separate covariance matrix.

Classification using sparse-sampled time series: The OGLEII data set is in some sense a "nice" subset of the data from its corresponding star survey. Stars with small number of samples are often removed in pre-processing steps (cf., [22]). The proposed method potentially provides a way to include these instances in the classification process. In the second experiment, we demonstrate the performance of the proposed method on times series with sparse samples. Similar to the synthetic data, we started from sub-sampled versions of the original time series to simulate the condition that we would encounter in further star surveys.[4] As in the previous experiment, each time series is universally phased, normalized and linearly-interpolated to length 50 to be plugged into GMM and 1-NN as well as the generalized Phased EM as discussed above. The RBF kernel is used for the proposed method and we used the same model order as above. Moreover, the performance for PKmeans is also presented, where the classification step is as follows: we learn the PKmeans model with $k = 15$ for each class

[4] For the proposed method, we clip the samples to a fine grid of 200 equally spaced time points on $[0, 1]$, which is also the set of allowed time shifts. This avoids having a very high dimensional $\breve{\mathbf{x}}$, e.g. over 18000 for OGLEII, which is not feasible for any kernel based regression method that relies on solving linear systems.

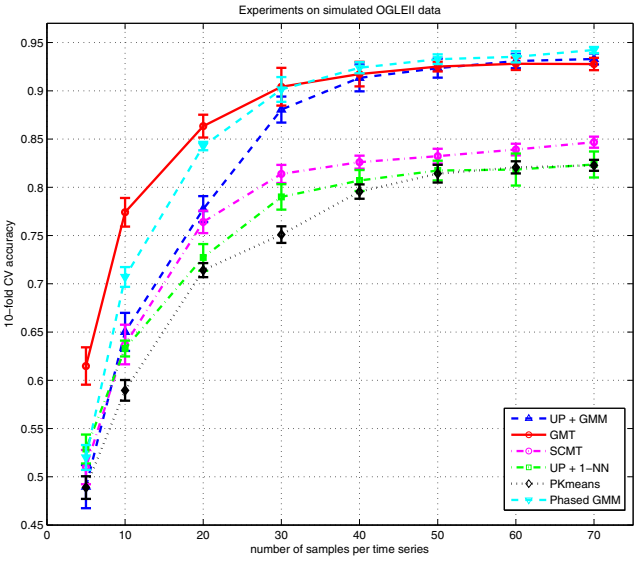

Fig. 4. OGLEII data: Comparison of algorithms with sparsely sampled data

and then the label of a new example is assigned to be the same as its closest centroid's label. PKmeans is also restarted 5 times and the best clustering is used for classification.

The results are shown in Fig. (4). When each time series has sparse samples (the number of samples per task is less than 30), the proposed method has a significant advantage over the other methods. As the number of samples per task increases, the proposed method improves fast and performs close to its optimal performance given by previous experiment. When the number of samples increases, the performance of the Phased EM gradually catches up and becomes better than the proposed method when each task has more than 50 samples. GMM plus universal phasing (UP) also achieves better performance when time series are densely sampled. One reason for the performance difference is the parametric form of the kernel in the GMT in this experiment (which is less flexible). The difference can also be attributed to the sharing of the covariance function in our model where the GMM and the Phased EM do not apply this constraint. Notice that the Phased EM algorithm always outperforms the GMM plus UP demonstrating that re-phasing the time series inside the EM algorithm improves the results. We also notice that for all 3 methods, the performance with dense data is lower than the results in Tab. (1). This is caused because the data set obtained by the interpolation of the sub-sampled measurements contains less information than that interpolated from the original measurements. Finally, note that PKmeans performs much worse than Phased EM showing that the probabilistic model adds a significant advantage. To summarize, we conclude from this experiment that the proposed method should be used when data is sparsely and non-synchronously sampled.

3.3 Class Discovery on Astrophysics Data

We show the potential of our model for class discovery by running GMT on the joint data set of the three classes (not using the labels) with a model order of 45. Then, each cluster is labeled according to the majority class of the instances that belong to the center. For a new test point, we determine which cluster it belongs to via the MAP probability and its label is given by the cluster that it is assigned to. We run 10 trials with different random initialization. The accuracy and standard deviation obtained are $[0.895, 0.010]$. Given the size of the data set and the relatively small number of clusters this is a significant indication of the potential for class discovery in astrophysics.

4 Conclusion and Future Work

We developed a novel Bayesian nonparametric multi-task learning model where each task is modeled as a sum of a group-specific function and an individual task function with a Gaussian process prior. We gave an efficient EM algorithm to learn the parameters of the model and demonstrated its effectiveness using experiments in regression, classification and class discovery.

The literature includes a significant amount of work on time series classification [23] (space constraints preclude a lengthy discussion). These include approaches based on feature extraction followed by feature based learning model [24], approaches based on similarities for time series [20] and hidden Markov models [25,26]. Despite the similar name, our model is different from mixture of experts where different GPs are in control of sub-regions of the input space [27]. Our work is most closely related to the so-called *mixture of regressions* [10,28,29]. As discussed above our model can be seen as a nonparametric Bayesian extension of the model in [10] and includes [21] as a special case with shared covariance matrix when we use the diagonal kernel function on the time grid.

For application in the astronomy context it is important to consider all steps of processing and classification so as to provide an end to end system. Therefore, two important issues to be addressed in future work include incorporating the period estimation phase into the method and developing an appropriate method for abstention in the classification step. Further, to get a full generalization of the phased GMM, it will be interesting to generalize our model to allow individual variations to come from cluster dependent RKHS. It would also be interesting to develop a corresponding discriminative model extending [5] to the GP context.

Acknowledgments

This research was partly supported by NSF grant IIS-0803409. The experiments in this paper were performed on the Odyssey cluster supported by the FAS Research Computing Group at Harvard and the Tufts Linux Research Cluster supported by Tufts UIT Research Computing.

References

1. Bi, J., Xiong, T., Yu, S., Dundar, M., Rao, R.: An improved multi-task learning approach with applications in medical diagnosis. In: ECML/PKDD, pp. 117–132 (2008)
2. Dinuzzo, F., Pillonetto, G., De Nicolao, G.: Client-server multi-task learning from distributed datasets. Arxiv preprint arXiv:0812.4235 (2008)
3. Bickel, S., Bogojeska, J., Lengauer, T., Scheffer, T.: Multi-task learning for HIV therapy screening. In: ICML, pp. 56–63 (2008)
4. Evgeniou, T., Micchelli, C., Pontil, M.: Learning multiple tasks with kernel methods. JMLR 6(1), 615–637 (2006)
5. Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multi-task learning for classification with Dirichlet process priors. JMLR 8, 35–63 (2007)
6. Yu, K., Tresp, V., Schwaighofer, A.: Learning Gaussian processes from multiple tasks. In: ICML, pp. 1012–1019 (2005)
7. Schwaighofer, A., Tresp, V., Yu, K.: Learning Gaussian process kernels via hierarchical Bayes. NIPS 17, 1209–1216 (2005)
8. Pillonetto, G., Dinuzzo, F., De Nicolao, G.: Bayesian Online Multitask Learning of Gaussian Processes. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(2), 193–205 (2010)
9. Rebbapragada, U., Protopapas, P., Brodley, C.E., Alcock, C.: Finding anomalous periodic time series. Machine Learning 74(3), 281–313 (2009)
10. Gaffney, S.J., Smyth, P.: Curve clustering with random effects regression mixtures. In: Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics (2003)
11. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. The MIT Press, Cambridge (2005)
12. Lu, Z., Leen, T., Huang, Y., Erdogmus, D.: A reproducing kernel Hilbert space framework for pairwise time series distances. In: ICML, pp. 624–631 (2008)
13. Seeger, M.: Gaussian processes for machine learning. International Journal of Neural Systems 14(2), 69–106 (2004)
14. Protopapas, P., Giammarco, J.M., Faccioli, L., Struble, M.F., Dave, R., Alcock, C.: Finding outlier light curves in catalogues of periodic variable stars. Monthly Notices of the Royal Astronomical Society 369, 677–696 (2006)
15. Udalski, A., Szymanski, M., Kubiak, M., Pietrzynski, G., Wozniak, P., Zebrun, Z.: Optical gravitational lensing experiment. photometry of the macho-smc-1 microlensing candidate. Acta Astronomica 47, 431–436 (1997)
16. Alcock, C., et al.: The MACHO Project - a Search for the Dark Matter in the Milky-Way. In: Soifer, B.T. (ed.) Sky Surveys. Protostars to Protogalaxies. Astronomical Society of the Pacific Conference Series, vol. 43, pp. 291–296 (1993)
17. Faccioli, L., Alcock, C., Cook, K., Prochter, G.E., Protopapas, P., Syphers, D.: Eclipsing Binary Stars in the Large and Small Magellanic Clouds from the MACHO Project: The Sample. Astronomy Journal 134, 1963–1993 (2007)
18. Hodapp, K.W., et al.: Design of the Pan-STARRS telescopes. Astronomische Nachrichten 325, 636–642 (2004)
19. Soszynski, I., Udalski, A., Szymanski, M.: The Optical Gravitational Lensing Experiment. Catalog of RR Lyr Stars in the Large Magellanic Cloud 06. Acta Astronomica 53, 93–116 (2003)
20. Wachman, G., Khardon, R., Protopapas, P., Alcock, C.: Kernels for Periodic Time Series Arising in Astronomy. In: ECML/PKDD, pp. 489–505 (2009)

21. Chudova, D., Gaffney, S.J., Mjolsness, E., Smyth, P.: Translation-invariant mixture models for curve clustering. In: SIGKDD, pp. 79–88 (2003)
22. Wachman, G.: Kernel Methods and Their Application to Structured Data. PhD thesis, Tufts University (2009)
23. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. VLDB 1(2), 1542–1552 (2008)
24. Osowski, S., Hoai, L., Markiewicz, T.: Support vector machine-based expert system for reliable heartbeat recognition. IEEE Transactions on Biomedical Engineering 51(4), 582–589 (2004)
25. Hughes, N., Tarassenko, L., Roberts, S.: Markov models for automated ECG interval analysis. NIPS 16 (2003)
26. Kim, S., Smyth, P.: Segmental Hidden Markov Models with Random Effects for Waveform Modeling. JMLR 7, 945–969 (2006)
27. Rasmussen, C.E., Ghahramani, Z.: Infinite mixtures of Gaussian process experts. NIPS 15, 881–888 (2002)
28. Gaffney, S.J., Smyth, P.: Trajectory clustering with mixtures of regression models. In: SIGKDD, pp. 63–72 (1999)
29. Gaffney, S.J., Smyth, P.: Joint probabilistic curve clustering and alignment. NIPS 17, 473–480 (2005)