# INFINITE MIXTURES OF MULTIVARIATE GAUSSIAN PROCESSES

**SHILIANG SUN**

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

**Abstract:**

This paper presents a new model called infinite mixtures of multivariate Gaussian processes, which can be used to learn vector-valued functions and applied to multitask learning. As an extension of the single multivariate Gaussian process, the mixture model has the advantages of modeling multimodal data and alleviating the computationally cubic complexity of the multivariate Gaussian process. A Dirichlet process prior is adopted to allow the (possibly infinite) number of mixture components to be automatically inferred from training data, and Markov chain Monte Carlo sampling techniques are used for parameter and latent variable inference. Preliminary experimental results on multivariate regression show the feasibility of the proposed model.

**Keywords:**

Gaussian process; Dirichlet process; Markov chain Monte Carlo; Multitask learning; Vector-valued function; Regression

## 1. Introduction

Gaussian processes provide a principled probabilistic approach to pattern recognition and machine learning. Formally, a Gaussian process is a collection of random variables such that any finite number of them obey a joint Gaussian prior distribution. As a Bayesian nonparametric model, the Gaussian process model proves to be very powerful for general function learning problems such as regression and classification [1, 2].

Recently, motivated by the need to learn vector-valued functions and for multitask learning, research on multivariate or multi-output Gaussian processes has attracted a lot of attention. By learning multiple related tasks jointly, the common knowledge underlying different tasks can be shared, and thus a performance gain is likely to be obtained [3]. Representative works on multivariate Gaussian processes include the methods given in [4, 5, 6].

However, it is well known that Gaussian processes suffer from two important limitations [2, 7]. First, limited by the inherent unimodality of Gaussian distributions, Gaussian processes cannot characterize multimodal data which are prevalent in practice. Second, they are computationally infeasible for big data, since inference requires the inversion of an $N \times N$ and $NM \times NM$ covariance matrix respectively for a single-variate and multivariate Gaussian process, where $N$ is the number of training examples and $M$ is the output dimensionality.

These two limitations can be greatly alleviated by making use of mixtures of Gaussian processes [8] where there are multiple Gaussian processes to jointly explain data and one example only belongs to one Gaussian process component. For mixtures of Gaussian processes, the infinite mixtures based on Dirichlet processes [9] are prevailing because they permit the number of components to be inferred directly from data and thus bypass the difficult model selection problem on the component number.

For single-variate or single-output Gaussian processes, there were already some variants and implementations for infinite mixtures which brought great success for data modeling and prediction applications [2, 7, 10]. However, no extension of multivariate Gaussian processes to mixture models has been presented yet. Here, we will fill this gap by proposing an infinite mixture model of multivariate Gaussian processes. It should be noted that the implementation of this infinite model is very challenging because the multivariate Gaussian processes are much more complicated than the single-variate Gaussian processes.

The rest of this paper is organized as follows. After providing the new infinite mixture model in Section 2, we show how the hidden variable inference and prediction problems are performed in Section 3 and Section 4, respectively. Then, we report experimental results on multivariate regression in Section 5. Finally, concluding remarks and future work directions are given in Section 6.

## 2. The proposed model

The graphical model for the proposed infinite mixture of multivariate Gaussian processes (IMMGP) on the observed training data $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ is depicted in Figure 1.
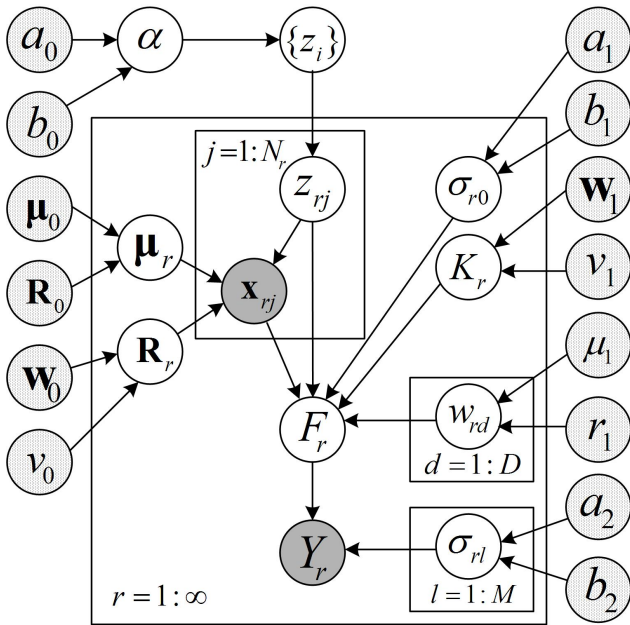


**Figure 1. The graphical model for IMMGP**

In the graphical model, $r$ indexes the $r$th Gaussian process component in the mixture, which can be infinitely large if enough data are provided. $N_r$ is the number of examples belonging to the $r$th component. $D$ and $M$ are the dimensions for the input and output space, respectively. The set $\{\alpha, \{\boldsymbol{\mu}_r\}, \{\mathbf{R}_r\}, \{\sigma_{r0}\}, \{K_r\}, \{w_{rd}\}, \{\sigma_{r\ell}\}\}$ includes all random parameters, which is denoted here by $\Theta$. The latent variables are $z_i$ ($i = 1, \ldots, N$) and $F_r$ ($r = 1 : \infty$), where $F_r$ can be removed from the graphical model by integration if we directly consider a distribution over $\{Y_r\}$. Denote the set of latent indicators by $Z$, that is, $Z = \{z_i\}_{i=1}^N$. Since $F_r$ is for illustrative purposes only, the latent indicators $Z$ and random parameters $\Theta$ constitute the total hidden variables. The circles in the left and right columns of the graphical model indicate the hyperparameters, whose values are usually found by maximum likelihood estimation or designated manually if people have a strong belief on them.

The observation likelihood for our IMMGP is

$$
\begin{aligned}
&p(\{\mathbf{x}_i, \mathbf{y}_i\}|\Theta) \\
&= \sum_Z p(Z|\Theta) \prod_r p(\{\mathbf{y}_i : z_i = r\}|\{\mathbf{x}_i : z_i = r\}, \Theta) \times \\
&\quad p(\{\mathbf{x}_i : z_i = r\}|\Theta) \\
&= \sum_Z p(Z|\Theta) \prod_r p(\{\mathbf{y}_i : z_i = r\}|\{\mathbf{x}_i : z_i = r\}, \Theta) \times \\
&\quad \prod_{j=1}^{N_r} p(\mathbf{x}_{rj}|\boldsymbol{\mu}_r, \mathbf{R}_r).
\end{aligned}
\tag{1}
$$

### 2.1. Distributions for hidden variables

$\alpha$ is the concentration parameter of the Dirichlet process, which controls the prior probability of assigning an example to a new mixture component and thus influences the total number of components in the mixture model. A gamma distribution $\mathcal{G}(\alpha|a_0, b_0)$ is used. We use the parameterization for the gamma distribution given in [11]. Given $\alpha$ and $\{z_i\}_{i=1}^n$, the distribution of $z_{n+1}$ is easy to get with the Chinese restaurant process metaphor [9].

The distribution over the input space for a mixture component is given by a Gaussian distribution with a full covariance

$$
p(\mathbf{x}|z = r, \boldsymbol{\mu}_r, \mathbf{R}_r) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_r, \mathbf{R}_r^{-1}),
\tag{2}
$$

where $\mathbf{R}_r$ is the precision (inverse covariance) matrix. This input model is often flexible enough to provide a good performance, though people can consider to adopt mixtures of Gaussian distributions to model the input space. Parameters $\boldsymbol{\mu}_r$ and $\mathbf{R}_r$ are further specified by a Gaussian distribution prior and a Wishart distribution prior, respectively

$$
\boldsymbol{\mu}_r \sim \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{R}_0^{-1}), \ \ \mathbf{R}_r \sim \mathcal{W}(\mathbf{W}_0, \nu_0).
\tag{3}
$$

The parameterization for the Wishart distribution is the same as that in [11].

A Gaussian process prior is placed over the latent functions $\{f_{r\ell}\}_{\ell=1}^M$ for component $r$ in our model. Assuming the Gaussian processes have zero mean values, we set

$$
\begin{aligned}
\mathbb{E}\big(f_{r\ell}(\mathbf{x})f_{rk}(\mathbf{x}')\big) &= \sigma_{r0} K_r(\ell, k) k_r(\mathbf{x}, \mathbf{x}'), \\
y_{r\ell}(\mathbf{x}) &\sim \mathcal{N}(f_{r\ell}(\mathbf{x}), \sigma_{r\ell}),
\end{aligned}
\tag{4}
$$

where scaling parameter $\sigma_{r0} > 0$, $K_r$ is a positive semi-definite matrix that specifies the inter-task similarities, $k_r(\cdot, \cdot)$ is a covariance function over inputs, and $\sigma_{r\ell}$ is the noise variance for the $\ell$th output of the $r$th component. The prior of the $M \times M$

positive semi-definite matrix $K_r$ is given by a Wishart distribution $\mathcal{W}(\mathbf{W}_1, \nu_1)$. $\sigma_{r0}$ and $\sigma_{r\ell}$ are given gamma priors $\mathcal{G}(\sigma_{r0}|a_1, b_1)$ and $\mathcal{G}(\sigma_{r\ell}|a_2, b_2)$, respectively. We set

$$k_r(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}\sum_{d=1}^{D} w_{rd}^2(\mathbf{x}_d - \mathbf{x}'_d)^2\right), \quad (5)$$

where $w_{rd}$ obeys a log-normal distribution $\mathcal{N}(\ln w_{rd}|\mu_1, r_1)$ with mean $\mu_1$ and variance $r_1$. The whole setup for a single Gaussian process component is in large difference with that in [4].

## 3. Inference

Since exact inference on the distribution $p(Z, \Theta|\mathcal{D})$ is infeasible, in this paper we use Markov chain Monte Carlo sampling techniques to obtain $L$ samples $\{Z_j, \Theta_j\}_{j=1}^{L}$ to approximate the distribution $p(Z, \Theta|\mathcal{D})$.

In particular, Gibbs sampling is adopted to represent the posterior of the hidden variables. First of all, we initialize all the variables in $\{Z, \Theta\}$ by sampling them from their priors. Then the variables are updated using the following steps.

(1) Update indicator variables $\{z_i\}_{i=1}^{N}$ one by one, by cycling through the training data.

(2) Update input and output space Gaussian process parameters $\{\{\boldsymbol{\mu}_r\}, \{\mathbf{R}_r\}, \{\sigma_{r0}\}, \{K_r\}, \{w_{rd}\}, \{\sigma_{r\ell}\}\}$ for each Gaussian process component in turn.

(3) Update Dirichlet process concentration parameter $\alpha$.

These three steps constitute a Gibbs sampling sweep over all hidden variables, which are repeated until the Markov chain has adequate samples. Note that samples in the burn-in stage should be removed from the Markov chain and are not used for approximating the posterior distribution.

In the following subsections, we provide the specific sampling method and formulations involved for each update.

### 3.1. Updating indicator variables

Let $Z_{-i} = Z\backslash z_i = \{z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_N\}$ and $\mathcal{D}_{-i} = \mathcal{D}\backslash\{\mathbf{x}_i, \mathbf{y}_i\}$. To sample $z_i$, we need the following posterior conditional distribution

$$p(z_i|Z_{-i}, \Theta, \mathcal{D})$$
$$\propto p(z_i|Z_{-i}, \Theta)p(\mathcal{D}|z_i, Z_{-i}, \Theta)$$
$$\propto p(z_i|Z_{-i}, \Theta)p\left(\mathbf{y}_i|\{\mathbf{y}_j : j \neq i, z_j = z_i\}, \{\mathbf{x}_j : z_j = z_i\}, \Theta\right)$$
$$\times p(\mathbf{x}_i|\boldsymbol{\mu}_{z_i}, \mathbf{R}_{z_i}), \quad (6)$$

where we have used a clear decomposition between the joint distributions of $\{\mathbf{x}_i, \mathbf{y}_i\}$ and $\mathcal{D}_{-i}$.

It is not difficult to calculate the three terms involved in the last two lines of (6). However, the computation of $p(\mathbf{y}_i|\{\mathbf{y}_j : j \neq i, z_j = z_i\}, \{\mathbf{x}_j : z_j = z_i\}, \Theta)$ may be more efficient if some approximation scheme or acceleration method is adopted. In addition, for exploring new experts, we just sample the parameters once from the prior, use them for the new expert, and then calculate (6), following [10, 12]. The indicator variable update method is also algorithm 8 from [13] with the auxiliary component parameter $m = 1$.

### 3.2. Updating input space component parameters

For the input space parameters $\boldsymbol{\mu}_r$ and $\mathbf{R}_r$, they can be sampled directly because their posterior conditional distributions have a simple formulation as a result of using conjugate priors.

$$p(\boldsymbol{\mu}_r|Z, \Theta\backslash\boldsymbol{\mu}_r, \mathcal{D}) = p(\boldsymbol{\mu}_r|\{\mathbf{x}_{rj}\}_{j=1}^{N_r}, \mathbf{R}_r)$$
$$\propto p(\boldsymbol{\mu}_r)p(\{\mathbf{x}_{rj}\}_{j=1}^{N_r}|\boldsymbol{\mu}_r, \mathbf{R}_r)$$
$$\propto |\mathbf{R}_0|^{1/2}\exp\{-\frac{1}{2}(\boldsymbol{\mu}_r - \boldsymbol{\mu}_0)^{\top}\mathbf{R}_0(\boldsymbol{\mu}_r - \boldsymbol{\mu}_0)\} \times$$
$$\prod_j |\mathbf{R}_r|^{1/2}\exp\{-\frac{1}{2}(\mathbf{x}_{rj} - \boldsymbol{\mu}_r)^{\top}\mathbf{R}_r(\mathbf{x}_{rj} - \boldsymbol{\mu}_r)\}$$
$$\propto \exp\{-\frac{1}{2}[\boldsymbol{\mu}_r^{\top}\mathbf{R}_0\boldsymbol{\mu}_r - 2\boldsymbol{\mu}_r^{\top}\mathbf{R}_0\boldsymbol{\mu}_0 +$$
$$\sum_j(\boldsymbol{\mu}_r^{\top}\mathbf{R}_r\boldsymbol{\mu}_r - 2\boldsymbol{\mu}_r^{\top}\mathbf{R}_r\mathbf{x}_{rj})]\}, \quad (7)$$

and therefore

$$p(\boldsymbol{\mu}_r|Z, \Theta\backslash\boldsymbol{\mu}_r, \mathcal{D})$$
$$= \mathcal{N}((\mathbf{R}_0 + N_r\mathbf{R}_r)^{-1}(\mathbf{R}_0\boldsymbol{\mu}_0 + \mathbf{R}_r\sum_j \mathbf{x}_j), (\mathbf{R}_0 + N_r\mathbf{R}_r)^{-1}).$$

$$p(\mathbf{R}_r|Z, \Theta\backslash\mathbf{R}_r, \mathcal{D}) = p(\mathbf{R}_r|\{\mathbf{x}_{rj}\}_{j=1}^{N_r}, \boldsymbol{\mu}_r)$$
$$\propto p(\mathbf{R}_r)p(\{\mathbf{x}_{rj}\}_{j=1}^{N_r}|\boldsymbol{\mu}_r, \mathbf{R}_r)$$
$$\propto |\mathbf{R}_r|^{(\nu_0-D-1)/2}\exp\{-\frac{1}{2}\text{Tr}(\mathbf{W}_0^{-1}\mathbf{R}_r)\} \times$$
$$\prod_j |\mathbf{R}_r|^{1/2}\exp\{-\frac{1}{2}(\mathbf{x}_{rj} - \boldsymbol{\mu}_r)^{\top}\mathbf{R}_r(\mathbf{x}_{rj} - \boldsymbol{\mu}_r)\}$$
$$\propto |\mathbf{R}_r|^{(\nu_0+N_r-D-1)/2} \times$$
$$\exp\{-\frac{1}{2}\text{Tr}((\mathbf{W}_0^{-1} + \sum_j(\mathbf{x}_{rj} - \boldsymbol{\mu}_r)(\mathbf{x}_{rj} - \boldsymbol{\mu}_r)^{\top})\mathbf{R}_r)\},$$

and thus

$$p(\mathbf{R}_r | Z, \Theta \backslash \mathbf{R}_r, \mathcal{D})$$
$$= \mathcal{W}\left(\left(\mathbf{W}_0^{-1} + \sum_j (\mathbf{x}_{rj} - \boldsymbol{\mu}_r)(\mathbf{x}_{rj} - \boldsymbol{\mu}_r)^\top\right)^{-1}, \nu_0 + N_r\right).$$

## 3.3. Updating output space component parameters

Note that $Y_r = \{\mathbf{y}_i : 1 \leq i \leq N, z_i = r\}$ and $|Y_r| = N_r$. In this subsection, we denote its $N_r$ elements by $\{\mathbf{y}_r^j\}_{j=1}^{N_r}$ which correspond to $\{\mathbf{x}_r^j\}_{j=1}^{N_r}$.

Define the complete $M$ outputs in the $r$th GP as

$$\mathbf{y}_r = (y_{r1}^1, \ldots, y_{r1}^{N_r}, y_{r2}^1, \ldots, y_{r2}^{N_r}, \ldots, y_{rM}^1, \ldots, y_{rM}^{N_r})^\top, \quad (8)$$

where $y_{r\ell}^j$ is the observation for the $\ell$th output on the $j$th input. According to the Gaussian process assumption given in (4), the observation $\mathbf{y}_r$ follows a Gaussian distribution

$$\mathbf{y}_r \sim \mathcal{N}(\mathbf{0}, \Sigma), \quad \Sigma = \sigma_{r0} K_r \otimes K_r^x + D_r \otimes I, \quad (9)$$

where $\otimes$ denotes the Kronecker product, $K_r^x$ is the $N_r \times N_r$ covariance matrix between inputs with $K_r^x(i,j) = k_r(\mathbf{x}_r^i, \mathbf{x}_r^j)$, $D_r$ is an $M \times M$ diagonal matrix with $D_r(i,i) = \sigma_{ri}$, $I$ is an $N_r \times N_r$ identity matrix, and therefore the size of $\Sigma$ is $MN_r \times MN_r$. The predictive distribution for the interested variable $\mathbf{f}^*$ on a new input $\mathbf{x}^*$ which belongs to the $r$th component is

$$\mathcal{N}(K^* \Sigma^{-1} \mathbf{y}_r, \ K^{**} - K^* \Sigma^{-1} K^{*\top}), \quad (10)$$

where $K^*_{(M \times MN_r)} = \sigma_{r0} K_r \otimes \mathbf{k}_r^{x*}$, $K^{**} = \sigma_{r0} K_r$, and $\mathbf{k}_r^{x*}$ is a $1 \times N_r$ row vector with the $i$th element being $k_r(\mathbf{x}^*, \mathbf{x}_r^i)$. Hence, the expected output on $\mathbf{x}^*$ is $K^* \Sigma^{-1} \mathbf{y}_r$.

Note that the calculation of $\Sigma^{-1}$ is a source for approximation to speed up training. However, this problem is easier than the original single GP model since we already reduced the inversion from an $MN \times MN$ matrix to several $MN_r \times MN_r$ matrices.

We use hybrid Monte Carlo [14] to update $\sigma_{r0}$, and the basic Metropolis-Hastings algorithm to update $K_r$, $\{w_{rd}\}$, and $\{\sigma_{r\ell}\}$ with the corresponding proposal distributions being their priors. Below we give the posteriors of the output space parameters and when necessary provide some useful technical details.

We have

$$p(\sigma_{r0} | Z, \Theta \backslash \sigma_{r0}, \mathcal{D})$$
$$\propto p(\sigma_{r0}) p(\mathbf{y}_r | \{\mathbf{x}_r^j\}_{j=1}^{N_r}, \sigma_{r0}, K_r, \{w_{rd}\}, \{\sigma_{r\ell}\})$$
$$\propto \sigma_{r0}^{a_1 - 1} \exp(-b_1 \sigma_{r0}) \times \frac{1}{|\Sigma|^{1/2}} \exp(-\frac{1}{2} \mathbf{y}_r^\top \Sigma^{-1} \mathbf{y}_r)$$
$$= \exp\left\{ -\left[(1 - a_1) \ln \sigma_{r0} + b_1 \sigma_{r0} + \frac{1}{2} \ln |\Sigma| + \frac{1}{2} \mathbf{y}_r^\top \Sigma^{-1} \mathbf{y}_r\right]\right\}, \quad (11)$$

and thus the potential energy $E(\sigma_{r0}) = (1 - a_1) \ln \sigma_{r0} + b_1 \sigma_{r0} + \frac{1}{2} \ln |\Sigma| + \frac{1}{2} \mathbf{y}_r^\top \Sigma^{-1} \mathbf{y}_r$. The gradient $dE(\sigma_{r0})/d\sigma_{r0}$ is needed in order to use hybrid Monte Carlo, which is given by

$$\frac{dE(\sigma_{r0})}{d\sigma_{r0}}$$
$$= \frac{1 - a_1}{\sigma_{r0}} + b_1 + \frac{1}{2} \text{Tr}[(\Sigma^{-1} - \Sigma^{-1} \mathbf{y}_r \mathbf{y}_r^\top \Sigma^{-1})(K_r \otimes K_r^x)].$$

$$p(K_r | Z, \Theta \backslash K_r, \mathcal{D})$$
$$\propto p(K_r) p(\mathbf{y}_r | \{\mathbf{x}_r^j\}_{j=1}^{N_r}, \sigma_{r0}, K_r, \{w_{rd}\}, \{\sigma_{r\ell}\})$$
$$\propto |K_r|^{(\nu_1 - M - 1)/2} \exp\{-\frac{1}{2} \text{Tr}(\mathbf{W}_1^{-1} K_r)\} \times$$
$$\frac{1}{|\Sigma|^{1/2}} \exp(-\frac{1}{2} \mathbf{y}_r^\top \Sigma^{-1} \mathbf{y}_r)$$
$$= \exp\left\{ -\frac{1}{2}\left[(M + 1 - \nu_1) \ln |K_r| + \text{Tr}(\mathbf{W}_1^{-1} K_r) + \ln |\Sigma| + \mathbf{y}_r^\top \Sigma^{-1} \mathbf{y}_r\right]\right\}. \quad (12)$$

$$p(w_{rd} | Z, \Theta \backslash w_{rd}, \mathcal{D})$$
$$\propto p(w_{rd}) p(\mathbf{y}_r | \{\mathbf{x}_r^j\}_{j=1}^{N_r}, \sigma_{r0}, K_r, \{w_{rd}\}, \{\sigma_{r\ell}\})$$
$$\propto w_{rd}^{-1} \exp\{-\frac{(\ln w_{rd} - \mu_1)^2}{2r_1}\} \times \frac{1}{|\Sigma|^{1/2}} \exp(-\frac{1}{2} \mathbf{y}_r^\top \Sigma^{-1} \mathbf{y}_r)$$
$$= \exp\left\{ -\left[\ln w_{rd} + \frac{(\ln w_{rd} - \mu_1)^2}{2r_1} + \frac{1}{2} \ln |\Sigma| + \frac{1}{2} \mathbf{y}_r^\top \Sigma^{-1} \mathbf{y}_r\right]\right\}. \quad (13)$$

$$p(\sigma_{r\ell} | Z, \Theta \backslash \sigma_{r\ell}, \mathcal{D})$$
$$\propto p(\sigma_{r\ell}) p(\mathbf{y}_r | \{\mathbf{x}_r^j\}_{j=1}^{N_r}, \sigma_{r0}, K_r, \{w_{rd}\}, \{\sigma_{r\ell}\})$$
$$\propto \sigma_{r\ell}^{a_2 - 1} \exp(-b_2 \sigma_{r\ell}) \times \frac{1}{|\Sigma|^{1/2}} \exp(-\frac{1}{2} \mathbf{y}_r^\top \Sigma^{-1} \mathbf{y}_r)$$
$$= \exp\left\{ -\left[(1 - a_2) \ln \sigma_{r\ell} + b_2 \sigma_{r\ell} + \frac{1}{2} \ln |\Sigma| + \frac{1}{2} \mathbf{y}_r^\top \Sigma^{-1} \mathbf{y}_r\right]\right\}. \quad (14)$$

## 3.4. Updating the concentration parameter $\alpha$

The basic Metropolis-Hastings algorithm is used to update $\alpha$. Let $1 \leq c \leq N$ be the number of distinct values in

$\{z_1, \ldots, z_N\}$. It is clear from [15] that

$$p(c|\alpha, N) = \beta_c^N \frac{\alpha^c \, \Gamma(\alpha)}{\Gamma(N + \alpha)}, \tag{15}$$

where coefficient $\beta_c^N$ is the absolute value of Stirling numbers of the first kind, and $\Gamma(\cdot)$ is the gamma function. With (15) as the likelihood, we can get the posterior of $\alpha$ is

$$p(\alpha|c, N) \propto p(\alpha)p(c|\alpha, N) \propto p(\alpha)\frac{\alpha^c \, \Gamma(\alpha)}{\Gamma(N + \alpha)}. \tag{16}$$

Since the gamma prior is used, it follows that,

$$p(\alpha|c, N) \propto \frac{\alpha^{c+a_0-1} \exp(-b_0\alpha)\Gamma(\alpha)}{\Gamma(N + \alpha)}. \tag{17}$$

## 4. Prediction

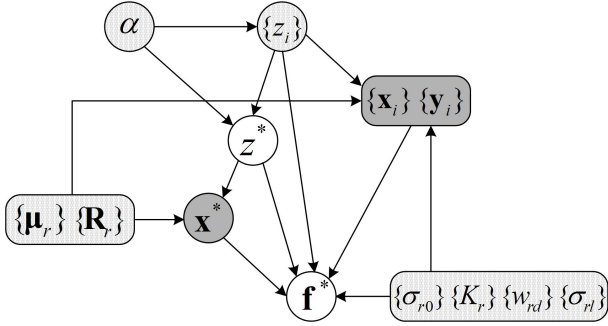The graphical model for prediction is shown in Figure 2.



**Figure 2. The graphical model for prediction on a new input $\mathbf{x}^*$**

The predictive distribution for the predicted output of a new test input $\mathbf{x}^*$ is

$$p(\mathbf{f}^*|\mathbf{x}^*, \mathcal{D}) = \int \sum_Z \sum_{z^*} p(\mathbf{f}^*, z^*, Z, \Theta|\mathbf{x}^*, \mathcal{D})d\Theta$$

$$= \int \sum_Z \sum_{z^*} p(z^*, Z, \Theta|\mathbf{x}^*, \mathcal{D})p(\mathbf{f}^*|z^*, Z, \Theta, \mathbf{x}^*, \mathcal{D})d\Theta$$

$$= \int \sum_{Z,z^*} p(z^*|Z, \Theta, \mathbf{x}^*)p(Z, \Theta|\mathbf{x}^*, \mathcal{D})p(\mathbf{f}^*|z^*, Z, \Theta, \mathbf{x}^*, \mathcal{D})d\Theta$$

$$\approx \int \sum_{Z,z^*} p(z^*|\mathbf{x}^*, Z, \Theta)p(Z, \Theta|\mathcal{D})p(\mathbf{f}^*|z^*, Z, \Theta, \mathbf{x}^*, \mathcal{D})d\Theta$$

$$= \int \sum_Z \left[ \sum_{z^*} p(z^*|\mathbf{x}^*, Z, \Theta)p(\mathbf{f}^*|z^*, Z, \Theta, \mathbf{x}^*, \mathcal{D}) \right] \times$$
$$p(Z, \Theta|\mathcal{D})d\Theta, \tag{18}$$

where we have made use of the conditional independence $p(z^*|Z, \Theta, \mathbf{x}^*, \mathcal{D}) = p(z^*|\mathbf{x}^*, Z, \Theta)$ and a reasonable approximation $p(Z, \Theta|\mathbf{x}^*, \mathcal{D}) \approx p(Z, \Theta|\mathcal{D})$.

With the Markov chain Monte Carlo samples $\{Z_i, \Theta_i\}_{i=1}^L$ to approximate the above summation and integration over $Z$ and $\Theta$, it follows that,

$$p(\mathbf{f}^*|\mathbf{x}^*, \mathcal{D})$$
$$\approx \frac{1}{L} \sum_{i=1}^L \left[ \sum_{z^*} p(z^*|\mathbf{x}^*, Z_i, \Theta_i)p(\mathbf{f}^*|z^*, Z_i, \Theta_i, \mathbf{x}^*, \mathcal{D}) \right].$$

Therefore, the prediction for $\mathbf{f}^*$ is

$$\hat{\mathbf{f}}^* = \frac{1}{L} \sum_{i=1}^L \left[ \sum_{z^*} p(z^*|\mathbf{x}^*, Z_i, \Theta_i)\mathbb{E}(\mathbf{f}^*|z^*, Z_i, \Theta_i, \mathbf{x}^*, \mathcal{D}) \right],$$

where the expectation involved is simple to calculate since $p(\mathbf{f}^*|z^*, Z_i, \Theta_i, \mathbf{x}^*, \mathcal{D})$ is a Gaussian distribution, and $z^*$ takes values from $Z_i$ or is different from $Z_i$ with the corresponding parameters sampled from the priors.

The computation of $p(z^*|\mathbf{x}^*, Z_i, \Theta_i)$ is given as follows.

$$p(z^*|\mathbf{x}^*, Z_i, \Theta_i)$$
$$= \frac{p(z^*|Z_i, \Theta_i)p(\mathbf{x}^*|z^*, Z_i, \Theta_i)}{p(\mathbf{x}^*|Z_i, \Theta_i)}$$
$$= \frac{p(z^*|Z_i, \Theta_i)p(\mathbf{x}^*|z^*, Z_i, \Theta_i)}{\sum_{z^*} p(z^*|Z_i, \Theta_i)p(\mathbf{x}^*|z^*, Z_i, \Theta_i)}$$
$$= \frac{p(z^*|Z_i, \Theta_i)p(\mathbf{x}^*|z^*, \Theta_i)}{\sum_{z^*} p(z^*|Z_i, \Theta_i)p(\mathbf{x}^*|z^*, \Theta_i)}, \tag{19}$$

where the last equality follows from the conditional independence. If $z^* = r \in Z_i$, then $p(z^* = r|Z_i, \Theta_i) = \frac{N_{ir}}{\alpha+N}$ with $N_{ir} = \#\{z : z \in Z_i, z = r\}$ and $p(\mathbf{x}^*|z^*, \Theta_i) = p(\mathbf{x}^*|\boldsymbol{\mu}_r, \mathbf{R}_r)$. If $z^* \notin Z_i$, then $p(z^*|Z_i, \Theta_i) = \frac{\alpha}{\alpha+N}$ and

$$p(\mathbf{x}^*|z^*, \Theta_i) = \int p(\mathbf{x}^*|\boldsymbol{\mu}, \mathbf{R})p(\boldsymbol{\mu}|\boldsymbol{\mu}_0, \mathbf{R}_0)p(\mathbf{R}|\mathbf{W}_0, \nu_0)d\boldsymbol{\mu}d\mathbf{R}.$$

Unfortunately, this integral is not analytically tractable. A Monte Carlo estimate by sampling $\boldsymbol{\mu}$ and $\mathbf{R}$ from the priors can be used to reach an approximation.

Note that, if $z^* \notin Z_i$, then $\mathbb{E}(\mathbf{f}^*|z^*, Z_i, \Theta_i, \mathbf{x}^*, \mathcal{D}) = \mathbf{0}$ as a result of zero-mean Gaussian process priors. Otherwise, $\mathbb{E}(\mathbf{f}^*|z^*, Z_i, \Theta_i, \mathbf{x}^*, \mathcal{D})$ can be calculated using standard Gaussian process regression formulations.

## 5. Experiment

To evaluate the proposed infinite mixture model and the used inference and prediction methods, we perform multivariate re-

gression on a synthetic data set. The data set includes 500 examples that are generated by ancestral sampling from the infinite mixture model. The dimensions for the input and output spaces are both set to two. From the whole data, 400 examples are randomly selected as training data and the other 100 examples serve as test data.

## 5.1. Hyperparameter setting

The hyperparameters for generating data are set as follows: $a_0 = 1$, $b_0 = 1$, $\boldsymbol{\mu}_0 = \mathbf{0}$, $\mathbf{R}_0 = \mathbf{I}/10$, $\mathbf{W}_0 = \mathbf{I}/(10D)$, $\nu_0 = D$, $a_1 = 1$, $b_1 = 1$, $\mathbf{W}_1 = \mathbf{I}/M$, $\nu_1 = M$, $\mu_1 = 0$, $r_1 = 0.01$, $a_2 = 0.1$, and $b_2 = 1$. The same hyperparameters are used for inference except $\boldsymbol{\mu}_0$, $\mathbf{R}_0$ and $\mathbf{W}_0$. $\boldsymbol{\mu}_0$ and $\mathbf{R}_0$ are set to the mean $\boldsymbol{\mu}_x$ and inverse covariance $\mathbf{R}_x$ of the training data, respectively. $\mathbf{W}_0$ is set to $\mathbf{R}_x/D$.

## 5.2. Prediction Performance

By Markov chain Monte Carlo sampling, we obtain 4000 samples where only the last 2000 samples are retained for prediction. For comparison purpose, the MTLNN approach (multitask learning neural networks without ensemble learning) [16] is adopted.

Table 1 reports the root mean squared error (RMSE) on the test data for our IMMGP model and the MTLNN approach. IMMGP1 only considers the existing Gaussian process components reflected by the samples, while IMMGP2 considers to choose a new component as well. The results indicate that IMMGP outperforms MTLNN and the difference between IMMGP1 and IMMGP2 is very small.

**Table 1. Prediction errors of different methods**

| MTLNN | IMMGP1 | IMMGP2 |
|-------|--------|--------|
| 2.0659 | 0.7963 | 0.7964 |

## 6. Conclusion

In this paper, we have presented a new model called infinite mixtures of multivariate Gaussian processes and applied it to multivariate regression with good performance returned. Interesting future directions include applying this model to large-scale data, adapting it to classification problems and devising fast deterministic approximate inference techniques.

## References

[1] C. Rasmussen and C. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge, MA, 2006.

[2] S. Sun and X. Xu, "Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction", IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 2, pp. 466-475, 2011.

[3] Y. Ji and S. Sun, "Multitask multiclass support vector machines: Model and experiments", Pattern Recognition, Vol. 46, No. 3, pp. 914-924, 2013.

[4] E. Bonilla, K. Chai, and C. Williams, "Multi-task Gaussian process prediction", Advances in Neural Information Processing Systems, Vol. 20, pp. 153-160, 2008.

[5] C. Yuan, "Conditional multi-output regression", Proceedings of the Interantional Joint Conference on Neural Networks, pp. 189-196, 2011.

[6] M. Alvarez and N. Lawrence, "Computationally efficient convolved multiple output Gaussian processes", Journal of Machine Learning Research, Vol. 12, pp. 1459-1500, 2011.

[7] C. Rasmussen and Z. Ghahramani, "Infinite mixtures of Gaussian process experts", Advances in Neural Information Processing Systems, Vol. 14, pp. 881-888, 2002.

[8] V. Tresp, "Mixtures of Gaussian processes", Advances in Neural Information Processing Systems, Vol. 13, pp. 654-660, 2001.

[9] Y. Teh, "Dirichlet processes", in Encyclopedia of Machine Learning, Springer-Verlag, Berlin, Germany, 2010.

[10] E. Meeds and S. Osindero, "An alternative infinite mixture of Gaussian process experts", Advances in Neural Information Processing Systems, Vol. 18, pp. 883-890, 2006.

[11] C. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, New York, 2006.

[12] C. Rasmussen, "The infinite Gaussian mixture model", Advances in Neural Information Processing Systems, Vol. 12, pp. 554-560, 2000.

[13] R. Neal, "Markov chain sampling methods for Dirichlet process mixture models", Technical Report 9815, Department of Statistics, University of Toronto, 1998.

[14] R. Neal, "Probabilistic inference using Markov chain Monte Carlo methods", Technical Report CRG-TR-93-1, University of Toronto, 1993.

[15] C. Antoniak, "Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems", Annals of Statistics, Vol. 2, No. 6, pp. 1152-1174, 1974.

[16] S. Sun, "Traffic flow forecasting based on multitask ensemble learning", Proceedings of the ACM SIGEVO World Summit on Genetic and Evolutionary Computation, pp. 961-964, 2009.