Automatic Model Selection Algorithm Based on BYY Harmon Learning for Mi ture of Gaussian Process Functional Regressions Models

Xiangyang Guo, Tao Hong, and Jinwen Ma(✉)

Department of Information and Computational Sciences, School of Mathematical Sciences and LMAM, Peking University, Beijing 100871, China
jwma@math.pku.edu.cn

Abstract. For finite mixture models, determining the number of components is referred to as model selection. This paper puts forward an automatic model selection algorithm based on Bayesian Ying-Yang (BYY) harmony learning for mixture of Gaussian process functional regressions (mix-GPFR) models. BYY harmony learning has been successfully applied to the model selection problem of Gaussian mixture models (GMMs), but it cannot be directly used for that of mix-GPFR models. We find out the cause of this problem and propose a coping mechanism of curve reconstruction based on Gaussian process (GP) models, through which, we transform a mix-GPFR model into a GMM. Thus, we can make model selection for mix-GPFR models via BYY harmony learning. Experimental results show that our proposed automatic model selection algorithm can find the optimal number of components in a multi-source curve dataset.

Ke ords: Mixtures of Gaussian Process Functional Regressions · Model Selection · Bayesian Ying-Yang Harmony Learning · Curve Reconstruction

1 Introduction

Gaussian process (GP) models are an effective tool for Bayesian nonlinear nonparametric classification and regression, e.g., classifying the images of handwritten digits and modeling the inverse dynamics of a robot arm [1]. However, they cannot deal with multi-source curve datasets effectively. To overcome this limitation, mixture of Gaussian process functional regressions (mix-GPFR) models were proposed [2, 3] and then extensive research has been devoted to estimating their parameters, analyzing their performance, and applying them to real-world problems [4–8].

Like other finite mixture models, mix-GPFR models also face the problem of model selection, namely determining the number of Gaussian process functional regression (GPFR) components. Since an inappropriate number of GPFR components will inevitably lead to poor generalization ability, model selection is of great importance. In addition to making model selection utilizing domain knowledge or experience, we can

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
D.-S. Huang et al. (Eds.): ICIC 2023, LNAI 14089, pp. 391–403, 2023.
https://doi.org/10.1007/978-981-99-4752-2_33

also design automatic model selection algorithms. The traditional method is to choose the optimal number of GPFR components through certain statistical selection criterion. For example, Qiang et al. [6] proposed the splitting expectation-maximization (SEM) algorithm based on the Bayesian information criterion (BIC) [9]. However, all the existing statistical selection criteria often cause an improper number of GPFR components and the use of a statistical selection criterion incurs a high time complexity, since we need to repeat the whole parameter estimating process for different numbers of GPFR components. Moreover, stochastic simulation methods, such as reversible jump Markov chain Monte Carlo [10] and Dirichlet processes [11], have also been used to deal with the model selection problem of mix-GPFR models [5, 7, 8]. However, these methods require collecting a large number of samples, which results in a high computational cost.

For Gaussian mixture models (GMMs), the automatic model selection algorithms based on Bayesian Ying-Yang (BYY) harmony learning [12, 13] have acquired better results and higher computation speed than those based on statistical selection criteria and stochastic simulation methods [14–20] [illegible]

Here, since the indicator variable is scalar, we denote it as z instead of . For the GMM, we establish the following BYY system: $q(z = g) = \pi_g$; $q(\ |z = g) = \mathcal{N}(\ |\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$; $p(\ ) = \frac{1}{I}\sum_{i=1}^{I}\delta(\ -\ _i)$, i.e. the empirical density function;

$$p(z = g|\ ) = \frac{\pi_g \mathcal{N}(\ |\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)}{\sum_{s=1}^{G}\pi_s \mathcal{N}(\ |\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)}. \tag{3}$$

Moreover, we ignore the regularization term r, i.e. set $r = 1$. Then, we have

$$H(p||q) = J(\Theta) = \frac{1}{I}\sum_{i=1}^{I}\sum_{g=1}^{G}\frac{\pi_g \mathcal{N}(\ _i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)}{\sum_{s=1}^{G}\pi_s \mathcal{N}(\ _i|\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)}\ln\left(\pi_g \mathcal{N}(\ _i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right), \tag{4}$$

where $J(\Theta)$ is called harmony function and $\Theta = \{\pi_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}_{g=1}^{G}$.

According to BYY harmony learning, the maximum of $J(\Theta)$ corresponds to the optimal number of Gaussian components and the best parameters [14–20]. Hence, we can make model selection and estimate the parameters by maximizing $J(\Theta)$. In the process of maximizing $J(\Theta)$, the mixing proportions of the redundant Gaussian components converge to zero. Compared with the automatic model selection algorithms based on statistical selection criteria and stochastic simulation methods, those based on BYY harmony learning have acquired better results and higher computation speed [14–20].

3 Automatic Model Selection Algorithm Based on BYY Harmon Learning

Firstly, we briefly introduce the mix-GPFR model. A GP is a collection of random variables, any finite subset of which is subject to a Gaussian distribution [1]. To specify a GP $\{f(\ )|\ \in \mathcal{X} \subseteq \mathbb{R}^D\}$, we only need to determine its mean function $m(\ )$ and covariance function $c(\ ,\ ')$, where.

$$m(\ ) = \mathbb{E}[f(\ )] \text{ and } c(\ ,\ ') = \mathbb{E}[(f(\ ) - m(\ ))(f(\ ') - m(\ '))]. \tag{5}$$

whereupon the GP is denoted as

$$f(\ ) \sim \mathcal{GP}(m(\ ), c(\ ,\ ')). \tag{6}$$

In mix-GPFR models, since $D = 1$, we denote the input as x instead of . Then, a mix-GPFR model with G GPFR components can be established through the following formulae:

$$q(z = g) = \pi_g,\ where\ \pi_g \geq 0 \text{ and } \sum_{g=1}^{G}\pi_g = 1; \tag{7}$$

$$q(y(x)|z = g) = \mathcal{GPFR}(x|\mathbf{b}_g, \boldsymbol{\theta}_g, r_g) = \mathcal{GP}\left(m(x|\mathbf{b}_g), c(x, x'|\boldsymbol{\theta}_g) + r_g^{-1}\delta(x, x')\right). \tag{8}$$

In Eq. (8), $\delta\left(x, x'\right)$ is the Kronecker delta function,

$$m\left(x|\mathbf{b}_g\right) = \varphi(x)^T\mathbf{b}_g \text{ and } c\left(x, x'|\boldsymbol{\theta}_g\right) = \theta_{g0}^2 \exp\left\{-\frac{\left(x-x'\right)^2}{2\theta_{g1}^2}\right\}, \tag{9}$$

where $\varphi(x) = [\varphi_1(x), \varphi_2(x), \ldots, \varphi_P(x)]^T$ is a column vector of B-splines [21] and $c\left(x, x'|\boldsymbol{\theta}_g\right)$ is referred to as the squared exponential covariance function. θ_{g0}, θ_{g1}, and r_g are positive parameters.

The Ying machine of the mix-GPFR model is

$$q(z=g, y(x)) = q(z=g)q(y(x)|z=g) = \pi_g \mathcal{GPFR}\left(x|\mathbf{b}_g, \boldsymbol{\theta}_g, r_g\right) \tag{10}$$

and its Yang machine is

$$p(z=g, y(x)) = p(y(x))p(z=g|y(x)) = p(y(x))\frac{\pi_g \mathcal{GPFR}\left(x|\mathbf{b}_g, \boldsymbol{\theta}_g, r_g\right)}{\sum_{s=1}^G \pi_s \mathcal{GPFR}(x|\mathbf{b}_s, \boldsymbol{\theta}_s, r_s)}. \tag{11}$$

We denote a training curve dataset as $\mathcal{D} = \{\mathcal{C}_i\}_{i=1}^I$, where $\mathcal{C}_i = \{(x_{in}, y_{in})\}_{n=1}^{N_i}$ represents a training curve of length N_i. It is generally assumed that $x_{i1}, \ldots, x_{iN_i}$ are randomly distributed in the interval $[x_{\min}, x_{\max}]$ $(i = 1, \ldots, I)$. Let ${}_i = \left[x_{i1}, \ldots, x_{iN_i}\right]^T$, ${}_i = \left[y_{i1}, \ldots, y_{iN_i}\right]^T$, and $\Theta = \left\{\pi_g, \mathbf{b}_g, \boldsymbol{\theta}_g, r_g\right\}_{g=1}^G$. For the mix-GPFR model,

$$H(p||q) = \sum_{g=1}^G \int p(y(x))p(z=g|y(x))\ln(q(z=g)q(y(x)|z=g))\mathrm{d}y(x) \tag{12}$$

cannot be approximated by

$$J(\Theta) = \frac{1}{I}\sum_{i=1}^I \sum_{g=1}^G \frac{\pi_g \mathcal{N}\left({}_i|\mathbf{m}_{ig}, \mathbf{C}_{ig}\right)}{\sum_{s=1}^G \pi_s \mathcal{N}({}_i|\mathbf{m}_{is}, \mathbf{C}_{is})} \ln\left(\pi_g \mathcal{N}\left({}_i|\mathbf{m}_{ig}, \mathbf{C}_{ig}\right)\right) \tag{13}$$

with $\mathbf{m}_{ig} = m\left({}_i|\mathbf{b}_g\right)$ and $\mathbf{C}_{ig} = c\left({}_i, {}_i|\boldsymbol{\theta}_g\right) + r_g^{-1}\mathbf{I}_{N_i}$, where $\mathbf{I}_{N_i}$ is the

$\hat{\mathcal{C}}_i = \{(x_n, \hat{y}_{in})\}_{n=1}^{N}$ from $\hat{f}_i(x)$ with $x_n = x_{\min} + (n-1)\Delta$. During sampling, the variance of Gaussian noise is

$$\sigma_1^2 = \frac{1}{N_i}\sum_{n=1}^{N_i}\left(y_{in} - \hat{f}_i(x_{in})\right)^2. \tag{14}$$

It is clear that

$$\sigma_2^2 = \frac{1}{N_i}\sum_{n=1}^{N_i}(y_{in} - f_i(x_{in}))^2 \tag{15}$$

is an unbiased estimate of the variance of Gaussian noise in the process of sampling $\mathcal{C}_i$ from $f_i(x)$. Hence, σ_1^2 is a good estimate of the variance on the assumption that there are no significant differences between $f_i(x)$ and $\hat{f}_i(x)$. Intuitively, $\hat{\mathcal{C}}_i$ is a good approximation of $\mathcal{C}_i$. That is to say, the difference between the posterior mean functions recovered from $\hat{\mathcal{C}}_i$ and $\mathcal{C}_i$, respectively, is small, which will be validated through experiments in Sect. 4.

Let $\hat{\mathcal{D}} = \hat{\mathcal{C}}_i {}_{i=1}^{I}$, $= [x_1, x_2, \ldots, x_N]^T$, and $\hat{}_i = \left[\hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{iN}\right]^T$. $\hat{}_i$ can be regarded as a sample of the following GMM:

$$q(z = g) = \pi_g \text{ where } \pi_g \geq 0 \text{ and } \sum_{g=1}^{G}\pi_g = 1;\ q\left(\hat{}|z = g\right) = \mathcal{N}\left(\hat{}|\mathbf{m}_g, \mathbf{C}_g\right), \tag{16}$$

where $\mathbf{m}_g = m\left(\ |\mathbf{b}_g\right)$ and $\mathbf{C}_g = c\left(\ ,\ |\boldsymbol{\theta}_g\right) + r_g^{-1}\mathbf{I}_N$. Its Ying machine is

$$q\left(z = g, \hat{}\right) = q(z = g)q\left(\hat{}|z = g\right) = \pi_g\mathcal{N}\left(\hat{}|\mathbf{m}_g, \mathbf{C}_g\right) \tag{17}$$

and its Yang machine is

$$p\left(z = g, \hat{}\right) = p\left(\hat{}\right)p\left(z = g|\hat{}\right) = p\left(\hat{}\right)\frac{\pi_g\mathcal{N}\left(\hat{}|\mathbf{m}_g, \mathbf{C}_g\right)}{\sum_{s=1}^{G}\pi_s\mathcal{N}\left(\hat{}|\mathbf{m}_s, \mathbf{C}_s\right)}. \tag{18}$$

Then, its corresponding harmony function is

$$J(\Theta) = \frac{1}{I}\sum_{i=1}^{I}\sum_{g=1}^{G}\frac{\pi_g\mathcal{N}\left(\hat{}_i|\mathbf{m}_g, \mathbf{C}_g\right)}{\sum_{s=1}^{G}\pi_s\mathcal{N}\left(\hat{}_i|\mathbf{m}_s, \mathbf{C}_s\right)}\ln\left(\pi_g\mathcal{N}\left(\hat{}_i|\mathbf{m}_g, \mathbf{C}_g\right)\right). \tag{19}$$

As is the case with GMMs, the maximum of $J(\Theta)$ corresponds to the optimal number of GPFR components and the best parameters. Therefore, we can make model selection and learn the parameters by maximizing $J(\Theta)$ through numerical optimization methods.

After the training process, we can determine the class of a training curve according to the maximum a posteriori probability, i.e. let

$$z_i = \underset{g\in\{1,2,\ldots,G\}}{\operatorname{argmax}}\frac{\pi_g\mathcal{N}\left(\hat{}_i|\mathbf{m}_g, \mathbf{C}_g\right)}{\sum_{s=1}^{G}\pi_s\mathcal{N}\left(\hat{}_i|\mathbf{m}_s, \mathbf{C}_s\right)}(i = 1, 2, \ldots, I). \tag{20}$$

The redundant GPFR components don't get any training curves due to their very small mixing proportions. The class of a test curve can also be determine in this way. Besides, for a test curve, we can predict the test outputs by calculating their conditional distribution given the known outputs. The details are referred to [4–8].

4 E perimental Results

In this section, we use nine synthetic datasets and two real-world datasets to verify the effectiveness of our proposed automatic model selection algorithm. We compare mix-GPFR models trained via our proposed algorithm with GP models, mix-GP models, GPFR models, and mix-GPFR models trained through the traditional EM algorithm [2, 3] and the SEM algorithm [6].

Since we are mainly concerned with the prediction ability of mix-GPFR models, the rooted mean square error (RMSE) is chosen as the evaluation metric. It is assumed that there are T test curves and the test outputs of the t th ($t = 1, 2, \ldots, T$) test curve are $y_{t1}, y_{t2}, \ldots, y_{tM}$, whose corresponding prediction values are $\hat{y}_{t1}, \hat{y}_{t2}, \ldots, \hat{y}_{tM}$, respectively. It follows that

$$\text{RMSE} = \sqrt{\frac{1}{TM} \sum_{t=1}^{T} \sum_{m=1}^{M} \left(y_{tm} - \hat{y}_{tm}\right)^2}. \tag{21}$$

Apparently, a smaller RMSE indicates a better prediction result.

4.1 On S nthetic Datasets

The nine synthetic datasets are denoted as $\mathcal{S}_2, \mathcal{S}_3, \ldots, S_{10}$, respectively, where the subscripts represent the numbers of components. For each component, we sample 20 training curves and 10 test curves from a GP with non-zero mean function. The mean functions and parameters of the Gaussian processes used to generate the nine synthetic datasets are list in Table 1, where $\mathcal{S}_l$ ($l = 2, 3, \ldots, 10$) are generated by the first l GPs. Each curve consists of 100 points, whose inputs are randomly distributed in $[-3, 3]$. The 60 points on the left side of a test curve are known and the 40 ones on the right side are used for testing.

Firstly, we demonstrate the effectiveness of curve reconstruction based on GP models through experiments. A training curve is randomly chosen from each component in $\mathcal{S}_9$. Figure 1 presents the reconstruction curves of the nine training curves. Figure 1 is composed of 9 sub-figures, each of which presents a training curve, its reconstruction curve, and their posterior mean functions. As can be seen from the figure, although there are significant differences between a training curve and its reconstruction curve, their posterior mean functions are similar, which implies that our proposed curve reconstruction based on GP models is effective.

When testing our proposed algorithm, G is initialized as $l+3$ for $\mathcal{S}_l$. To illustrate the bad effect of a wrong number of GPFR components on prediction ability, we train mix-GPFR models consisting of $l-1$ and $l+1$ GPFR components via the EM algorithm [2, 3], which are denoted as "mix-GPFR (-1)" and "mix-GPFR (+1)", respectively. Similarly, mix-GP models with $l-1$ and $l+1$ GP components are denoted as "mix-GP (-1)" and "mix-GP (+1)", respectively. Besides, P is set to be 20. Table 2 presents the experimental results.

From Table 2, we see that the RMSEs of the GPFR model and the mix-GPFR model are smaller than those of the GP model and the mix-GP model, respectively, which

Table 1. Mean functions and parameters of the Gaussian processes used to generate the nine synthetic datasets.

Mean functions	$\boldsymbol{\theta}^T$	$\sqrt{r^{-1}}$
x^2	[0.5, 0.5]	0.15
$\left(-4(x+1.5)^2+9\right)1_{\{x<0\}}+\left(4(x-1.5)^2-9\right)1_{\{x\geq 0\}}$	[0.528, 0.4]	0.144
$8\sin(1.5x-1)$	[0.556, 0.3]	0.139
$\sin(1.5x)+2x-5$	[0.583, 0.2]	0.133
$\sin(4x)-0.5x^2-2x$	[0.611, 0.1]	0.128
$-x^2$	[0.639, 0.1]	0.122
$\left(4(x+1.5)^2-9\right)1_{\{x<0\}}+\left(-4(x-1.5)^2+9\right)1_{\{x\geq 0\}}$	[0.667, 0.2]	0.117
$5\cos(3x+2)$	[0.694, 0.3]	0.111
$\cos(1.5x)-2x+5$	[0.722, 0.4]	0.106
$\cos(4x)+0.5x^2+2x$	[0.75, 0.5]	0.1

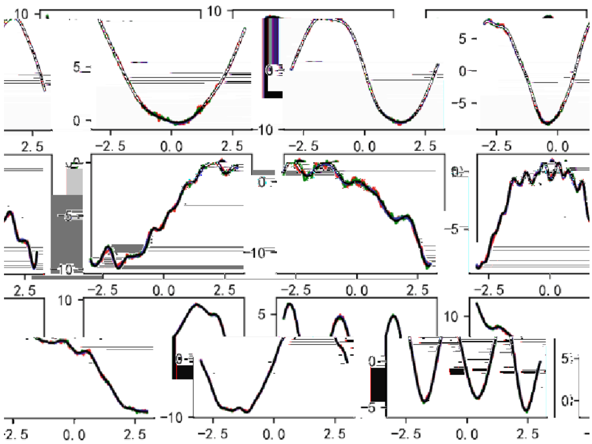

Fig. 1. The results of curve reconstruction on the synthetic datasets. The red, green, blue, and black curves represent the original curve, the reconstructed curve, the posterior mean function of the original curve, and the posterior mean function of the reconstructed curve, respectively.

demonstrates the effectiveness of modeling the mean function as a linear combination of B-splines. By comparing the mix-GP (mix-GPFR) model and the GP (GPFR) model, the need for introducing the mixture structure is demonstrated. Furthermore, we can see that a wrong number of GPFR components affects the prediction results badly. For $\mathcal{S}_2, \mathcal{S}_3, \ldots, \mathcal{S}_9$, both the SEM algorithm and our proposed algorithm find the correct number of GPFR components and their RMSEs are close. However, the time complexity of the SEM algorithm is higher than that of our proposed algorithm. On the one hand, the SEM algorithm needs to repeat the whole parameter learning process for different numbers of GPFR components. On the other hand, since different training curves have different inputs, we have to use the loop structure when programming. This is the main reason why the SEM algorithm has a high time complexity. For $\mathcal{S}_{10}$, since the SEM algorithm fails to find the true number of GPFR components, its RMSE is larger than that of our proposed algorithm.

Taking $\mathcal{S}_9$ for example, we present the clustering results of our proposed algorithm in Fig. 2, where different colors represent different components. On the left and right sides of Fig. 2 are the clustering results of our proposed algorithm on the training and test datasets, respectively. It is clear that our proposed algorithm correctly find all the components.

Table 2. RMSE and running time of all the methods on the synthetic datasets.

	$\mathcal{S}_2$		$\mathcal{S}_3$		$\mathcal{S}_4$	
	RMSE	Time (min)	RMSE	Time (min)	RMSE	Time (min)
GP	5.5831	6.87	4.7878	9.88	4.7798	13.09
mix-GP (-1)	5.5239	6.12	4.6125	8.90	4.3580	15.84
mix-GP (+1)	4.8240	7.39	4.6035	17.40	4.3488	23.31
GPFR	5.0759	6.68	4.6864	12.42	4.3051	17.72
mix-GPFR (-1)	5.0214	8.07	0.9416	15.95	0.9510	18.93
mix-GPFR (+1)	1.6846	14.20	1.0680	22.66	0.9319	25.79
mix-GPFR (SEM)	**0.4312**	20.63	0.4856	41.59	0.5469	58.97
mix-GPFR (BYY)	0.4401	9.46	**0.4746**	15.03	**0.5403**	18.64
	$\mathcal{S}_5$		$\mathcal{S}_6$		$\mathcal{S}_7$	
	RMSE	Time (min)	RMSE	Time (min)	RMSE	Time (min)
GP	4.9213	17.85	4.9897	15.07	5.3096	20.47
mix-GP (-1)	4.4775	15.03	4.3834	20.14	4.3025	30.66
mix-GP (+1)	4.5205	28.59	4.3813	29.19	4.3082	30.53
GPFR	4.8079	26.73	4.8649	24.25	4.9871	21.45
mix-GPFR (-1)	0.8756	31.66	1.0776	29.67	1.3295	32.09
mix-GPFR (+1)	0.9252	30.58	1.0270	35.37	1.0281	38.44
mix-GPFR (SEM)	0.5638	81.34	**0.6057**	87.52	**0.6540**	92.78
mix-GPFR (BYY)	**0.5573**	25.49	0.6137	23.66	0.6571	27.82
	$\mathcal{S}_8$		$\mathcal{S}_9$		$\mathcal{S}_{10}$	
	RMSE	Time (min)	RMSE	Time (min)	RMSE	Time (min)
GP	4.8180	19.67	4.4758	17.82	4.8438	21.67
mix-GP (-1)	4.4904	24.13	4.1223	32.36	4.5730	32.78

(*continued*)

Table 2. (*continued*)

	$\mathcal{S}_2$		$\mathcal{S}_3$		$\mathcal{S}_4$	
	RMSE	Time (min)	RMSE	Time (min)	RMSE	Time (min)
mix-GP (+1)	4.4818	23.70	4.1214	30.95	4.5878	28.14
GPFR	4.6871	26.73	4.3686	21.67	4.6865	20.97
mix-GPFR (-1)	1.5325	33.78	1.0585	40.27	1.5279	41.56
mix-GPFR (+1)	1.1891	35.96	0.9789	39.38	1.0343	49.78
mix-GPFR (SEM)	0.6448	99.49	0.6233	116.85	1.4379	130.65
mix-GPFR (BYY)	**0.6421**	28.91	**0.6199**	28.62	**0.6317**	32.46

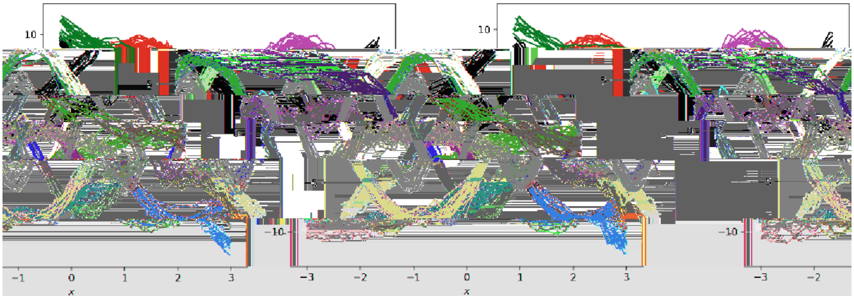

Fig. 2. Clustering results of our proposed automatic model selection algorithm on $\mathcal{S}_7$ and $\mathcal{S}_9$.

4.2 On Real-World Datasets

Here, we utilize the electricity load dataset issued by the Northwest China Grid Company [8], which records electricity loads every 15 min in 2009 and 2010. Hence, daily electricity loads can be regarded as a curve with 96 points. We divide the dataset into two sub-datasets according to the year, which are referred to as $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. Each sub-dataset consists of 200 training curves for and 165 test curves. Moreover, the 56 points on the left side of a test curve are known and the 40 ones on the right side are used for testing.

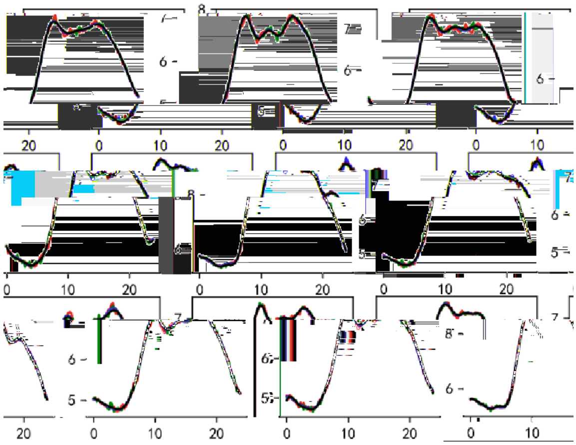

Fig. 3. The results of curve reconstruction on the electricity load dataset. The red, green, blue, and black curves represent the original curve, the reconstructed curve, the posterior mean function of the original curve, and the posterior mean function of the reconstructed curve, respectively.

Although all the curves have the same inputs, we treat them as if they don't have the same inputs. Like the synthetic datasets, we randomly choose 9 training curves of $\mathcal{R}_1$, whose reconstruction curves are presented in Fig. 3. As can be seen from the figure, our proposed curve reconstruction based on GP models is effective for the electricity load dataset.

Since the numbers of components in $\mathcal{R}_1$ and $\mathcal{R}_2$ are unknown, we set $G = 3, 6, 9, 12, 15$ for the mix-GP and mix-GPFR models trained using the EM algorithm. For our proposed algorithm and the SEM algorithm, G is set to be 15. Besides, P is set to be 30. The experimental results are described in Table 3. For $\mathcal{R}_1$ and $\mathcal{R}_2$, the RMSE of our proposed algorithm is smaller than that of the SEM algorithm since the number of components given by the SEM algorithm is smaller than the optimal one. The clustering results are presented in Fig. 4. On the left and right sides of Fig. 4 are the clustering results of our proposed algorithm on the training and test datasets, respectively. For $\mathcal{R}_1$ and $\mathcal{R}_2$, the numbers of components gotten via our proposed algorithm are 13 and 11, respectively. As can be seen from Fig. 4, curves belonging to different components are obviously different in a certain input interval, that is to say, the clustering results given by the algorithm are reasonable.

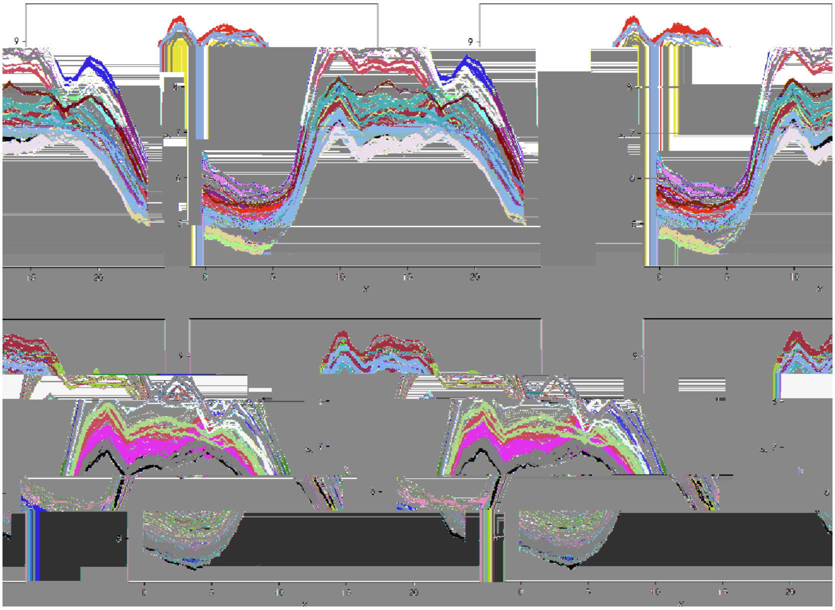

Fig. 4. Clustering results of our proposed automatic model selection algorithm on $\mathcal{R}_1$ and $\mathcal{R}_2$.

Table 3. RMSE and running time of all the methods on $\mathcal{R}_1$ and $\mathcal{R}_2$.

	$\mathcal{R}_1$		$\mathcal{R}_2$	
	RMSE	Time (min)	RMSE	Time (min)
GP	0.9599	19.43	0.8977	20.39
mix-GP (3)	0.9390	20.33	0.8846	21.42
mix-GP (6)	0.9387	22.54	0.8854	22.66
mix-GP (9)	0.9380	25.99	0.8853	26.09
mix-GP (12)	0.9395	29.83	0.8847	31.23
mix-GP (15)	0.9401	34.76	0.8872	36.91
GPFR	0.5584	21.30	0.5499	21.59
mix-GPFR (3)	0.2089	24.45	0.2133	20.64
mix-GPFR (6)	0.1701	25.76	0.1731	24.77
mix-GPFR (9)	0.1356	28.93	0.1455	29.45
mix-GPFR (12)	0.1248	34.65	0.1314	33.63
mix-GPFR (15)	0.1178	35.88	0.1301	36.78
mix-GPFR (SEM)	0.1323	150.76	0.1377	170.17
mix-GPFR (BYY)	**0.1109**	33.97	**0.1201**	34.58

5 Conclusion

In this paper, we propose an automatic model selection algorithm based on BYY harmony learning for mix-GPFR models. Since different training curves have different inputs, BYY harmony learning cannot be directly applied to the model selection problem of mix-GPFR models. To tackle this, we propose curve reconstruction based on GP models, through which, we unify the inputs of all the training curves. Then, we can make model selection for mix-GPFR models via BYY harmony learning. Experimental results on synthetic and real-world datasets show that our proposed automatic model selection algorithm can find the optimal number of components in a multi-source curve dataset and its time complexity is lower than that of the SEM algorithm.

Ackno ledgement. This work is supported by the National Key R & D Program of China (2018AAA0100205).

References

1. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
2. Shi, J.Q., Wang, B.: Curve prediction and clustering with mixtures of Gaussian process functional regression models. Statist. Comput. **18**, 267–283 (2008)
3. Shi, J.Q., Choi, T.: Gaussian Process Regression Analysis for Functional Data. CRC Press, Boca Raton (2011)
4. Wu, D., Ma, J.: A DAEM algorithm for mixtures of Gaussian process functional regressions. In: Huang, D.-S., Han, K., Hussain, A. (eds.) ICIC 2016. LNCS (LNAI), vol. 9773, pp. 294–303. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42297-8_28
5. Qiang, Z., Ma, J.: Automatic model selection of the mixtures of Gaussian processes for regression. In: Hu, X., Xia, Y., Zhang, Y., Zhao, D. (eds.) ISNN 2015. LNCS, vol. 9377, pp. 335–344. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25393-0_37
6. Qiang, Z., Luo, J., Ma, J.: Curve clustering via the split learning of mixtures of Gaussian processes. In: 2016 IEEE 13th International Conference on Signal Processing (ICSP), pp. 1089–1094 (2016)
7. Qiang, Z., Ma, J.: Model selection prediction for the mixture of Gaussian processes with RJMCMC. In: Shi, Z., Pennartz, C., Huang, T. (eds.) ICIS 2018. IAICT, vol. 539, pp. 310–317. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01313-4_33
8. Li, T., Ma, J.: Dirichlet process mixture of Gaussian process functional regressions and its variational EM algorithm. Pattern Recogn. **134**, 109–129 (2023)
9. Schwarz, G.: Estimating the dimension of a model. Ann. Statist. **6**(2), 461–464 (1978)
10. Richardson, S., Green, P.J.: On Bayesian analysis of mixtures with an unknown number of components. J Royal Statist. Soc. (Ser. B) **59**(4), 731–792 (1997)
11. Escobar, M.D., West, M.: Bayesian density estimation and inference using mixtures. J. Am. Statist. Assoc. **90**(430), 577–588 (1995)
12. Xu, L.: Ying-Yang machine: a Bayesian-Kullback scheme for unified learnings and new results on vector quantization. In: Proceedings of the 1995 International Conference on Neural Information Processing, vol. 2, pp. 977–988 (1995)
13. Xu, L.: Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on Gaussian mixtures, three-layer nets and ME-RBF-SVM models. Int. J. Neural Syst. **11**(1), 43–69 (2001)

14. Chen, G., Li, L., Ma, J.: A gradient BYY harmony learning algorithm for straight line detection. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.) ISNN 2008. LNCS, vol. 5263, pp. 618–626. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87732-5_69
15. Li, L., Ma, J.: A BYY scale-incremental EM algorithm for Gaussian mixture learning. Appl. Math. Comput. **205**(2), 832–840 (2008)
16. Li, L., Ma, J.: A BYY split-and-merge EM algorithm for Gaussian mixture learning. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.) ISNN 2008. LNCS, vol. 5263, pp. 600–609. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87732-5_67
17. Ma, J., Wang, T., Xu, L.: A gradient BYY harmony learning rule on Gaussian mixture with automated model selection. Neurocomputing **56**, 481–487 (2004)
18. Ma, J., Gao, B., Wang, Y., Cheng, Q.: Conjugate and natural gradient rules for BYY harmony learning on Gaussian mixture with automated model selection. Int. J. Pattern Recogn. Artif. Intell. **19**(5), 701–713 (2005)
19. Ma, J., Liu, J.: The BYY annealing learning algorithm for Gaussian mixture with automated model selection. Pattern Recogn **40**(7), 2029–2037 (2007)
20. Ma, J., He, X.: A fast fixed-point BYY harmony learning algorithm on Gaussian mixture with automated model selection. Pattern Recogn. Lett. **29**(6), 701–711 (2008)
21. Boor, C.D.: On calculating with B-splines. J. Approximation Theor. **6**, 50–62 (1972)