Automatic Model Selection of the Mixtures of Gaussian Processes for Regression

Zhe Qiang and Jinwen Ma

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

Abstract. For the learning of mixtures of Gaussian processes, model selection is an important but difficult problem. In this paper, we develop an automatic model selection algorithm for mixtures of Gaussian processes in the light of the reversible jump Markov chain Monte Carlo framework for Gaussian mixtures. In this way, the component number and the parameters are updated according the five types of random moves and model selection can be made automatically. The key idea is that the moves of component splitting or merging preserve the zeroth, first and second moments of the components so that the covariance parameters of the new components can be related to the origin ones. It is demonstrated by the simulation experiments that this automatic model selection algorithm is feasible and effective.

Keywords: Mixtures of Gaussian processes, Reversible jump MCMC, Model selection, Regression, Split and merge moves.

1 Introduction

As a powerful statistical learning tool, Gaussian Process (GP) is widely used in machine learning and pattern recognition [1]. Since a single GP model cannot deal with the multimodality dataset, the Mixture of Gaussian Processes (MGP) [2] have been developed to model a general multimodality dataset. Obviously, MGP can also overcome a major disadvantage of the Gaussian process modeling that calculating the inversion of an $\times$ covariance matrix requires the time complexity of $(^3)$ for a training dataset with points. Structurally, Shi et al. [3],[4] considered MGP as a hierarchical model and fit the data in two levels. Moreover, they proposed the hybrid Markove Chain Monte-Carlo algorithm to train the covariance parameters and then to utilize BIC to determine the number of GP components in the mixture. However, BIC is not so effective for the model selection on the mixtures of Gaussian processes.

On the other hand, Green [5] proposed the reversible jump Markov chain Monte Carlo (RJMCMC) framework to determine the dimension of parameters through the Markov chain Monte Carlo simulation. Later on, according to this theory, Richardson and Green [6] developed a RJMCMC approach to decide the number of actual components in the Gaussian mixture. Although this approach

X. Hu et al. (Eds.): ISNN 2015, LNCS 9377, pp. 335–344, 2015.
DOI: 10.1007/978-3-319-25393-0_37

is effective for Gaussian mixtures, its idea can also used to design the learning algorithm for MGPs and solve the model selection problem in this case. However, the structure of the likelihood function of MGP is quite different from that of Gaussian mixture so that the split and merge moves in the algorithm cannot be implemented directly.

In this paper, we develop an automatic model selection algorithm for MGPs in the light of the RJMCMC framework by making the split and merge moves feasible and effective. As we find, the difficulty in the split moves focuses on that the overall dispersion should keep relatively constant when a covariance matrix is split into two matrices. In order to solve this difficulty, we can keep the first two moments of the components to be same during a split or merge move. By mathematics analysis, we find that once the sampling interval is set small enough, these moments remains almost constant. In this way, our automatic model selection algorithm can be constructed effectively. It is demonstrated by simulation experiments that this automatic model selection algorithm is feasible and effective in the same way as the RJMCMC algorithm for Gaussian mixtures.

The rest of the paper is organised as follows. In Section 2, we revisit the mixture of Gaussian processes and give the Bayesian forms of the priors of the parameters. Section 3 presents the automatic model selection algorithm with five move types. Simulation experiments are conducted in Section 4. We conclude briefly in Section 5.

2 The Hierarchical Mixtures of Gaussian Processes

2.1 The Basic Model

For clarity, we consider MGP as the hierarchical mixture of Gaussian processes described Shi et al. [3]. Clearly, it tries to model a large dataset with groups of repeated measurements. Actually, the lower-level model fits the measurements in the same group, while the higher-level model tries to describe the heterogeneity among different groups. For example, the dataset used in [4] is a set of repeated standing-up trajectories corresponding to different paraplegic patients. For this case, the lower-level model fitted those standing-up trajectories of paraplegic patients in the same type, while the higher-level model described the heterogeneity among different types of paraplegic patients.

Mathematically, in the higher-level model, we let $\{(\ _{mn}\ \ _{mn})\ \ = 1\ \cdots\ \ \}$ be curves belong to a number of Gaussian processes, which can represent all the points on the training curves. The mixture model can be given by

$$\boldsymbol{y}_m \sim \sum_{k=1}^{K}\ _{k}\ \ (\boldsymbol{\theta}_k) \tag{1}$$

In the lower-level model, each curve is assumed to be a function $_{m}(\cdot)$ plus a white noise, that is,

$$_{mn} = _{m}(\ _{mn}) + \ _{mn}$$

where $_{mn} \sim \mathcal{N}(0\ \ ^{2}_{k})$. The th curve follows a Gaussian process if any finite subsequence or subset of the curve, say $\mathcal{D}_m = \{\boldsymbol{X}_m\ \boldsymbol{Y}_m\} = \{(\ _{mn}\ \ _{mn})\ \ =$

$1 \cdots \ _m\}$, follows a Gaussian distribution if $\boldsymbol{Y}_m \sim \mathcal{N}(\mathbf{0}\ \boldsymbol{\Sigma}(\boldsymbol{X}_m\ \boldsymbol{X}_m;\boldsymbol{\theta}_k))$, where

$$\boldsymbol{\Sigma}(\ _{mi}\ \ _{mj};\boldsymbol{\theta}_k) = \ (\ _{mi}\ \ _{mj};\boldsymbol{\theta}_k) + \ ^2_k\ _{ij}$$

where

$$(\ _{mi}\ \ _{mj};\boldsymbol{\theta}_k) = \ _k \exp\left(-\frac{1}{2}\ _k(\ _{mi} - \ _{mj})^2\right).$$

So, all the covariance parameters are $\boldsymbol{\theta}_k = (\ _k\ \ _k\ \ ^2_k)$.

For a test input $\ ^*$, if we assume it is on the $\ $th curve, and the $\ $th curve is belong to the $\ $-th component, i.e., Gaussian process, the mean and variance of its prediction can be obtained as follows:

$$[\ _m(\ ^*)|\mathcal{D}_m] = \boldsymbol{\sigma}^T(\ ^*)\ \ ^{-1}(\boldsymbol{X}_m\ \boldsymbol{X}_m;\boldsymbol{\theta}_k)\boldsymbol{Y}_m;$$

$$\ r[\ _m(\ ^*|\mathcal{D}_m)] = \ (\ ^*\ \ ^*) - \boldsymbol{\sigma}^T(\ ^*)\ \ ^{-1}(\boldsymbol{X}_m\ \boldsymbol{X}_m;\boldsymbol{\theta}_k)\boldsymbol{\sigma}(\ ^*)$$

where $\boldsymbol{\sigma}(\ ^*) = (\ (\ ^*\ \ _{m,1}) \cdots \ (\ ^*\ \ _{m,N_m}))^T$.

For the parameter learning of this hierarchical mixture model of Gaussian process, we introduce the latent variables z_m as follows:

$$\ _m(\boldsymbol{X}_m)|z_m = \ \sim \ \ (\ _k)$$

and assume the mixing form of Eq.(1).

2.2 The Priors and its Bayesian Form

In the hierarchical mixtures of Gaussian processes, we adopt the forms of the priors as given in [4], that is,

$$\ _k \sim \ (\frac{1}{2}\ \frac{1}{2})\ \ _k \sim \mathcal{LN}(-1\ 1^2)\ \ ^2_k \sim \mathcal{LN}(-3\ 3^2)\ \ = 1 \cdots$$

where $\ $ represents the inverse gamma distribution, and

$$(\ _1 \cdots \ _K) \sim \ \ r(1 \cdots 1).$$

In the Bayesian analysis, the priors are as important as the posteriors and likelihood function. Supposing that $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_K)$, $\boldsymbol{\pi} = (\ _1 \cdots \ _K)$, $\mathcal{D}$ is the set of training data, we then have

(i) The posterior of the parameters:

$$(\boldsymbol{\Theta}\ \boldsymbol{\pi}|\mathcal{D}) \propto \ (\boldsymbol{\Theta}\ \boldsymbol{\pi})\ \ (\mathcal{D}|\boldsymbol{\Theta}\ \boldsymbol{\pi});$$

(ii) The likelihood of the mixture model:

$$(\mathcal{D}|\boldsymbol{\Theta}\ \boldsymbol{\pi}) = \prod_{m=1}^{M}\sum_{k=1}^{K}\ _k\mathcal{N}(\boldsymbol{y}_m|\mathbf{0}\ \boldsymbol{\Sigma}_k(\boldsymbol{x}_m));$$

(iii) The prior distribution:

$$(\boldsymbol{\Theta}\ \boldsymbol{\pi}) = \ (\boldsymbol{\pi})\prod_{k=1}^{K}\ (\boldsymbol{\theta}_k).$$

In the next section, we will sample parameters from $\ (\boldsymbol{\Theta}\ \boldsymbol{\pi}|\mathcal{D})$.

3 The Markov Chain Monte Carlo Algorithm for the Hierarchical Mixtures of Gaussian Processes

In this section, we construct the Markov Chain Monte Carlo (MCMC) algorithm for the hierarchical mixtures of Gaussian processes in a similar way as the MCMC algorithm for Gaussian mixtures in [6]. Actually, the mathematical framework of our MCMC algorithm keeps the same, including a number of moves of the components or their parameters with time, but the components become Gaussian processes.

3.1 The Move Types

In [6], six types of moves were used. But here, since the adopted priors have no hyperparameters, there is no need for updating the hyperparameters. So, remaining five types of moves can be given as follows:

(a) $\boldsymbol{\pi} = (\ _1 \cdots \ _K)$;
(b) $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_K)$, where $\boldsymbol{\theta}_k = (\ _k \ _k \ ^2_k)$;
(c) $\boldsymbol{z} = (z_1 \cdots z_M)$;
(d) Split or merge;
(e) Remove empty components.

For clarity, we refer to one process of implementing these five moves completely as a sweep, being set as the basic step of the MCMC algorithm.

3.2 The Moves with the Component Number Fixed

The moves with fixed component number include the first three steps(a)(b)(c) in section 3.1. For this part, we adopt the algorithm in [4]:

For $\boldsymbol{\pi}$ and $\boldsymbol{z}$, we use Gibbs sampling:

(i) sample z_m from $(z_m = \ |\mathcal{D}\ \boldsymbol{\Theta}\ \boldsymbol{\pi}) \propto \ _k\ (\boldsymbol{y}_m|\boldsymbol{\theta}_k)\ \ = 1 \cdots \ = 1 \cdots$
(ii) sample $\boldsymbol{\pi}$ from $(\ _1 \cdots \ _K)|\boldsymbol{z} \sim \ r(1 + _1 \cdots 1 + _K)$

Here, $_k$ represents the element number in the set $\{\ = 1 \cdots \ |z_m = \ \}$ and $r(\cdot)$ represents Dirichlet distribution.

And for $\boldsymbol{\Theta}$, we sample it from its posterior:

$$(\boldsymbol{\Theta}|\mathcal{D}\ \boldsymbol{z}) \propto \prod_{k=1}^{K}\ (\boldsymbol{\theta}_k|\mathcal{D}_m\ \boldsymbol{z})$$

Then, the posterior of $\boldsymbol{\theta_k}\ \ = 1 \cdots$ are independent with each other and we can sample each $\boldsymbol{\theta_k}$ separately. Here we adopt Hybrid Monte Carlo or Hamiltonian Monte Carlo to sample $\boldsymbol{\theta}_k$. Actually, this sampling method used to be adopted to simulate a physical system where a puck moves up and down along with a smooth surface and the total energy remains constant. However, in our application, we restate the potential energy as $\mathcal{E}(\boldsymbol{\theta}_k) = -\log\ (\boldsymbol{\theta_k}|\mathcal{D}_m\ \boldsymbol{z})$ and the

kinetic energy as: $\mathcal{K}(\boldsymbol{\phi}_k) = \frac{1}{2}\sum \ {}^{2}_{k,i}$, $\ _{k,i} \sim \mathcal{N}(0\ 1)$ $\boldsymbol{\phi}_k = (\ _{k,1}\ \ _{k,2}\ \ _{k,3})$. Then the total energy of the hamiltonian system for $\boldsymbol{\theta}_k$ is $\mathcal{H}(\boldsymbol{\theta}_k\ \boldsymbol{\phi}_k) = \mathcal{E}(\boldsymbol{\theta}_k) + \mathcal{K}(\boldsymbol{\phi}_k)$.

For the convenience of calculation, we split $\mathcal{H}(\boldsymbol{\theta}_k\ \boldsymbol{\phi}_k)$ as follows:

$$\mathcal{H}(q,p) \ = \ \underbrace{-\frac{1}{2}\sum_{m\in\{z_m=k\}} \log p(\boldsymbol{y}_m|\boldsymbol{\theta}_k)}_{\frac{U_0}{2}} + \underbrace{[-\log p(\boldsymbol{\theta}_k) \ + \ \mathcal{K}(\boldsymbol{\phi}_k)]}_{U_1} \underbrace{-\frac{1}{2}\sum_{m\in\{z_m=k\}} \log p(\boldsymbol{y}_m|\boldsymbol{\theta}_k)}_{\frac{U_0}{2}}$$

thus the sample update step is:

(i) From the current state $(\boldsymbol{\theta}_k\ \boldsymbol{\phi}_k)$, we use a leapfrog step with step size $\ $ to calculate the new state $(\boldsymbol{\theta}^*_k\ \boldsymbol{\phi}^*_k)$:

$$\boldsymbol{\phi}^*_k = \boldsymbol{\phi}_k - \frac{}{2}\frac{^0}{\boldsymbol{\theta}_k}(\boldsymbol{\theta}_k)$$
$$\boldsymbol{\theta}^*_k = \boldsymbol{\theta}_k + \ \boldsymbol{\phi}^*_k$$
$$\boldsymbol{\phi}^*_k = \boldsymbol{\phi}^*_k - \frac{}{2}\frac{^1}{\boldsymbol{\theta}_k}(\boldsymbol{\theta}^*_k\ \boldsymbol{\phi}^*_k)$$
$$\boldsymbol{\phi}^*_k = \boldsymbol{\phi}^*_k - \frac{}{2}\frac{^0}{\boldsymbol{\theta}_k}(\boldsymbol{\theta}^*_k)$$

(ii) Then new current state is:

$$(\boldsymbol{\theta}^*_k\ \boldsymbol{\phi}^*_k) = \begin{cases} (\boldsymbol{\theta}^*_k\ \boldsymbol{\phi}^*_k) & r\ , = \min(1\ \exp(-\mathcal{H}(\boldsymbol{\theta}^*_k\ \boldsymbol{\phi}^*_k) + \mathcal{H}(\boldsymbol{\theta}_k\ \boldsymbol{\phi}_k))) \\ (\boldsymbol{\theta}_k\ -\boldsymbol{\phi}_k) & \ , r \end{cases}$$

(iii) Finally, generate $\ _{k,i} \sim \mathcal{N}(0\ 1)$, and update $\boldsymbol{\phi}_k$ as: $\boldsymbol{\phi}^*_k = \ \boldsymbol{\phi}^*_k + \sqrt{1 - \ ^2}\boldsymbol{\nu}_k$. In addition, according to the advise in [7], we set $\ = 0.5\ \ ^{-1/2}_m \ = 0.95$.

For $\ = 1\ \cdots\ $, by repeating (i)(ii)(iii) $\ = 20$ times, we finish one sweep of step(b). Note that $\ _k\ \ _k\ \ ^2_k \ 0$, through the analysis of the method of handling constraints in [8], we reject the current state when $\ _k\ \ _k\ \ ^2_k \ 0$

3.3 The Moves with the Component Number Changed

The moves with the component number changed include the last two steps(d)(e). For move(e), all the components whose $\ _k \ 1\%$ are considered to be empty components and we delete them directly.

The Detailed Balance Framework. For move(d), we construct a detailed balance framework for the move with the component number changed so that the covariance parameters of the new components can be related to the original ones. The key to success is that the first two moments remain almost constant. For convenience we shall denote the splitted component as *th component and the two new components as $\ _1\ \ _2$.

(I) Actually, for the zeroth moment of $\boldsymbol{y}_m$, by simple mathematics calculation we find it is $\sum_{k=1}^{K} \ _k$, thus keeping the zeroth moment constant means $\ _{k^*} = \ _{k_1} + \ _{k_2}$.

(II) In the case of the first moment, it is $\mathbf{0}$ since we assume that the Gaussian process is zero mean. Thus its first moment always keep constant.

(III) However, the second moment is somewhat complicated and it is $\sum_{k=1}^{K} \quad k \quad k$. Then keeping the second moment constant means $\quad k^* \quad k^* = \quad k_1 \quad k_1 + \quad k_2 \quad k_2$.

(i) For the diagonal entry, let $_ = \ $, then we get a balance formula:

$$\quad k^*(\ k^* + \ \overset{2}{k^*}) = \quad k_1(\ k_1 + \ \overset{2}{k_1}) + \quad k_2(\ k_2 + \ \overset{2}{k_2})$$

(ii) For the off diagonal entry, let $_ \neq \ $, we have:

$$\quad k^*(\ k^* \exp\left(-\frac{\ k^*(\ \ i - \ \ j)^2}{2}\right)) = \quad k_1(\ k_1 \exp\left(-\frac{\ k_1(\ \ i - \ \ j)^2}{2}\right)) + \quad k_2(\ k_2 \exp\left(-\frac{\ k_2(\ \ i - \ \ j)^2}{2}\right)) \quad (2)$$

For convenience, denote Eq.(2) as a function of $(\ \ i - \ \ j)^2$:

$$\ (\ ) := \quad k^*(\ k^* \exp\left(-\frac{\ k^*}{2}\right)) - \quad k_1(\ k_1 \exp\left(-\frac{\ k_1}{2}\right)) - \quad k_2(\ k_2 \exp\left(-\frac{\ k_2}{2}\right)) \equiv 0 \quad (3)$$

In fact, $\ (\ )$ is exponential decreasing about $\ $ and $\ $ is in $\{\ ^2 \ 2^2 \ ^2 \ \cdots \ \overset{2}{m} \ ^2\}$, thus we can conclude that when $\ $ gets the minimum $\ ^2$, $\ (\ )$ reaches the maximum. Therefore, in order to keep $\ (\ ) \approx 0$ we just need keep $\ (\ ^2) \approx 0$. Do Taylor's expansion of $\ (\ )$ at $\ = 0$ and denote the n-th term as $\ _n$ for convenience. What surprise us is that when $\ = \ ^2$ and if $\ $ is small enough, $\ _n$ will be monotonic decreasing, then $\ _0$ will be the largest term and $\ _1$ will be the second largest term. Therefore just let $\ _0 \ \ _1$ be zero, the second moment will keep almost constant.

Then, we have got our detailed balance framework:

$$\quad k^* = \quad k_1 + \quad k_2 \quad (4a)$$

$$\quad k^* \ \overset{2}{k^*} = \quad k_1 \ \overset{2}{k_1} + \quad k_2 \ \overset{2}{k_2} \quad (4b)$$

$$\quad k^* \ k^* = \quad k_1 \ k_1 + \quad k_2 \ k_2 \quad (4c)$$

$$\quad k^* \ k^* \ k^* = \quad k_1 \ k_1 \ k_1 + \quad k_2 \ k_2 \ k_2 \quad (4d)$$

The Merge and Split Formula. For merge move, we can use Eq.(4) to merge two components, so Eq.(4) is also our merge formula. For split move, reversible jump in [5] is needed. According to the theory above, 4 dimensions random variable $\boldsymbol{u} = (\ _1 \ _2 \ _3 \ _4)$ need to be generated to decide these new parameters: $\ _i \sim \boldsymbol{Beta}(2 \ 2)$, $\_ = 1 \ \cdots \ 4$. By combining these random parameters, the balance formula Eq.(4) and the reversible jump theory, we can write the split formula as:

$$\quad k_1 = \ _1 \quad k^* \qquad \quad k_2 = (1 - \ _1) \quad k^* \ _1 \in (0 \ 1) \quad (5a)$$

$$\ \overset{2}{k_1} = \ _2 \ \overset{2}{k^*} \frac{k^*}{k_1} \qquad \ \overset{2}{k_2} = (1 - \ _2) \ \overset{2}{k^*} \frac{k^*}{k_2} \ _2 \in (0 \ 1) \quad (5b)$$

$$\ k_1 = \ _3 \ k^* \frac{k^*}{k_1} \qquad \ k_2 = (1 - \ _3) \ ^{k^*}_0 \frac{k^*}{k_2} \ _3 \in (0 \ 1) \quad (5c)$$

$$\ k_1 = \frac{1 - \ _4}{\ _3} \ k^* \qquad \ k_2 = \frac{\ _4}{1 - \ _3} \ k^* \ _4 \in (0 \ 1) \quad (5d)$$

It is easy to validate that split formula Eq.(5) matches with merge formula Eq.(4).

The split and merge move is a Markov birth-death chain, and we set the split probability and merge probability as $_{,k}$ $_{-k} = 1 - _{,k}$ respectively, depending on k: $_{-1} = 0$ $_{,k_{\max}} = 0$ $_{,k} = _{-k} = 0.5$ $\forall$ $= 2 \cdots$ $_{\max} - 1$, where $_{\max}$ is the maximum component number that we set according to individual cases. Then based on the acceptance probability formula of reversible jump move in [5] the acceptance ratio for a split move is min(1) and a merge move is min(1 $^{-1}$) where

$$= \prod_{m=1}^{M} \frac{(\boldsymbol{Y}_m|\boldsymbol{\theta}_{k+1})}{(\boldsymbol{Y}_m|\boldsymbol{\theta}_k)} \times \frac{_{-k+1}}{_{,k}} \times \underbrace{\frac{(\boldsymbol{\theta}_{k+1})}{(\boldsymbol{\theta}_k)}}_{T_1} \times \underbrace{\frac{1}{\boldsymbol{Beta}(\boldsymbol{u}|\boldsymbol{\theta}_{k+1}\ \boldsymbol{\theta}_k)}}_{T_2} \times \underbrace{\left|\frac{\boldsymbol{\theta}_{k+1}}{(\boldsymbol{\theta}_k\ \boldsymbol{u})}\right|}_{T_3}$$

Here, * $_1$ $_2$ are chosen randomly from the components.

4 Experimental Results

Our experiments are conducted on a set of simulated data generated from a mixture of three Gaussian processes with $_1 = (1.0\ 0.2\ 0.0025)$, $_2 = (0.5\ 1.0\ 0.001)$, $_3 = (10\ 0.2\ 0.0005)$, respectively, each with 3 curves. The data points are with $= -4 : 0.08 : 4$, being equally spaced. We have two kinds of prediction, type I prediction: choosing half data randomly from each of the 9 curves as training data, the rest as test data; type II prediction: generating a new curve from the first GP, choosing half data randomly on this curve as known data, using this half and training parameters from the type I prediction to simulate the other unknown half data. In the later two subsections, the two type of prediction is used to verify the component number fixed algorithm and the component number varied algorithm for regression and model selection. For both of the two algorithms, we run them for 20000 sweep. Here we discard the first 10000 iterations and select one sample from each 200 iterations, in order to have approximately independent draws.

4.1 The Component Number Fixed

First, we fix $= 3$, and use only the moves with the component number fixed for training. The log-likelihood tends to stabilize after about 1700 iteration. Table (1) presents the predictions. We will compare it with the component number varied result in the next subsection.

4.2 The Component Number Varied

Then we use our component number varied method to train and predict the same curves as before. Fig.(1) is ($|\mathcal{D})$ $= 1 \cdots 6$, from which we can conclude

Table 1. RMSE and correlation coefficient(r) between true and predicted responses

Training data:half data on the first 9 curves		
model: fixed component number of GP mixture regression model		
Test data	RMSE	r
the first GP	0.2329	0.9485
the second GP	0.3612	0.8520
the third GP	0.2266	0.9024
the 10th curve	0.2037	0.7566
Training data:half data on the first 9 curves		
model: varied component number of GP mixture regression model		
Test data	RMSE	r
the first GP	0.0602	0.9900
the second GP	0.0253	0.9982
the third GP	0.0645	0.9867
the 10th curve	0.0493	0.9820

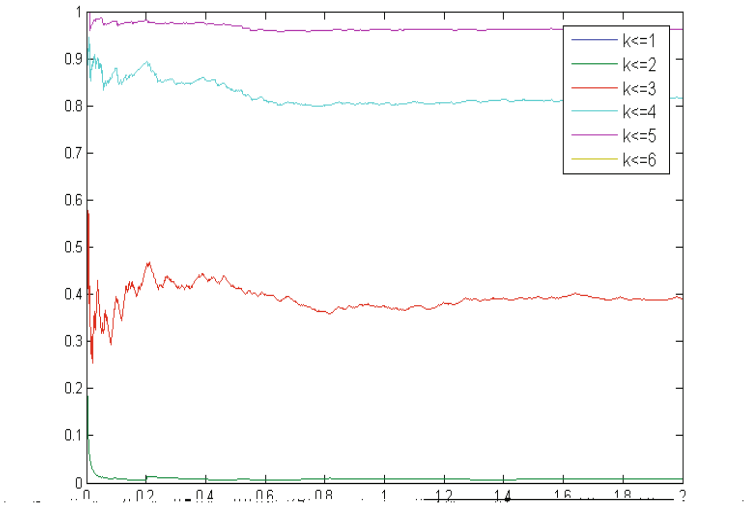

Fig. 1. $p(k < j|\mathcal{D}), j = 1, \cdots, 6$ for 40000 iterations

Table 2. posterior of k for Gaussian process mixture model

curve number	$p(k\|\mathcal{D})$	proportion (%) of moves accepted	
		split	merge
9	p(1)=0.0001 p(2) = 0.0563 p(3)=0.3811 p(4)=0.3677 p(5) = 0.1558 $\sum_{k\geq 6} p(k) = 0.0391$	14	2

that the algorithm has converged after about 10000 iteration. Additionally, we also present the posterior of for mixtures of Gaussian processes in Table (2). From this table, we can conclude this model favors $3-4$ components. In this

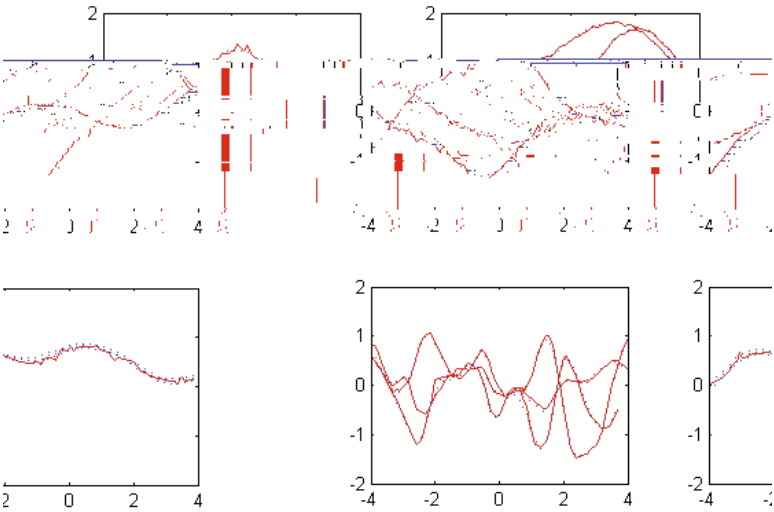

Fig. 2. predict curve with the component number varied method on 9 curves, the solid line represent the real value, and the dash line represents the predict line with 95% confidence interval

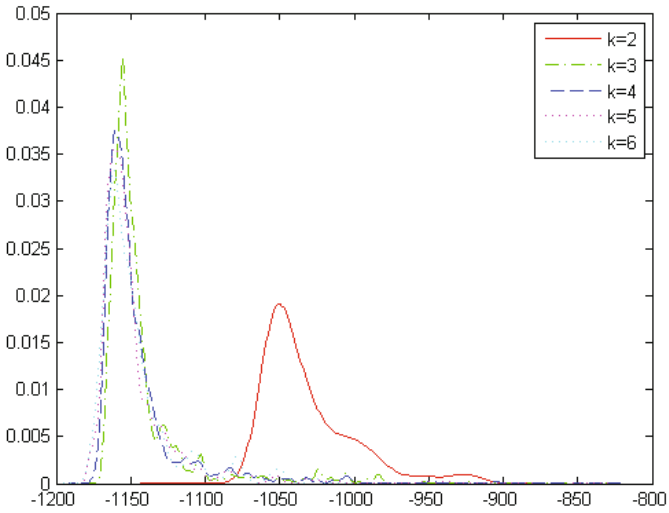

Fig. 3. the deviance of the kernel smooth density posterior of k

part, we initially add the birth and death moves in [6] to the algorithm in order to increase the opportunities for split and merge thus to speed up convergence of Markov chain, but finally it doesn't work, and we delete this type move.

The final predict results are presented in Fig.(2). We can see that predict curves almost overlap with the true curves. For type II prediction, the 95%

confidence interval is quite small. RMSE and correlation coefficient are in Table (1). The RMSE is about one tenth of that for the component number fixed case and the correlation coefficient ≥ 0.9800 which is also improved.

In addition, we analyze the deviance of the kernel smooth density of posterior of $-2 * \log(|\mathcal{D})$ in Fig.(3). Since the deviance of $= 2$ separate with $= 3$ and from $= 3$ to $= 6$ the deviances are overlapping together, we can conclude that $= 3$ can represent most information of train data.

Increasing the curve number to 30 and even to 90, we find that it does not improve the RMSE and correlation coefficient obviously. Therefore, this become a burden to waste time but of no use for improving algorithm accuracy and is unnecessary.

5 Conclusion

We have developed an automatic model selection algorithm for mixtures of Gaussian processes in the light of the reversible jump Markov chain Monte Carlo framework for Gaussian mixtures. The split and merge moves of Gaussian processes in the iteration keep the first two moments constant. In this way, the automatic model selection algorithm makes it possible to do the Bayesian analysis of both parameter estimation and model selection for the mixtures of Gaussian processes. Moreover, it is demonstrated that this developed algorithm is feasible and outperform the the hybrid MCMC algorithm.

Acknowledgments. This work was supported by the National Science Foundation of China under Grant 61171138.

References

1. Rasmussen, C.E., Williams, C.K.I.: Gaussian process for machine learning. MIT Press, Cambridge (2006)
2. Tresp, V.: Mixtures of Gaussian processes. In: Proc. of the Conf. on Neural Information Processing Systems (NIPS), pp. 654–660 (2000)
3. Shi, J.Q., Murray-Smith, R., Titterington, D.M.: Bayesian regression and classification using mixtures of Gaussian processes. International Journal of Adaptive Control and Signal Processing 17(2), 149–161 (2003)
4. Shi, J.Q., Murray-Smith, R., Titterington, D.M.: Hierarchical Gaussian process mixtures for regression. Statistics and Computing 15(1), 31–41 (2005)
5. Green, P.J.: Reversible jump markov chain monte carlo computation and bayesian model determination. Biometrika 82(4), 711–732 (1995)
6. Richardson, S., Green, P.J.: On Bayesian Analysis of Mixtures with an Unknown Number of Components (with discussion). Journal of the Royal Statistical Society: Series B (Statistical Methodology) 59(4), 731–792 (1997)
7. Rasmussen, C.E.: Evaluation of Gaussian processes and other methods for non-linear regression. Diss. University of Toronto (1996)
8. Neal, R.M.: MCMC using Hamiltonian dynamics. In: Handbook of Markov Chain Monte Carlo (2011)