An Efficient EM Approach to Parameter Learning of the Mixture of Gaussian Processes

Yan Yang and Jinwen Ma*

Department of Information Science,
School of Mathematical Sciences & LMAM Peking University,
Beijing, 100871, P.R. China

Abstract. The mixture of Gaussian processes (MGP) is an important probabilistic model which is often applied to the regression and classification of temporal data. But the existing EM algorithms for its parameter learning encounters a hard difficulty on how to compute the expectations of those assignment variables (as the hidden ones). In this paper, we utilize the leave-one-out cross-validation probability decomposition for the conditional probability and develop an efficient EM algorithm for the MGP model in which the expectations of the assignment variables can be solved directly in the E-step. In the M-step, a conjugate gradient method under a standard Wolfe-Powell line search is implemented to learn the parameters. Furthermore, the proposed EM algorithm can be carried out in a hard cutting way such that each data point is assigned to the GP expert with the highest posterior in the E-step and then the parameters of each GP expert can be learned with these assigned data points in the M-step. Therefore, it has a potential advantage of handling large datasets in comparison with those soft cutting methods. The exper-

by a gating network. With the help of the divide-and-conquer strategy, the MGP model is more flexible for modeling a temporal dataset than a single Gaussian process. Moreover, Gaussian process has been shown to have a good performance on regression and classification. However, just as many other powerful tools, Gaussian process is not so perfect and has two main limitations. First, a Gaussian process has a stationary covariance function and this characteristic cannot be adapted in the cases of temporal datasets which have varying noises in different times. Second, the computational cost of the parameter learning or inference is

* Corresponding author, jwma@math.pku.edu.cn

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 165–174, 2011.

very high since it is involved in the computation of the inversion of an $n \times n$ matrix where n is the number of the training dataset.

Since the MGP model was firstly investigated by Tresp [11], there have appeared some variants of the MGP model and the corresponding learning methods have been established ([3-4], [12], etc.). For clarity, we summarize all these investigations from two aspects: gating network and inference method. In fact, there are three kinds of gating networks in the literature. Firstly, as the MGP model is inspired by the ME model, the gating network of the ME model can be straightforward inherited [7, 9, 11]. The second kind of gating network is just a set of mixing coefficients which are assumed to follow a Dirichlet distribution [6] or be generated from a Dirichlet process [4] (in this case, the finite mixture model can be generalized to an infinite one). The third kind of gating network is based on the distribution of the data points from the input space. In this situation, data points from one GP expert space are assumed to be subject to a Gaussian distribution [3], or a Gaussian Mixture distribution [12].

With the diversity of gating networks, there have developed two main inference methods: the Bayesian inference method and the non-Bayesian parameter estimation method. By the Bayesian inference method, all the parameters are assumed to have some prior distributions and certain sophisticated techniques like the Markov Chain Monte Carlo methods are used for the parameter learning or estimation [3-4], [12]. On the other hand, since the well-known EM algorithm has been successfully implemented to learn the ME model [1-2], several implementations of the EM algorithm have been proposed to learn the parameters of the MGP model (e.g., [7], [9], [11]). However, since the outputs of the MGP model are not independent as those of the ME model, it becomes a very difficult problem to compute the posterior probability that a data point belongs to each GP expert. Actually, the computation schemes of the posterior probabilities in the existing EM algorithms are heuristic, in lack of theoretical proofs, and often lead to a low efficiency.

In this paper, in order to solve this difficult problem more efficiently, we utilize the leave-one-out cross-validation probability decomposition for these conditional probabilities and develop an efficient EM algorithm for the MGP model in which the expectations of the assignment variables can be computed directly. In fact, the leave-one-out cross-validation probability decomposition was already used for the parameter learning in the single GP model [5], [10], but it has not been used for the parameter learning of the MGP model. Here, as the conditional probability of the output with respect to the input and the parameters is expressed by the leave-one-out cross-validation probability decomposition, we can get a novel expression of the posterior probability that each data point belongs to a GP expert in the E-step. In the M-step, we implement a conjugate gradient method under a standard Wolfe-Powell line search to maximize the log likelihood with the gradients being computed via the expressions given by Sundararajan et al. [10].

cause a great computation cost in dealing with a large dataset. To get rid of this difficulty, we further modify the proposed EM algorithm in a hard cutting way by assigning every data point to the GP expert with the highest posterior in the E-step. Then, in the M-step, only these assigned data points are used to learn the parameters of each expert. Therefore, the modified EM algorithm is more adapted to deal with the learning problem of a large dataset. To demonstrate the proposed algorithms in this situation, we conduct experiments on the motorcycle dataset.

The remainder of this paper is organized as follows. In Section 2, we introduce the MGP model and the leave-one-out cross-validation probability decomposition. The new EM algorithm is derived and investigated in Section 3, with the experimental results being illustrated in Section 4. In Section 5, we make a brief conclusion.

2 MGP and Leave-One-Out Cross-Validation Probability Decomposition

We begin with a brief introduction to the Gaussian Process according to the work by Rasmussen and Williams [5]. Give a set of training data $X = [x_1^T, \cdots, x_n^T]^T$ as inputs and $Y = [y_1^T, \cdots, y_n^T]^T$ as the corresponding outputs, where n is the number of the training data. This dataset is said to follow a Gaussian Process if $Y \sim \mathcal{N}(m(X), K_y(X, X))$, where $m(X)$ is a prior defined mean function and $K_y(X, X)$ is a covariance matrix function with its element $K_y(x_p, x_q)$ being a kernel function. For simplicity, we assume that the mean function $m(X)$ is zero. There are some varying forms for the covariance function and here we use the common one named the *squared exponential* (SE) covariance function as follows:

$$K_y(x_p, x_q) = l^2 \exp\{-\frac{\sigma_f^2}{2}||x_p - x_q||^2\} + \delta_{pq}\sigma_n^2, \tag{1}$$

where l, σ_f and σ_n are nonzero real values. $\delta_{pq} = 1$ if $p = q$; otherwise, $\delta_{pq} = 0$.

The MGP model, as an extension of mixture of experts (ME) architecture, is a combination of several single Gaussian processes by a gating network $g(x|\varphi)$, where φ denotes the set of all the parameters in the gating network. The gating network aims to divide the input space into regions for specific Gaussian processes making predictions. As described in [3], we assume that data points in the input space are i.i.d. and those from the same GP expert are Gaussian distributed.

Suppose that there is a training dataset $\{Y, X\} = \{y_t, x_t\}_{t=1}^N$ being generated from a mixture of Gaussian processes containing M single components. The covariance matrix K_j of the j-th GP component is specified by the parameters $\theta_j = \{l_j, \sigma_{fj}, \sigma_{nj}\}$ and each Gaussian component in the input space (i.e., $\mathbb{R}^d$) is specified by the parameters $\varphi_j = \{\nu_j, \Sigma_j\}$. Let Y_{-t} and X_{-t} be the corresponding datasets leaving out y_t and x_t, respectively. The leave-one-out cross-validation

probability decomposition can be given by

$$p(Y, X, \Theta) = \prod_{t=1}^{N} \sum_{j=1}^{M} \alpha_{tj} p(y_t|x_t, Y_{-t}, X_{-t}, A, \theta_j) p(x_t|\varphi_j), \tag{2}$$

where $A = \{\alpha_{tj}\}$, α_{tj} is the probability that (y_t, x_t) belongs to the j-th component, under the constraint that $\sum_{j=1}^{M} \alpha_{tj} = 1$. In our consideration, the gating network is set by $g(x|\varphi) = [p(x|\varphi_1), \cdots, p(x|\varphi_M)]^T$. Specifically, we have

$$p(x_t|\varphi_j) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\{-1/2(x_t - \nu_j)^T \Sigma_j^{-1}(x_t - \nu_j)\}. \tag{3}$$

For any pair of y_p and y_q in given Y_{-t}, the covariance of them can be written as $K(x_p, x_q) = Cov(y_p, y_q) = \sum_{i=1}^{M} \alpha_{pi}\alpha_{qi} K_i(x_p, x_q)$, where K_i is the covariance function of the i-th GP component. Under the assumption that y_t belongs to the j-th component, the covariance of y_t and any y_p in Y_{-t} is $Cov(y_p, y_t) = \alpha_{pj} K_j(x_p, x_t)$. Therefore, we have

$$\begin{bmatrix} Y_{-t} \\ y_t \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X_{-t}, X_{-t}) & \beta_{tj} \\ \beta_{tj}^T & K_j(x_t, x_t) \end{bmatrix}\right)$$

where $\beta_{tj}(x_p, x_t) = \alpha_{pj} K_j(x_p, x_t)$. Hence, we further get

$$p(y_t|x_t, Y_{-t}, X_{-t}, A, \theta_j) \sim \mathcal{N}(\mu_{tj}, \sigma_{tj}^2) \tag{4}$$

where

$$\mu_{tj} = \beta_{tj}^T K(X_{-t}, X_{-t})^{-1} Y_{-t}, \tag{5}$$

$$\sigma_{tj}^2 = K_j(x_t, x_t) - \beta_{tj}^T K(X_{-t}, X_{-t})^{-1} \beta_{tj}. \tag{6}$$

Until now we have specified the leave-one-out cross-validation probability decomposition (2), and in the following analysis we will try to maximize it under an EM framework.

3 Proposed EM Algorithm for MGP

Let $\{Y, X\} = \{y_t, x_t\}_{t=1}^{N}$ be a dataset drawn from a MGP model which contains M components, where N is the number of training data points. In order to carry out the EM algorithm for MGP, we first consider a set of binary variables $Z = \{z_{tj}\}$ such that $z_{tj} = 1$, if (y_t, x_t) is drawn from the j-th GP expert; otherwise, $z_{tj} = 0$. Obviously, $\sum_{j=1}^{M} z_{tj} = 1$. Let Y^j and X^j denote the output and input data points of the j-th GP, and Y_{-t}^j and X_{-t}^j be the corresponding datasets leaving out y_t and x_t, respectively.

Suppose that all the values of the binary variables $Z = \{z_{tj}\}$ are known, the joint probability eqn. (2) can be written as

$$p(Y, X|\Theta, \Phi) = \prod_{t=1}^{N} \prod_{j=1}^{M} (\alpha_{tj}(p(x_t|\varphi_j)))^{z_{tj}} p(y_t|x_t, X_{-t}, Y_{-t}, Z, \theta_j), \tag{7}$$

where $p(y_t|x_t, X_{-t}, Y_{-t}, Z, \theta_j)$ is obtained by replacing α_{ti} with z_{tj} in eqn. (4). We then get the log likelihood function as follows:

$$l_0(\Theta, \Phi; Y, X, Z) = \sum_{t=1}^{N}(\log p(y_t|x_t, X_{-t}, Y_{-t}, Z, \Theta) + \sum_{j=1}^{M} z_{tj} \log \alpha_{tj} p(x_t|\varphi_j)). \quad (8)$$

In this situation, the missing data are the hidden variables Z, the observed data are $\{Y, X\}$ and l_0 is the log likelihood of the complete data which we aim to maximize. We further define a so-called Q function as the expectation of the log likelihood w.r.t. the missing data Z:

$$Q(\Theta, \Phi|\Theta^{(k)}, \Phi^{(k)}) = E_Z\{l_0(\Theta^{(k)}, \Phi^{(k)}; X, Y, Z)\}, \quad (9)$$

where $\Theta = \{\theta_j\}$, $\Phi = \{\varphi_j\}$. In the EM framework, we actually do not maximize
)3.7(TJ E.6(.1(lik(n)-324.31(38 0 0 e)] 05946 o7.8(or4)-3030 0 xs)-25 1 Tf 9.iz-4.6(ca7io)-

and variance σ_t^2 can be expressed as $\mu_t = y_t - [K^{-1}Y]_t/[K^{-1}]_{tt}$, $\sigma_t^2 = 1/[K^{-1}]_{tt}$, and the notations $[\cdot]_t$, $[\cdot]_{tt}$ stand for the tth element of the specified vector and the tth diagonal element of the specified matrix, respectively. K is the covariance function defined by $K(x_p, x_q) = \sum_{j=1}^{M} h_j(p)h_j(q)K_j(x_p, x_q)$. Its gradient can be given by

$$\frac{\partial Q_1}{\partial \theta_j} = \sum_{t=1}^{N} \frac{\alpha_t [Z_j \alpha]_t}{[K^{-1}]_{tt}} - \frac{\alpha_t^2 [Z_j K^{-1}]_{tt}}{2[K^{-1}]_{tt}^2} - \frac{[Z_j K^{-1}]_{tt}}{2[K^{-1}]_{tt}}, \tag{14}$$

where $\alpha = K^{-1}Y$, $Z_j = K^{-1}\partial K/\partial\theta_j$ and $\partial K/\partial\theta_j = h_j(p)h_j(q)\partial K_j/\partial\theta_j$. According to the definition of the covariance function eqn. (1), we have

$$\frac{\partial K_j(p,q)}{\partial l_j} = 2l_j \exp\{-\frac{\sigma_{fj}^2}{2}||x_p - x_q||^2\}, \qquad \frac{\partial K_j(p,q)}{\partial \sigma_{nj}} = 2\delta_{pq}\sigma_{nj},$$

$$\frac{\partial K_j(p,q)}{\partial \sigma_{fj}} = -\sigma_{fj}||x_p - x_q||^2 l_j^2 \exp\{-\frac{\sigma_{fj}^2}{2}||x_p - x_q||^2\}.$$

With the above preparations, we now give our new EM algorithm for MGP as follows.

1. Initialize the parameters $\{\alpha_{tj}\}$, $\{\theta_j\}$, $\{\varphi_j\}$.
2. Calculate the posteriors according to eqn. (10).
3. Calculate $\hat{\alpha}_{tj}$, $\hat{\nu}_j$, $\hat{\Sigma}_j$ according to eqn. (12, 13). Calculate θ_j by maximizing Q_1 using a conjugate gradient method under a standard Wolfe-Powell line search.
4. Repeat step 2-4, until convergence.

In the E-step, we compute the posteriors by eqn. (10). In the M-step, we estimate the parameters α_{tj}, $\hat{\nu}_j$, $\hat{\Sigma}_j$ by eqn. (12 & 13) and θ_j by implementing the conjugate gradient method with the help of eqn. (14). Repeat the two steps until convergence.

As an disadvantage of the non-Bayesian methods [11] in comparison with the Bayesian methods [3-4],[12], the computation complexity problem is so knotty that they require the inverse of an $n \times n$ matrix for every GP expert, where n is the number of the training data points. In order to overcome this complexity problem, we can modify our proposed EM algorithm in a hard cutting mode. That is, in the E-step, after getting all the posteriors, we assign each data point to the GP expert with the largest posterior. In the M-step, only the assigned data points are used for learning each GP expert. In such a way, the computation complexity of the modified hard cutting EM algorithm is reduced considerably.

4 Experimental Results

To test the performance of our proposed EM algorithm for MGP, we conduct some experiments on an artificial toy dataset given in [3] and the motorcycle dataset given in [8]. The artificial toy dataset consists of four continuous functions which have different levels of noise. The four continuous functions are:

$f_1(a_1) = 0.25a_1^2 - 40 + \sqrt{7}n_t$, $f_2(a_2) = -0.0625(a_2 - 18)^2 + 0.5a_2 + 20 + \sqrt{7}n_t$, $f_3(a_3) = 0.008(a_3 - 60)^3 - 70 + \sqrt{4}n_t$, $f_4(a_4) = -\sin(a_4) - 6 + \sqrt{2}n_t$, where $a_1 \in (0, 15)$, $a_2 \in (35, 60)$, $a_3 \in (45, 80)$, $a_4 \in (80, 100)$ and $n_t \sim \mathcal{N}(0, 1)$ that denotes a standard Gaussian distribution (with zero mean and variance 1). We generate 200 samples (50 samples for each function) from this toy model. We apply a mixture of four Gaussian Processes to model this dataset and implement the EM algorithm to learn the parameters of the mixture. The experimental results are shown in Figure 1. The noise values of each expert learned by our proposed EM algorithm are very close to the true ones: 7.04, 6.69, 3.98, 1.59. In the input space the centroids of the experts are 7.28, 46.65, 64.22, 90.90.

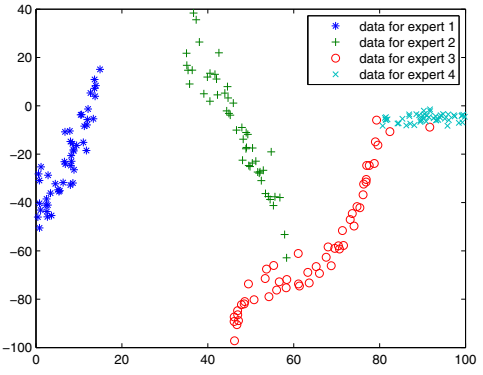
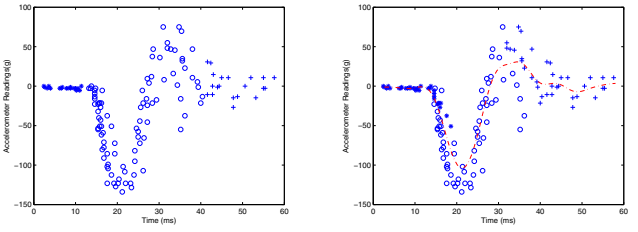


Fig. 1. Experimental results of the proposed EM algorithm on the toy dataset. The notations '*', '+', 'o', 'x' represent samples from four expert.

The motorcycle dataset consists of 133 observations of accelerometer readings taken through time. These observations belong to three strata and we present them in terms of intervals along the time axis: [2.4, 11.4], (11.4 40.4] and (40.4, 57.6]. In the left plot of Figure 2, we illustrate the dataset and denote those belonging to the same stratum by notations 'o', '*' and '+', respectively.

In this case, we set the number of GP experts as 3, and then implement the proposed EM algorithm for MGP on the dataset. For convenience, we begin to initialize the posteriors rather than the parameters, as in the MGP model, the prediction task is impossible to be done only with the parameters. In the M-step, the conjugate gradient method under a standard Wolfe-Powell line search is applied to estimating the parameters in the GP experts. In this situation, the conjugate gradient method is considered to get a maximum solution when the absolute values of the derivatives w.r.t all the parameters are less than 0.01. We repeat the E-step and the M-step until convergence. In this particular case we stop the algorithm as long as the average norm of the difference of the parameters in the latest two iterations is less than 0.1.

We list the estimated parameters learned by the proposed EM algorithm in Table 1. It shows clearly that three GP experts divide the input space and model the corresponding data points, respectively. They have different degree of noises



(a) (b)

Fig. 2. (a) Three strata of the motorcycle dataset denoted by 'o', '*' and '+'. (b) Clustering result by the proposed EM algorithm for MGP. Three clusters are denoted by the notations 'o', '*' and '+'. We illustrate the predictive medians by the dash-dot line '-·', with 100 samples at each of the 84 equispaced locations according to the posterior distribution.

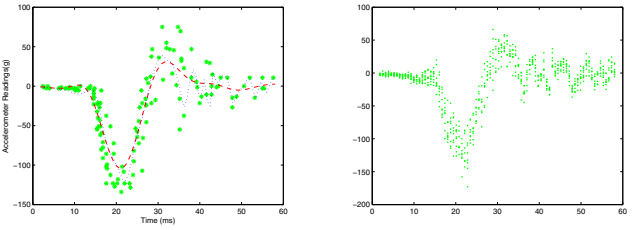
Table 1. The parameters of the MGP model on the motorcycle dataset estimated by the EM algorithm

	l	σ	σ	$\hat{\mu}$	$\hat{\sigma}^2$
GP expert 1	0.937	0.139	1.106	8.916	15.653
GP expert 2	30.902	0.319	24.212	23.060	53.370
GP expert 3	13.719	1.218	7.833	42.470	73.352

(σ_n^2) according to the varying intervals. The GP expert 1 mainly model the data points at the beginning of the dataset where the data points seem flat. Therefore, the noise in this area learned by the EM algorithm is much smaller than those in the other areas.

To show the flexibility of our leave-one-out cross-validation MGP model, we illustrate the predictive median of the predictive distribution using a dotted line on the left plot in Figure 3. Meanwhile, we use a dashed line to represent the median of the predictive distribution of a single stationary covariance GP model. According to the difference between the two lines, we can observe that the dotted line performs better especially in the intervals where time < 12ms and > 40ms. As speculated in [3], our model also performs well by not inferring an "flat" GP expert [4] at the beginning of the dataset where time < 11ms. In the interval where time > 45ms, the data points are not as dense as those around at 30ms. As compared with the prediction in this interval in [3], the predictive mean of our model almost passes through every data point. The experimental result shows that our leave-one-out cross-validation model may be in more agreement with the idea of the Gaussian process regression that the more closer in the input space the more closer in the output space. The right plot in Figure 3 shows a set of samples drawn from the predictive distribution. We select 84 equispaced locations along the time axis and draw 100 samples at each location.

We further apply the modified hard cutting algorithm on the motorcycle dataset and illustrate the result in the right plot in Figure 2. Three clusters



(a)

Acknowledgments

This work was supported by the Natural Science Foundation of China for grant 60771061.

References

1. Jordan, M.I., Jacobs, R.A.: Hierarchies mixtures of experts and the EM algorithm. Neural Computation 6, 181–214 (1994)
2. Jordan, M.I., Xu, L.: Convergence Results for the EM Approach to Mixtures of Experts Architectures. Neural Computation 8(9), 1409–1431 (1995)
3. Meeds, E., Osindero, S.: An Alternative Infinite Mixture of Gaussian Process Experts. Advances in Neural Information Processing System 18, 883–890 (2006)
4. Rasmussen, C.E., Ghahramani, Z.: Infinite Mixtures of Gaussian Process Experts. Advances in Neural Information Processing System 14, 881–888 (2002)
5. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
6. 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)
7. Shi, J.Q., Wang, B.: Curve Prediction and Clustering with Mixtures of Gaussian Process Functional Regression Models. Statistics and Computing 18, 267–283 (2008)
8. Silverman, B.W.: Some aspects of the spline smoothing approach to non-parametric regression curve fitting. Journal of the Royal Statistical Society. Series B 47(1), 1–52 (1985)
9. Stachniss, C., Plagemann, C., Lilienthal, A., Burgard, W.: Gas distribution modeling using sparse Gaussian process mixture models. In: Proc. of Robotics: Science and Systems (RSS), Zurich, Switzerland (2008)
10. Sundararajan, S., Keerthi, S.S.: Predictive Approaches for Choosing Hyperparameters in Gaussian Processes. Neural Computation 13(5), 1103–1118 (2001)
11. Tresp, V.: Mixtures of Gaussian Processes. Advances in Neural Information Processing System 13, 654–660 (2001)
12. Yuan, C., Neubauer, C.: Variational Mixture of Gaussian Process Experts. Advances in Neural Information Processing System 21, 1897–1904 (2008)