Pattern Recognition 127 (2022) 108592

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Bayesian mixture of gaussian processes for data association problem

Younghwan Jeon, Ganguk Hwang*

Department of Mathematical Sciences, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Republic of Korea

ARTICLE INFO

Article history:
Received 17 April 2021
Revised 13 January 2022
Accepted 16 February 2022
Available online 19 February 2022

Keywords:
Gaussian processes
Bayesian models
Variational inference
Expectation maximization

ABSTRACT

We address the data association problem and propose a Bayesian approach based on a mixture of Gaussian Processes (GPs) having two key components, the assignment probabilities and the GPs. In the proposed approach, the two key components are simultaneously updated according to observations through an efficient Expectation-Maximization (EM) algorithm that we develop. The proposed approach is thus more adaptive to the observations than the existing approaches for data association. To validate the performance of the proposed approach, we provide experimental results with real data sets as well as two synthetic data sets. We also provide a theoretical analysis to show the effectiveness of the Bayesian update.

1. Introduction

A Gaussian process (GP) [1] is a probabilistic model that is known to capture the latent properties of real-world data well and is widely used as a tool for inferring the latent generative functions in various machine learning domains such as metric learning [2] for searching a better feature map, topic modelling [3], and crowdsourcing data annotation [4].

When the observations are generated from several unknown sources, it is interesting to find the unknown sources which are usually modeled by latent functions and to label the observations according to the latent functions, i.e., sources. This sort of problem is usually referred to as data association [5] and GP plays an important role in addressing it. One recent approach is the overlapping mixture of Gaussian processes (OMGP) [6], which uses independent GPs and determines the association of observations to GPs by introducing the assignment probabilities. The assignment probabilities and GPs are updated via functional derivatives and expectation maximization (EM). However, as the assignment probabilities are considered as hyper-parameters and updated jointly with the other hyper-parameters in the model, a complex hyper-parameter space should be considered, which may lead to poor data association, as shown later. Moreover, the model requires additional assumptions to estimate the assignment probabilities for prediction.

To resolve the drawbacks of the OMGP model, we propose the Bayesian Mixture of Gaussian Processes (BMGP) which uses a mixture of GPs to directly estimate both the latent functions of observations and the mixture weight of GPs. By estimating the mixture weight, it can capture the impact of each latent function on the observations. The main contributions of our model, compared with the OMGP model, are summarized below.

- The BMGP model uses a newly developed EM algorithm. Unlike the OMGP model, the assignment probabilities in the BMGP model are considered as variational parameters, not the model hyper-parameters. They are updated only in the E-step by *iterating* the functional derivatives for GP update, as in Hensman et al. [7], and the natural gradient method. This makes the hyper-parameter space of the BMGP model become smaller than that of the OMGP model, leading to precise data association.
- We derive two properties of the BMGP model that make it more useful than the OMGP model. First, we derive a theoretical lower bound for the difference between the covariance matrices of the prior GPs and the posterior GPs to estimate the effectiveness of our EM algorithm. Second, we estimate the assignment probabilities at a test point in the BMGP model, which is impossible in the OMGP model.

The remainder of this paper is organized as follows. In Section 2, we review the GP and OMGP models which are two baseline models for the BMGP model. In Section 3, we propose the BMGP model and provide our learning scheme based on the EM algorithm to update the GP priors and the assignment probabilities. We also explain how to use the BMGP model in prediction and derive a lower bound that shows the effectiveness of the update in the BMGP model. In Section 4, we review previous related works and explain their differences from our work and their limitations. In Section 5, we provide experimental results to show that

* Corresponding author.
E-mail address: guhwang@kaist.edu (G. Hwang).

https://doi.org/10.1016/j.patcog.2022.108592
0031-3203/

the BMGP model works well on various synthetic and real data sets.

2. Background and notation

We start with briefly reviewing the traditional approaches for modelling with GPs. We then explain an existing model, the overlapping mixtures of Gaussian processes (OMGP), which is a starting point of our research.

2.1. GP modelling

Let $X = \{\mathbf{x}_n\}_{n=1}^N$ and $Y = \{y_n\}_{n=1}^N$ be a set of input points and outputs with $\mathbf{x}_n \in \mathbb{R}^Q$ and $y_n \in \mathbb{R}$. When f is assumed to be a Gaussian process, i.e, $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, it is called a GP modelling. Here, $m : \mathbb{R}^Q \to \mathbb{R}$ is the *mean function*. In this paper, we assume that $m \equiv 0$. $k : \mathbb{R}^Q \times \mathbb{R}^Q \to \mathbb{R}$ is the *kernel function*, which models the covariance between two function values.

GP regression is a model that assumes that each output y_n is generated from $f(\mathbf{x}_n)$ with an independent Gaussian noise, i.e, $y_n = f(\mathbf{x}_n) + \epsilon_n$, $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$. In this setting, $Y \sim \mathcal{N}(\mathbf{0}, K + \sigma^2 I_N)$ where $K = [k(\mathbf{x}_i, \mathbf{x}_j)]_{1 \le i,j \le n}$ is the Gram matrix of the kernel function k with respect to the input points X. The kernel function determines the properties of the latent function f such as periodicity, smoothness, etc. One of the most well known examples is the Automatic Relevance Determination (ARD) squared exponential function which forces f to be infinitely smooth: $k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\big(-\frac{1}{2}\sum_{q=1}^Q \frac{(x_q - x'_q)^2}{\ell_q^2}\big)$. Each kernel function has kernel hyper-parameters such as $(\sigma_f^2, \{\ell_q^2\}_{q=1}^Q)$ in the ARD squared exponential kernel. For simplicity, we refer to all kernel hyper-parameters as $\boldsymbol{\theta}$. Such kernel hyper-parameters are optimized by maximizing the following log likelihood function $\log P(Y|\boldsymbol{\theta}, \sigma^2) = -\frac{1}{2}Y^\top (K + \sigma^2 I_N)^{-1} Y - \frac{1}{2}|K + \sigma^2 I_N| - \frac{N}{2}\log(2\pi)$.

From now on, we assume a model that has M different GP latent functions $\{g^{(m)} \sim \mathcal{GP}(\mathbf{0}, k_m)\}_{m=1}^M$ and each output is generated by evaluating one of M latent functions at the corresponding input with an independent Gaussian noise having variance σ^2. Similarly, let K_m be the Gram matrix of the kernel function k_m with respect to X. The association of each output with latent functions is determined by a 1-of-M encoded indicator variable $\mathbf{z}_n = (z_{n1}, \ldots, z_{nM})^\top$, where z_{nm} is binary and $\sum_{m=1}^M z_{nm} = 1$ for all n.

Next, for a multi-dimensional output $\{\mathbf{y}_n\}_{n=1}^N$ where $\mathbf{y}_n \in \mathbb{R}^D$, we denote by Y the output matrix, which is defined by stacking all $\{\mathbf{y}_n\}_{n=1}^N$. In this case, D latent functions $\{g_d^{(m)}\}_{m,d=1}^{M,D}$ are used for modelling. Here, we assume that $\{g_d^{(m)}\}_{d=1}^D$ are independent and have the same distribution as the mth GP $g^{(m)}$. Let $Y_{(d)} = (Y_{1d}, \ldots, Y_{Nd})^\top$ be the dth column of Y, $g_d^{(m)}(X) = (g_d^{(m)}(\mathbf{x}_1), \ldots, g_d^{(m)}(\mathbf{x}_N))$, $\mathbf{g}^{(m)} = (g_1^{(m)}, \ldots, g_D^{(m)})$ and $\mathbf{Z} = \{\mathbf{z}_n\}_{n=1}^N$. We will denote all the GPs as $\mathbf{G} = \{\mathbf{g}^{(m)}\}_{m=1}^M$.

2.2. Overlapping mixtures of Gaussian processes

The OMGP model uses a mixture of GPs as latent functions [6]. Under the assumptions in the previous section, the likelihood of the OMGP model is

$$P(Y \mid \mathbf{Z}, \mathbf{G}, X) = \prod_{n=1,m=1,d=1}^{N,M,D} \mathcal{N}(Y_{nd} | g_d^{(m)}(\mathbf{x}_n), \sigma^2)^{z_{nm}} \tag{1}$$

with the following independent priors on the latent variables.

$$P(\mathbf{Z}) = \prod_{n=1,m=1}^{N,M} \pi_{nm}^{z_{nm}}, P(\mathbf{G}|X) = \prod_{m=1}^{M}\prod_{d=1}^{D} \mathcal{N}(g_d^{(m)}(X)|\mathbf{0}, K_m).$$

Here, for each $n = 1, \ldots, M$, $\{\pi_{nm}\}_{m=1,\ldots,M}$ are the prior probabilities of indicator $\mathbf{z}_n$ (i.e, $P(z_{nm} = 1) = \pi_{nm}$, $\pi_{nm} \ge 0$ and $\sum_{m=1}^M \pi_{nm} = 1$). In this model, $\{\pi_{nm}\}_{n,m}$ are also considered as the hyper-parameters of the model.

As the posterior distribution $P(\mathbf{Z}, \mathbf{G}|X, Y)$ is intractable, it is approximated by using variational inference, i.e, $P(\mathbf{Z}, \mathbf{G}|X, Y) \simeq q(\mathbf{Z}, \mathbf{G})$, where $q(\mathbf{Z}, \mathbf{G})$ optimizes the following lower bound:

$$\mathcal{L}(q(\mathbf{Z}, \mathbf{G})) := \int q(\mathbf{Z}, \mathbf{G}) \log \frac{P(Y, \mathbf{Z}, \mathbf{G}|X)}{q(\mathbf{Z}, \mathbf{G})} d\mathbf{Z} d\mathbf{G} \tag{2}$$

over a collection of distributions $q(\mathbf{Z}, \mathbf{G}) \in \mathcal{Q}$, satisfying

1. $q(\mathbf{Z}, \mathbf{G}) = q(\mathbf{Z})q(\mathbf{G})$,
2. $q(\mathbf{Z}) = \prod_{n=1}^N q(\mathbf{z}_n)$ and $\{q(\mathbf{z}_n)\}_{n=1}^N$ are independent categorical distributions with M categories, and
3. $q(\mathbf{G}) = \prod_{d=1}^D \prod_{m=1}^M \mathcal{N}(g_d^{(m)}(X)|\boldsymbol{\mu}_d^{(m)}, \Sigma_d^{(m)})$ for some $\{\boldsymbol{\mu}_d^{(m)}\}_{d,m=1}^{D,M}$ and positive definite matrices $\{\Sigma_d^{(m)}\}_{d,m=1}^{D,M}$.

Note that (2) is derived by using the Jensen's inequality on the marginal likelihood as follows:

$$\log p(Y|X) = \log \int P(Y, \mathbf{Z}, \mathbf{G}|X) d\mathbf{Z} d\mathbf{G} = \log \int \frac{P(Y, \mathbf{Z}, \mathbf{G}|X)}{q(\mathbf{Z}, \mathbf{G})} q(\mathbf{Z}, \mathbf{G}) d\mathbf{Z} d\mathbf{G}$$
$$\ge \int q(\mathbf{Z}, \mathbf{G}) \log \frac{P(Y, \mathbf{Z}, \mathbf{G}|X)}{q(\mathbf{Z}, \mathbf{G})} d\mathbf{Z} d\mathbf{G}. \tag{3}$$

Here, $q(\mathbf{Z}, \mathbf{G})$ and all hyper-parameters $\{\boldsymbol{\theta}, \sigma^2, \{\pi_{nm}\}\}$ are updated via an EM algorithm for (2). In the E-step, for given hyper-parameters the optimal distribution $Q^* := q^*(\mathbf{Z})q^*(\mathbf{G}) \in \mathcal{Q}$ is computed by optimizing (2) via functional derivatives with respect to $q(\mathbf{Z})$ and $q(\mathbf{G})$. Once $Q^* = q^*(\mathbf{Z})q^*(\mathbf{G})$ is determined, the hyper-parameters $\{\boldsymbol{\theta}, \sigma^2, \{\pi_{nm}\}\}$ are selected to maximize $\mathcal{L}(q^*(\mathbf{Z})q^*(\mathbf{G}))$ and this is called the M step. After alternating the E-step and M-step, the OMGP model uses the final $q^*(\mathbf{Z})$ for the association.

3. Model description and analysis

In this section, we propose our BMGP model with a new inference procedure that induces performance differences between the OMGP and BMGP models. In addition, we derive two theoretical results with regard to how efficiently the prior Gaussian processes can be updated to adapt to a given data set and how to use our BMGP model for prediction, which are the main advantages of the BMGP model over the existing models including the OMGP model.

3.1. Model

Let $\mathbf{y}(\mathbf{x})$ be an observed output at an input $\mathbf{x}$. Unlike the OMGP model, we introduce a mixture weight $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_M)$ for the latent functions over the whole input space as $\mathbf{y}(\mathbf{x}) = \mathbf{g}^{(m)}(\mathbf{x}) + \boldsymbol{\epsilon}(\mathbf{x})$ with probability π_m for any input point $\mathbf{x}$. Here, $\boldsymbol{\epsilon}(\mathbf{x})$ is the additive white Gaussian noise, which has variance σ^2. Setting a mixture weight over a function space is a widely used approach, e.g., Li et al. [8] and this is a natural generalization of the Gaussian mixture model. We use a Dirichlet prior $\boldsymbol{\pi} \sim \mathrm{Dir}(\alpha, \ldots, \alpha)$ in this paper. Even though we consider the same mixture weight $\boldsymbol{\pi}$ over all input points, the updated assignment probabilities are different from input point to input point due to our Bayesian inference. The detailed estimation is explained later.

With the estimated assignment probabilities we can associate the outputs with latent GPs. Moreover, the Dirichlet prior distribution of $\boldsymbol{\pi}$ in our model can be updated from the associated results to be a posterior distribution $\boldsymbol{\pi}|X, Y \sim \mathrm{Dir}(\alpha + c_1, \ldots, \alpha + c_n)$, where c_i is the number of outputs that are associated to the ith GP. The posterior distribution will be used for prediction at a test point as explained later. A flowchart of the BMGP model is given in Fig. 1. From now, for simplicity, given input points $X = \{\mathbf{x}_n\}_{n=1}^N$, let $\mathbf{z}(\mathbf{x}_n) := \mathbf{z}_n$. Under our modelling assumption, while (1) also

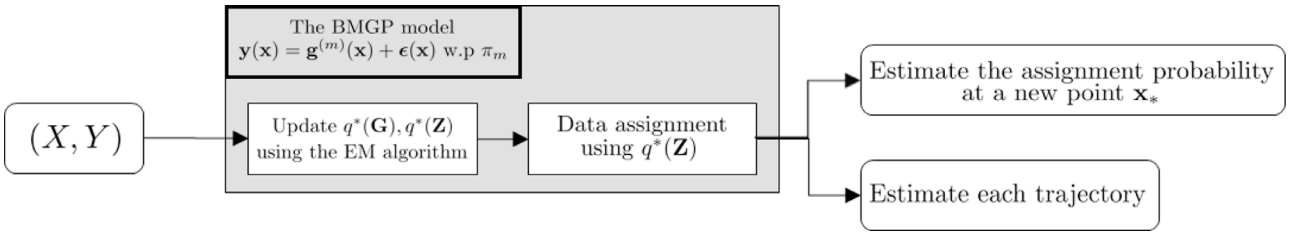


Fig. 1. The flowchart of the BMGP model.

holds for the likelihood function of our model $P(Y|\mathbf{Z}, \mathbf{G}, \boldsymbol{\pi})$, $P(\mathbf{Z})$ is changed to $P(\mathbf{Z}\,|\,\boldsymbol{\pi}) = \prod_{n,m=1}^{N,M} \pi_m^{z_{nm}}$.

3.2. Lower bound for variational inference

Note that the true posterior distribution $P(\mathbf{Z}, \mathbf{G}|X, Y, \boldsymbol{\pi})$ of our model is not tractable. Therefore, we approximate it by a distribution $q(\mathbf{Z}, \mathbf{G}) \in \mathcal{Q}$, which optimizes some objective function, as in (2). Before we derive a new lower bound, first observe that $q(\mathbf{Z}) = \prod_{n=1}^{N} q(\mathbf{z}_n)$ can be parametrized by using the softmax function $q(\mathbf{z}_n) = \prod_{m=1}^{M} \phi_{nm}^{z_{nm}}$, $\phi_{nm} = \frac{e^{\gamma_{nm}}}{\sum_{j=1}^{M} e^{\gamma_{nj}}}$. From now on $\{\phi_{nm}\}$ are called the assignment probabilities of the BMGP model.

We can then we can substitute the procedure of variational inference on the latent $\mathbf{Z}$ in the optimization to that on the parameters $\{\phi_{nm}\}_{n,m=1}^{N,M}$, as in Hensman et al. [7]. Bearing the setting in mind, we start with the following inequality that can be derived similarly as in (3). Hereafter, we drop the notation X for simplicity.

$$\begin{aligned} \log P(Y|\boldsymbol{\pi}) &= \log \int P(Y, \mathbf{Z}, \mathbf{G}|\boldsymbol{\pi}) d\mathbf{Z} d\mathbf{G} \\ &\geq \int q(\mathbf{Z}, \mathbf{G}) \log \frac{P(Y, \mathbf{Z}, \mathbf{G}|\boldsymbol{\pi})}{q(\mathbf{Z}, \mathbf{G})} d\mathbf{Z} d\mathbf{G} =: \mathcal{L}_{\boldsymbol{\pi}}(q(\mathbf{Z}, \mathbf{G})). \end{aligned} \tag{4}$$

Note that $P(Y, \mathbf{Z}, \mathbf{G}|\boldsymbol{\pi}) = P(Y|\mathbf{Z}, \mathbf{G}, \boldsymbol{\pi}) P(\mathbf{G}) P(\mathbf{Z}|\boldsymbol{\pi})$. As $q(\mathbf{Z}, \mathbf{G}) \in \mathcal{Q}$, $q(\mathbf{Z}, \mathbf{G}) = q(\mathbf{G}) q(\mathbf{Z})$. From this, $\mathcal{L}_{\boldsymbol{\pi}}(q(\mathbf{Z}, \mathbf{G}))$ can be expressed as

$$\begin{aligned} \mathcal{L}_{\boldsymbol{\pi}}(q(\mathbf{Z}, \mathbf{G})) &= E_{q(\mathbf{Z})} E_{q(\mathbf{G})}[\log P(Y|\mathbf{Z}, \mathbf{G}, \boldsymbol{\pi})] - \mathrm{KL}(q(\mathbf{G}) || P(\mathbf{G})) + H(q(\mathbf{Z})) \\ &\quad + E_{q(\mathbf{Z})}[\log P(\mathbf{Z}|\boldsymbol{\pi})] =: \mathcal{L}_{VB}(q(\mathbf{Z}, \mathbf{G})) + E_{q(\mathbf{Z})}[\log P(\mathbf{Z}|\boldsymbol{\pi})] \end{aligned} \tag{5}$$

where $H(q(\mathbf{Z}))$ is the entropy of $q(\mathbf{Z})$. We now derive a lower bound of the log marginal likelihood from $\mathcal{L}_{\boldsymbol{\pi}}(q(\mathbf{Z}, \mathbf{G}))$ as

$$\begin{aligned} \log P(Y) &= \log \int P(Y|\boldsymbol{\pi}) P(\boldsymbol{\pi}) d\boldsymbol{\pi} \\ &\geq \log \int \exp(\mathcal{L}_{\boldsymbol{\pi}}(q(\mathbf{Z}, \mathbf{G}))) P(\boldsymbol{\pi}) d\boldsymbol{\pi} =: \tilde{\mathcal{L}}_E(q(\mathbf{Z}, \mathbf{G})). \end{aligned} \tag{6}$$

On the other hand, $E_{q(\mathbf{Z})} E_{q(\mathbf{G})}[\log P(Y|\mathbf{Z}, \mathbf{G}, \boldsymbol{\pi})]$ does not depend on $\boldsymbol{\pi}$ from (1). So, $E_{q(\mathbf{Z})}[\log P(\mathbf{Z}|\boldsymbol{\pi})]$ is the only term in $\mathcal{L}_{\boldsymbol{\pi}}(q(\mathbf{Z}, \mathbf{G}))$ that depends on $\boldsymbol{\pi}$. From this fact, $\tilde{\mathcal{L}}_E(q(\mathbf{Z}, \mathbf{G}))$ is expressed as follows:

$$\tilde{\mathcal{L}}_E(q(\mathbf{Z}, \mathbf{G})) = \mathcal{L}_{VB}(q(\mathbf{Z}, \mathbf{G})) + \log \int \exp(E_{q(\mathbf{Z})}[\log P(\mathbf{Z}|\boldsymbol{\pi})]) P(\boldsymbol{\pi}) d\boldsymbol{\pi}. \tag{7}$$

Note that $\tilde{\mathcal{L}}_E(q(\mathbf{Z}, \mathbf{G}))$ is analytically tractable and can be expressed in terms of $\{\phi_{nm}\}_{n,m=1}^{N,M}$ and $\{\boldsymbol{\mu}_d^{(m)}, \Sigma_d^{(m)}\}_{d,m=1}^{D,N}$ as given in the following theorem.

Theorem 1. *For a variational distribution $q(\mathbf{Z}, \mathbf{G}) \in \mathcal{Q}$, the objective function $\tilde{\mathcal{L}}_E(q(\mathbf{Z}, \mathbf{G}))$ satisfies the following equality:*

$$\begin{aligned} &\tilde{\mathcal{L}}_E(q(\mathbf{Z}, \mathbf{G})) \\ &= \mathrm{Tr}(A\Phi^\top) - \frac{DN}{2} \log 2\pi\sigma^2 - \sum_{d,m=1}^{D,M} \mathrm{KL}(\mathcal{N}(\boldsymbol{\mu}_d^{(m)}, \Sigma_d^{(m)}) \,\|\, \mathcal{N}(0, K_m)) \\ &+ H(q(\mathbf{Z})) + \log \frac{\Gamma(M\alpha)}{\Gamma(M\alpha + N)} + \sum_{m=1}^{M} \log \frac{\Gamma(\alpha + \hat{\phi}_m)}{\Gamma(\alpha)}. \end{aligned} \tag{8}$$

where $\hat{\phi}_m = \sum_{n=1}^{N} \phi_{nm}$, $\Phi = [\phi_{nm}]_{n,m=1}^{N,M}$, $A = [a_{nm}]_{n,m=1}^{N,M}$ and

$$a_{nm} = -\frac{1}{2\sigma^2} \sum_{d=1}^{D} \left(\left(Y_{nd} - (\boldsymbol{\mu}_d^{(m)})_n \right)^2 + (\Sigma_d^{(m)})_{nn} \right).$$

As both $P(\mathbf{G})$ and $q(\mathbf{G})$ are Gaussian distributions, all terms in $\mathcal{L}_{VB}(q(\mathbf{Z}, \mathbf{G}))$ are tractable. See Appendix A for the detailed proof.

3.3. Inference and optimization of the lower bound

The inference is carried out in the following form of the EM algorithm. A schematic diagram of the EM algorithm is provided in Fig. 2.

The E-step is twofold. In the first sub-step, we update $q(\mathbf{G})$ for fixed $\boldsymbol{\theta}$, σ^2 and Φ by applying the functional derivative to $\tilde{\mathcal{L}}_E(q(\mathbf{Z}, \mathbf{G}))$ with respect to $q(\mathbf{G})$. The optimized results are summarized as follows.

Theorem 2. *For fixed $\boldsymbol{\theta}, \sigma^2$ and Φ, the following distribution $q^*(\mathbf{G}) = \prod_{d,m=1}^{D,M} \mathcal{N}(\boldsymbol{\mu}_d^{(m)*}, \Sigma_d^{(m)*})$ optimizes $\tilde{\mathcal{L}}_E(q(\mathbf{Z}, \mathbf{G}))$ with respect to $q(\mathbf{G})$ where $\Sigma_d^{(m)*} = \Sigma^{(m)*} = (K_m^{-1} + B_m)^{-1}$, $\boldsymbol{\mu}_d^{(m)*} = \Sigma^{(m)*} B_m Y_{(d)}$, and $B_m = \mathrm{diag}(\frac{\phi_{nm}}{\sigma^2})_{n=1,\ldots,N}$.*

The proof of Theorem 2 is provided in Appendix B. In the second sub-step, given $\boldsymbol{\theta}, \sigma^2$ and $q^*(\mathbf{G})$ in Theorem 2, the objective function $\tilde{\mathcal{L}}_E$ is now considered a function of Φ and we update the variational parameters Φ by applying a combined method of natural gradient ascent and conjugate gradient ascent, as in Kuusela et al. [9].

Our E-step looks similar to Hensman et al. [7], but there is a clear difference. When we update Φ, $\{\frac{\partial \Sigma^{(m)*}}{\partial \phi_{np}}, \frac{\partial \boldsymbol{\mu}_d^{(m)*}}{\partial \phi_{np}}\}_{n,m,d,p}$ are not considered in $\frac{\partial \tilde{\mathcal{L}}_E}{\partial \phi_{nm}}$ because $q^*(\mathbf{G})$ is fixed during the update of $q(\mathbf{Z})$. More precisely, when we update Φ, we only use $\tilde{\mathcal{L}}_\Phi := \mathrm{Tr}(A\Phi^\top) + H(q(\mathbf{Z})) + \sum_{m=1}^{M} \log \frac{\Gamma(\alpha + \hat{\phi}_m)}{\Gamma(\alpha)}$, which is a part of $\tilde{\mathcal{L}}_E(q(\mathbf{Z}, \mathbf{G}))$, by considering $\boldsymbol{\mu}_d^{(m)*}$ and $\Sigma^{(m)*}$ as constants with respect to Φ, which is clearly different from Hensman et al. [7]. In summary, once $\{\boldsymbol{\theta}, \sigma^2, \Phi\}$ are initialized, we compute $q^*(\mathbf{Z})$ and $q^*(\mathbf{G})$ by repeating Theorem 2 and the update of Φ using $\tilde{\mathcal{L}}_\Phi$ until $\tilde{\mathcal{L}}_E(q(\mathbf{Z}, \mathbf{G}))$ converges. As $\tilde{\mathcal{L}}_E(Q^*)$ is bounded by $\log p(Y)$ when $\boldsymbol{\theta}$ is fixed, convergence is guaranteed whenever the second sub-step using a gradient method converges well.

We now focus on the M-step. With fixed $q^*(\mathbf{Z})$ and $q^*(\mathbf{G})$, $\tilde{\mathcal{L}}_E(Q^*)$ can be now considered a function of $\{\boldsymbol{\theta}, \sigma^2\}$. The M-step hence can be carried out by using the gradient ascent method of $\tilde{\mathcal{L}}_E(Q^*)$ with respect to $\{\boldsymbol{\theta}, \sigma^2\}$. By chain rule, $\frac{\partial \tilde{\mathcal{L}}_E(Q^*)}{\partial \boldsymbol{\theta}} = \frac{\partial \tilde{\mathcal{L}}_E(Q^*)}{\partial K_m} \cdot \frac{\partial K_m}{\partial \boldsymbol{\theta}}$. While $\frac{\partial K_m}{\partial \boldsymbol{\theta}}$ are well known for many kernel functions, $\frac{\partial \tilde{\mathcal{L}}_E(Q^*)}{\partial K_m}$ cannot be analytically implementable in its current form for each m because $\{\boldsymbol{\mu}_d^{(m)*}\}_{d,m}$ and $\{\Sigma^{(m)*}\}_m$ are functions of K_m and $\boldsymbol{\mu}_d^{(m)*}$, which are contained in the KL-divergence term in $\tilde{\mathcal{L}}_E(Q^*)$. To solve this problem, we derive the following bound that is equivalent to $\tilde{\mathcal{L}}_E(Q^*)$.

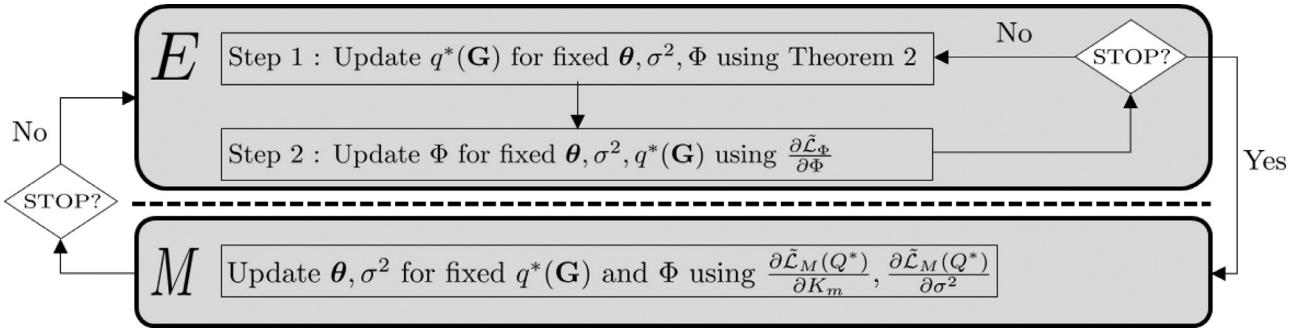


Fig. 2. Diagram summarizing the EM algorithm of the BMGP model.

Theorem 3. *For a given $Q^* = q^*(\mathbf{Z})q^*(\mathbf{G})$, the following equality holds.*

$$\tilde{\mathcal{L}}_M(Q^*) := \log \frac{\Gamma(M\alpha)}{\Gamma(M\alpha+N)} + \sum_{m=1}^{M} \log \frac{\Gamma(\alpha+\hat{\phi}_m)}{\Gamma(\alpha)} + \frac{ND(M-1)}{2}\log 2\pi\sigma^2 - \frac{D}{2}\sum_{n,m=1}^{N,M}\log \phi_{im} + \sum_{d,m=1}^{D,M}\log \mathcal{N}(Y_{(d)}|\mathbf{0}, K_m + B_m^{-1}) + H(q^*(\mathbf{Z})) = \tilde{\mathcal{L}}_E(Q^*)$$

Note that a direct substitution of Theorem 2 into (8) is intractable because of the matrix A and $\mathrm{KL}(q^*(\mathbf{G})||P(\mathbf{G}))$. We therefore use another form of $q^*(\mathbf{G}) = \frac{1}{C}P(\mathbf{G})\exp\left(E_{q^*(\mathbf{Z})}[\log P(Y|\mathbf{G},\boldsymbol{\pi},\mathbf{Z})]\right)$ which is derived in the proof of Theorem 2. See Appendix C for the proof. Even though the OMGP model also suggests another lower bound referred to as the KL-corrected bound which is similar to $\tilde{\mathcal{L}}_M(Q^*)$, it provides insufficient derivation. Due to Theorem 3, we can use $\tilde{\mathcal{L}}_M(Q^*)$ to update $\boldsymbol{\theta}$ and σ^2. Note that as $\tilde{\mathcal{L}}_M(Q^*)$ does not contain the terms of $\boldsymbol{\mu}_d^{(m)*}$ and $\Sigma^{(m)*}$, the exact computation of $\frac{\partial \tilde{\mathcal{L}}_M(Q^*)}{\partial K_m}$ and $\frac{\partial \tilde{\mathcal{L}}_M(Q^*)}{\partial \sigma^2}$ is possible by using the matrix derivative in Petersen and Pedersen [10]. Algorithm 1 is a pseudo code of our EM algorithm.

Algorithm 1 EM algorithm.

Input: Initial hyper-parameters $\boldsymbol{\theta}_0, \sigma_0^2$ and Φ_0
Set $\boldsymbol{\theta} = \boldsymbol{\theta}_0, \sigma^2 = \sigma_0^2$ and $\Phi = \Phi_0$
repeat
  Set the current hyper-parameters $\boldsymbol{\theta}, \sigma^2$ as $\boldsymbol{\theta}_{\text{old}}, \sigma^2_{\text{old}}$.
  For a given hyper-parameters and Φ, update Φ from the E-step
  Update $\boldsymbol{\theta}_{\text{new}}$, σ^2_{new} using a gradient based method with $\{\frac{\partial \tilde{\mathcal{L}}_M(Q^*)}{\partial K_m}, \frac{\partial \tilde{\mathcal{L}}_M(Q^*)}{\partial \sigma^2}\}$.
until $|\tilde{\mathcal{L}}_M(Q^*(\boldsymbol{\theta}_{\text{new}})) - \tilde{\mathcal{L}}_M(Q^*(\boldsymbol{\theta}_{\text{old}}))| < \epsilon'$ for some $\epsilon' > 0$.

3.4. Major differences of the EM algorithm between the OMGP and the BMGP model

There are two main differences in updating the assignment probabilities between the OMGP and BMGP models. First, $\{\pi_{nm}\}$ of the OMGP are considered hyper-parameters and are updated simultaneously with kernel hyper-parameters $\boldsymbol{\theta}$ and σ^2 in the M-step. However, $\{\phi_{nm}\}$ of the BMGP are updated in the E-step. After the E-step, only $\boldsymbol{\theta}$ and σ^2 are updated in the M-step via $\max_{\boldsymbol{\theta},\sigma^2} \tilde{\mathcal{L}}_M(q^*(\mathbf{G})q^*(\mathbf{Z}))$. The OMGP model thus should consider a more complex hyper-parameter space containing $\{\pi_{nm}\}$, which may lead to choosing a bad $\{\pi_{nm}\}$ and hence results in poor data association. The BMGP model separates the updates of assignment probabilities and kernel hyper-parameters to remedy the curse of dimensionality problem [11].

Second, the BMGP model directly parameterizes the distribution $q(\mathbf{Z})$ via a softmax function and this makes it possible to compute the natural gradient in the E-step by using the Fisher information matrix of $q(\mathbf{Z})$ in terms of Φ. Hence, unlike the OMGP model the update of the assignment probabilities can be based on the natural gradient which has a clear superiority because it uses the steepest gradient direction in the distribution space compared with the baseline gradient [12]. These two differences in updating the assignment probabilities can lead to better data association results in the BMGP model than in the OMGP model as shown experimentally in Section 5.

3.5. The differences of the prior covariance matrix and the approximate posterior covariance matrix

For given $\boldsymbol{\theta}$ and σ^2, we compare the prior covariance matrix $\{K_m\}_{m=1}^M$ and the approximate posterior covariance matrix $\{\Sigma^{(m)*}\}_{m=1}^M$ in this subsection. We give a quantitative analysis of the differences between the approximate posterior GP $q^*(\mathbf{G})$ and the prior GP $P(\mathbf{G})$ in the E-step. For each m, if we find a posterior covariance function k_m^* that has Gram matrix $K_m^*(X,X) = \Sigma^{(m)*}$, then we can analytically compute the difference between the prior kernel k_m and the posterior kernel k_m^* for each GP. However, we only obtain a posterior covariance $\Sigma^{(m)*}$, not the posterior kernel function. Therefore, instead of considering the difference between kernel functions, we concentrate on the difference between covariance matrices $\|K_m - \Sigma^{(m)*}\|$ for each m. If $\|K_m - \Sigma^{(m)*}\|$ is large, then it is reasonable to say that k_m and an unknown posterior kernel function k_m^* are quite different, which implies that the association matrix B_m results in a large difference between them and the learning result becomes more suitable for given observations.

Proposition 1. *Suppose that $\boldsymbol{\theta}$ and σ^2 satisfy*

$$\sigma^2 < \frac{\lambda_1(K_m)\lambda_N(K_m)\max_{n=1,\dots,N}\phi_{nm}}{\lambda_1(K_m) - \lambda_N(K_m)} \tag{9}$$

where $\lambda_i(A)$ are eigenvalues of a matrix A with $\lambda_1(A) \geq \dots \lambda_N(A)$, and $\phi_{nm} > 0$ for all $n = 1,\dots,N$ and $m = 1,\dots,M$. For each m, if we define $\phi_{*m} := \max_{n=1,\dots,N}\phi_{nm}$, then the following inequality holds.

$$\|K_m - \Sigma^{(m)*}\|_2 \geq \lambda_N(K_m) - \frac{\sigma^2\lambda_1(K_m)}{\sigma^2 + \lambda_1(K_m)\phi_{*m}}. \tag{10}$$

The proof of Proposition 1 is based on the Woodbury matrix formula and Weyl's inequality. See Appendix D for details. Note that an assumption (9) is needed to make the lower bound (10) valid. From Proposition 1, we see that under a mild assumption on hyper-parameters, the difference between the prior kernel function and approximate posterior kernel function for a GP increases as $\{\phi_{*m}\}_{m=1}^M$ increase, i.e, it increases as the model is more suitable for given observations. In summary, Proposition 1 theoretically shows how much our model is updated for data association

and this derivation is possible due to our use of the mixture of GPs.

3.6. The approximated predictive distributions

When the inference is done, we can compute the number of outputs, c_m, that are believed to be generated from $g^{(m)}$ using the updated Φ. By using the posterior distribution $\boldsymbol{\pi}|X, Y$ as the mixture weight at a test point $\mathbf{x}^*$, we can obtain the approximated predictive distribution of y_d^* as follows.

Proposition 2. *Let c_m be the number of observations that are believed to be generated from $g^{(m)}$ using the updated BMGP. Then,*

$$P(y_d^* \mid \mathbf{x}^*, X, Y) \simeq \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(y_d^* \mid \mu_d^{(m)*}(\mathbf{x}^*), \sigma_d^{2(m)*}(\mathbf{x}^*))$$

where $\hat{\pi}_m = \frac{\alpha + c_m}{M\alpha + N}$, $\mu_d^{(m)*}(\mathbf{x}^*) = K_m(X, \mathbf{x}^*)^\top (K_m + B_m^{-1})^{-1} Y_{(d)}$, *and* $\sigma_d^{2(m)*}(\mathbf{x}^*) = \sigma^2 + k_m(\mathbf{x}^*, \mathbf{x}^*) - K_m(X, \mathbf{x}^*)^\top (K_m + B_m^{-1})^{-1} K_m(X, \mathbf{x}^*)$.

See Appendix E for the proof. Note that the OMGP model cannot estimate the assignment probabilities π_{*m} at a new test point $\mathbf{x}^*$ by itself without any additional assumption or hyper-parameters. On the other hand, the BMGP model does not need any additional information for prediction as explained above, which is another benefit of the BMGP model.

4. Related works

In this section, we explain some related works from two aspects: data association and the use of the mixture of GPs. Data association was initially employed to deal with the multiple target tracking (MTT) problem, whose goal is to jointly detect targets and estimate the trajectories of targets from observations. Hence, there are many related data association methods and some traditional methods include filter based methods such as joint probabilistic data association filter [13] and Markov chain Monte Carlo data association [14]. Such traditional methods have limitations in application to real data sets because they require some domain knowledge due to their complex modelling or some heuristics on motion geometry [15]. Recently, most of the data association methods have been focused on tracking objects in videos based on the tracking-by-detection approach, which first detects and segments some objects in each frame and then links the detections to obtain the trajectory of each object. In particular, deep neural networks such as the convolutional neural network [16] and recurrent neural network [17] are widely used for detection and linking. One recent approach uses the variational autoencoder (VAE) for data association, e.g., image segmentation in video [18] and clustering [19]. However, the VAE based approach performs data association in the latent space (not in the observation space) and hence still cannot capture the latent functions of the observations lying in the observation space.

Regarding the use of a mixture of GPs, early works [20] partitioned the input data and updated GPs locally on the partitions. The whole data hence could not be used to update the GPs. Later, the OMGP model [6] was proposed to use the whole data by introducing the assignment probabilities of observations. An optimization problem called a constrained max K-section problem is used in Lázaro-Gredilla and Van Vaerenbergh [21] for data association instead of using the gradient descent method for the assignment probabilities. However, the work in Lázaro-Gredilla and Van Vaerenbergh [21] is limited to cases where two mixtures are enough because the problem can be analytically solved only when $K = 2$. Recently, the author of Kaiser et al. [22] suggested a Bayesian model that introduces additional GPs to model the assignment probabilities of outputs with a softmax function and multinomial distribution. Later, Liu et al. [23] proposed a similar method that has a tighter bound than [22]. As a drawback of Kaiser et al. [22], Liu et al. [23], they require some numerical approximations in the inference procedure, which makes it difficult to theoretically analyze how effectively the model is updated via this procedure. In addition, the application of both methods is limited to cases where the output only has one dimension.

5. Experiments

In this section, we carry out experiments for multiple data association tasks. For GPs, we use M ARD SE kernel functions $k_m(\mathbf{x}, \mathbf{x}') = \sigma_{f,m}^2 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} \frac{(x_q - x_q')^2}{\ell_{q,m}^2}\right)$ in all experiments. Moreover, since we assume zero mean GPs, we normalize the output matrix Y for each dimension. The implementation of our model is based on GPy [7,24].[1]

Note that as $\tilde{\mathcal{L}}_M(Q^*)$ is a non-convex function in the hyper-parameters, selecting proper initial hyper-parameters is important [25]. First, we use $\sigma_{f,m}^2 = \sigma_f^2 = \frac{\mathrm{Var}(\mathbf{y}(\mathbf{x}))}{D} \simeq$ the mean of the sample variance of $\{Y_{(d)}\}_{d=1}^D$. Next, we randomly sample from the prior $\ell_{q,m} \sim \mathrm{Unif}[10^{-3}, \mathrm{MaxI}_q + 10^{-3}]$ for each $m = 1, \ldots, M$ and $q = 1, \ldots, Q$, which is suggested in Chen and Wang [26]. Here, $\mathrm{MaxI}_q = \max_{i,j=1,\ldots,N} \|x_{i,q} - x_{j,q}\|$ is the *maximal range of the inputs* which was used in Wilson [27] for lengthscale parameters.

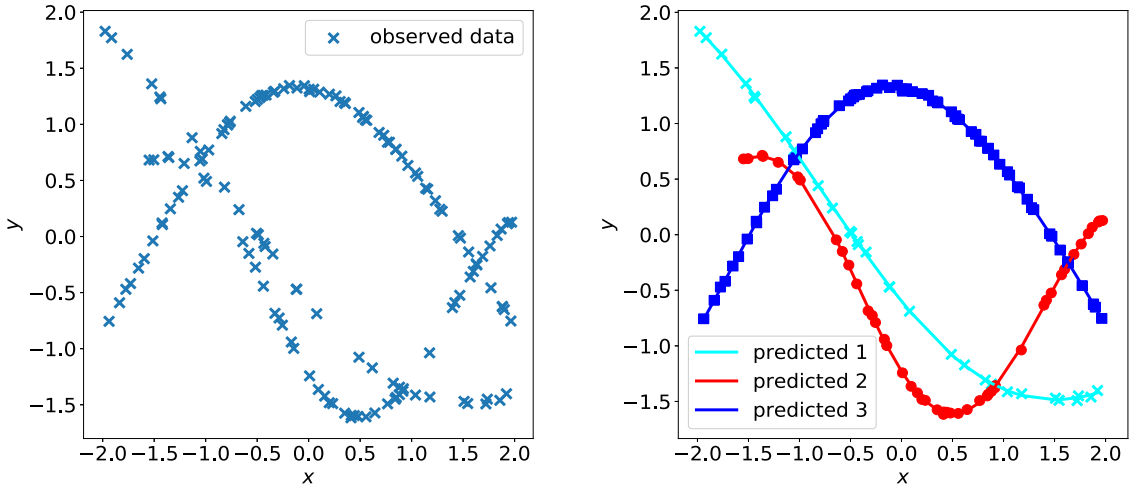
We first compare the BMGP model with the OMGP model[2] which is based on the GPML toolbox [28] on two synthetic data sets. For a clear comparison, we use the same L-BFGS-B [29] optimizer for kernel hyper-parameters of both models with the same initial values of kernel hyper-parameters to emphasize that the different experimental results of the two models originate from our new EM algorithm. We next compare our method with a VAE based method called DGG [19], which was recently proposed for data association, using some real data sets. As DGG requires several neural networks for encoders and decoders, we use the same settings as given by the authors' open sources[3] for the implementation. The experimental results are provided in Tables 1–3 where the values in bold indicate the best results in comparison.

5.1. Toy data set

We first apply the BMGP model to a data set that is generated by a mixture of three independent zero mean GPs f_1, f_2, and f_3 with the mixture weights $\pi_1 = 0.3$, $\pi_2 = 0.5$, $\pi_3 = 0.2$. The GPs have the SE kernel functions with lengthscale parameters $l = 1, 1.5, 2$ and variance parameters $\sigma^2 = 1, 2, 3$, respectively. We uniformly sample 150 input points in $[-2, 2]$. Based on π_1, π_2, π_3, we split 150 points into X_1, X_2, X_3, where X_m is the set of input points, which are assigned to the mth GP. For simplicity, let C_m be the covariance matrix of f_m for the input points X_m. We obtain the observations from sampling $Y_m \sim f_m(X_m) = \mathcal{N}(\mathbf{0}, C_m)$ with noise $\mathcal{N}(0, 0.01^2)$. For clarity, denote $X_m = \{\mathbf{x}_1^{(m)}, \ldots, \mathbf{x}_{c_m}^{(m)}\}$, $Y_m = \{Y_1^{(m)}, \ldots, Y_{c_m}^{(m)}\}$ with c_m defined in Proposition 2 for each $m = 1, 2, 3$. We use $M = 3$ for both the OMGP and BMGP models.

As the results depend on the sampling of observations, we repeat the experiment 10 times with different sampled observations. One case of the experiments and its association results of our BMGP model are provided in Fig. 3. Since both the BMGP and OMGP models associate three trajectories perfectly in every experiment, we only provide the results of the BMGP model in the figure.

[1] github.com/SheffieldML/GPclust.
[2] www.tsc.uc3m.es/~miguel/downloads.php.
[3] https://github.com/dodoyang0929/DGG.git.



(a) Observations (b) The BMGP model

Fig. 3. One case from 10 experiments in Section 4.1. Different trajectories are marked by three different markers ×, ●, and ■.

Table 1
The performance results of the OMGP model and the BMGP model on the toy data set. All results are reported with the bounds of 95% confidence intervals.

Measures	OMGP	BMGP
KL_1	169.9422 ± 21.4267	$\mathbf{37.8851 \pm 4.6759}$
KL_2	265.0992 ± 21.1025	$\mathbf{40.2215 \pm 7.6487}$
KL_3	110.5419 ± 17.3248	$\mathbf{39.9676 \pm 14.2380}$
RMSE	0.00947 ± 0.00025	$\mathbf{0.00927 \pm 0.00027}$
$KL(\hat{\boldsymbol{\pi}}\|\|\boldsymbol{\pi})$	0.0125 ± 0.01675	$\mathbf{0.0056 \pm 0.0038}$

We use various quantitative measures of performance and they are reported in Table 1 with means and bounds of the 95% confidence intervals over 10 experiments. The measures are explained below.

First, we compute the root mean squared error (RMSE) between the approximated predictive mean and the observation defined as $\text{RMSE} = \sqrt{\frac{1}{150}\sum_{m=1}^{3}\sum_{i=1}^{c_m} \|\boldsymbol{\mu}^{(m)*}(\mathbf{x}_i^{(m)}) - Y_i^{(m)}\|_2^2}$, where $\|\cdot\|_2$ is the Euclidean norm. As both models allocate each trajectory well, the difference in the average RMSE is not significant.

However, there is a huge difference in estimating the three latent Gaussian distributions. To measure how well the approximated posterior GPs estimate the true generative processes, we compute $\text{KL}_m := \text{KL}(q^*(g^{(m)}(X_m)) \,\|\, \mathcal{N}(\mathbf{0}, C_m))$ for each $m = 1, 2, 3$. We also compute $\text{KL}(\hat{\boldsymbol{\pi}} \,||\, \boldsymbol{\pi})$, where $\boldsymbol{\pi} = (0.3, 0.5, 0.2)$ is the true mixture weight and $\hat{\boldsymbol{\pi}} = (\hat{\pi}_1, \hat{\pi}_2, \hat{\pi}_3)$ is the approximated mixture weight suggested in Section 3.6. Since the OMGP model cannot estimate the true mixture weight by itself, we assume that the initial mixture weights for all data points and a test point are the same and use the updated assignment probabilities by the OMGP model as an estimate for the mixture weight.

As we can see in Table 1, the BMGP model performs better than the OMGP model in estimating the latent generative processes and the true mixture weights as $\text{KL}_1, \text{KL}_2, \text{KL}_3$ and $\text{KL}(\hat{\boldsymbol{\pi}} \,||\, \boldsymbol{\pi})$ of the BMGP model are much smaller than those of the OMGP model.

5.2. Missile-to-air multi-target tracking simulation

We consider two data sets obtained from missile tracking simulation in Karlsson and Gustafsson [30]. They are generated by a state-space equation where each state consists of the position and velocity components of a source, $s_t = (X_t, Y_t, Z_t, V_{x,t}, V_{y,t}, V_{z,t})$. The motion dynamics are defined as follows:

$$s_{t+1} = \begin{bmatrix} I_{3\times3} & TI_{3\times3} \\ \mathbf{0} & I_{3\times3} \end{bmatrix} s_t + \begin{bmatrix} \frac{T^2}{2} I_{3\times3} \\ TI_{3\times3} \end{bmatrix} v_t, \; r_t = h(s_t) = \begin{bmatrix} \sqrt{X_t^2 + Y_t^2 + Z_t^2} \\ \arctan(\frac{Y_t}{X_t}) \\ \arctan(-\frac{Z_t}{\sqrt{X_t^2+Y_t^2}}) \end{bmatrix} + e_t$$

where T is the sampling interval and r_t is the observation at time t. Here, v_t and e_t are noise processes that are assumed to be Gaussian with $v_t \sim \mathcal{N}(0, R_1)$ and $e_t \sim \mathcal{N}(0, R_2)$ for some covariance matrices R_1 and R_2. We use the setting in Karlsson and Gustafsson [30]. For a visualization we use the first and second coordinates of r_t as the observations.

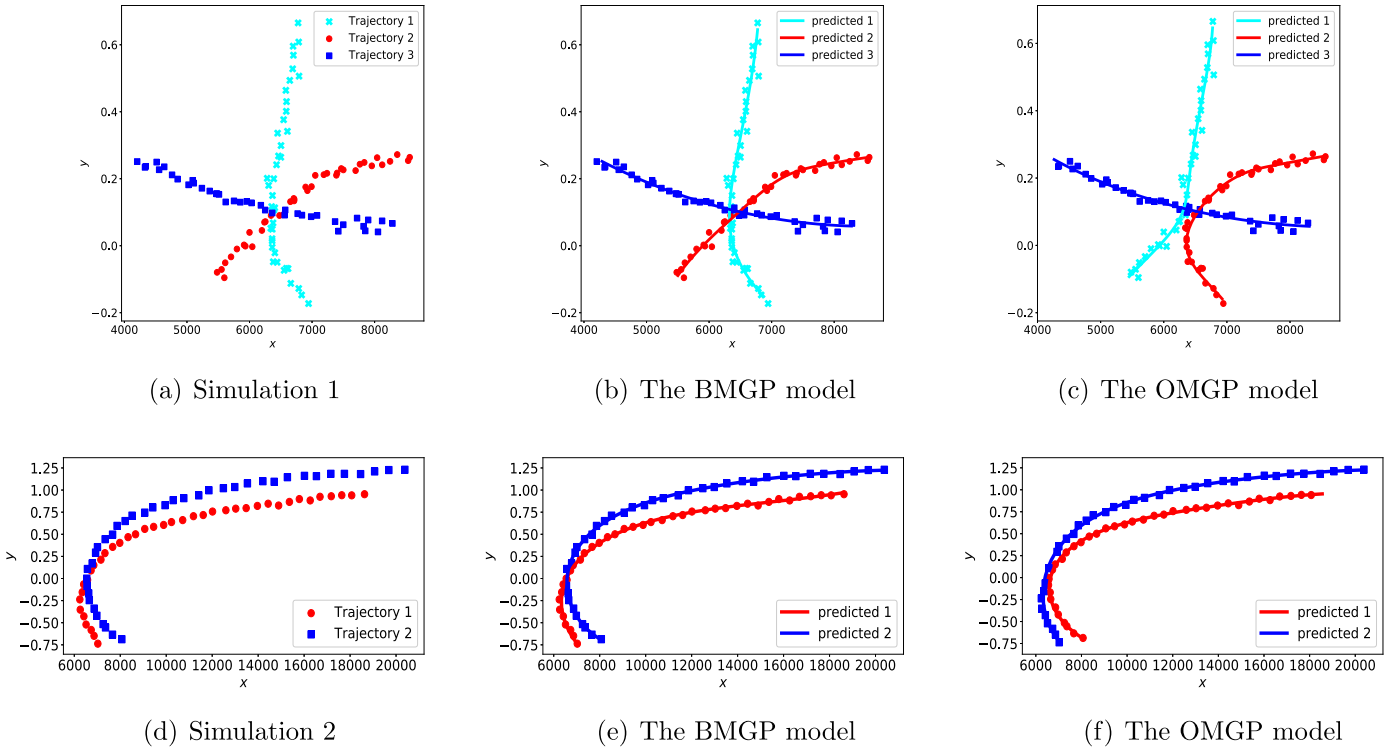
We consider two cases. The first is the three source case discussed in Lázaro-Gredilla et al. [6] and the second is the two source case discussed in Lázaro-Gredilla and Van Vaerenbergh [21] on which the OMGP model performs poorly. We provide the generated observations and the experimental results in Fig. 4. When we train both the BMGP and OMGP models, the true association and generative state-space relations are hidden and the goal of the two cases is to make an exact association, as shown in Fig. 4(a) and (d).

We use $M = 3$ in Simulation 1 and $M = 2$ in Simulation 2 for both models.

For both cases, there are points where all trajectories cross over and it is known that such cross-over points make the assignment task become more difficult. While the OMGP model does not perform well, the BMGP model performs well as shown in Fig. 4. For a precise evaluation of the performance of each model, we check the number of observations with wrong assignment, denoted by n_{err}, and compute the average of the normalized mean squared error (NMSE) between the approximated predictive means and the observations, defined as follows:

$$\text{Average NMSE} := \frac{1}{D}\sum_{d=1}^{D} \frac{\frac{1}{N}\sum_{m=1}^{M}\sum_{i=1}^{c_m} \|\boldsymbol{\mu}_d^{(m)*}(\mathbf{x}_i^{(m)}) - Y_{id}^{(m)}\|_2^2}{\text{Sample variance of } Y_{(d)}}$$

where we use the same notation X_m, Y_m, and c_m given in Section 5.1. We use the average NMSE because the scales of all axes are quite different. The results are provided in Table 2. As seen in the table, the BMGP model performs better than the OMGP model in both data association and estimation of the latent property which generates the observations.



(a) Simulation 1 (b) The BMGP model (c) The OMGP model

(d) Simulation 2 (e) The BMGP model (f) The OMGP model

Fig. 4. Simulation 1 is generated from $s_0^{(1)} = (6500, -1000, 2000, -50, 100, 0)$, $s_0^{(2)} = (5050, -450, 2000, 100, 50, 0)$ and $s_0^{(3)} = (8000, 500, 2000, -100, 0, 0)$. Simulation 2 is generated from $s_0^{(1)} = (6000, -5000, 2000, 10, 550, 0)$ and $s_0^{(2)} = (5050, -4500, 2000, 100, 500, 0)$. Each of trajectories has 40 observations. Different trajectories are marked by different markers.

Table 2
The average NMSE and n_{err} comparison results on two simulations of missile-to-air data association.

	BMGP		OMGP	
	average NMSE	n_{err}	average NMSE	n_{err}
Simulation 1	**0.0039**	**4**	1.7052	34
Simulation 2	**0.0003**	**1**	0.0029	21

Table 3
The RMSE and n_{err} comparison results on two real data sets.

	BMGP		OMGP		DGG
	RMSE	n_{err}	RMSE	n_{err}	n_{err}
T-drive ($N = 296$)	**0.0095**	**4**	0.0170	20	78
KITTI-17 ($N = 192$)	**12.3908**	**6**	23.9599	40	101
TUD-Stadtmitte ($N = 567$)	**0.8411**	**5**	1.3673	22	68
MOT16-09 ($N = 772$)	**5.5330**	**0**	6.8189	34	202

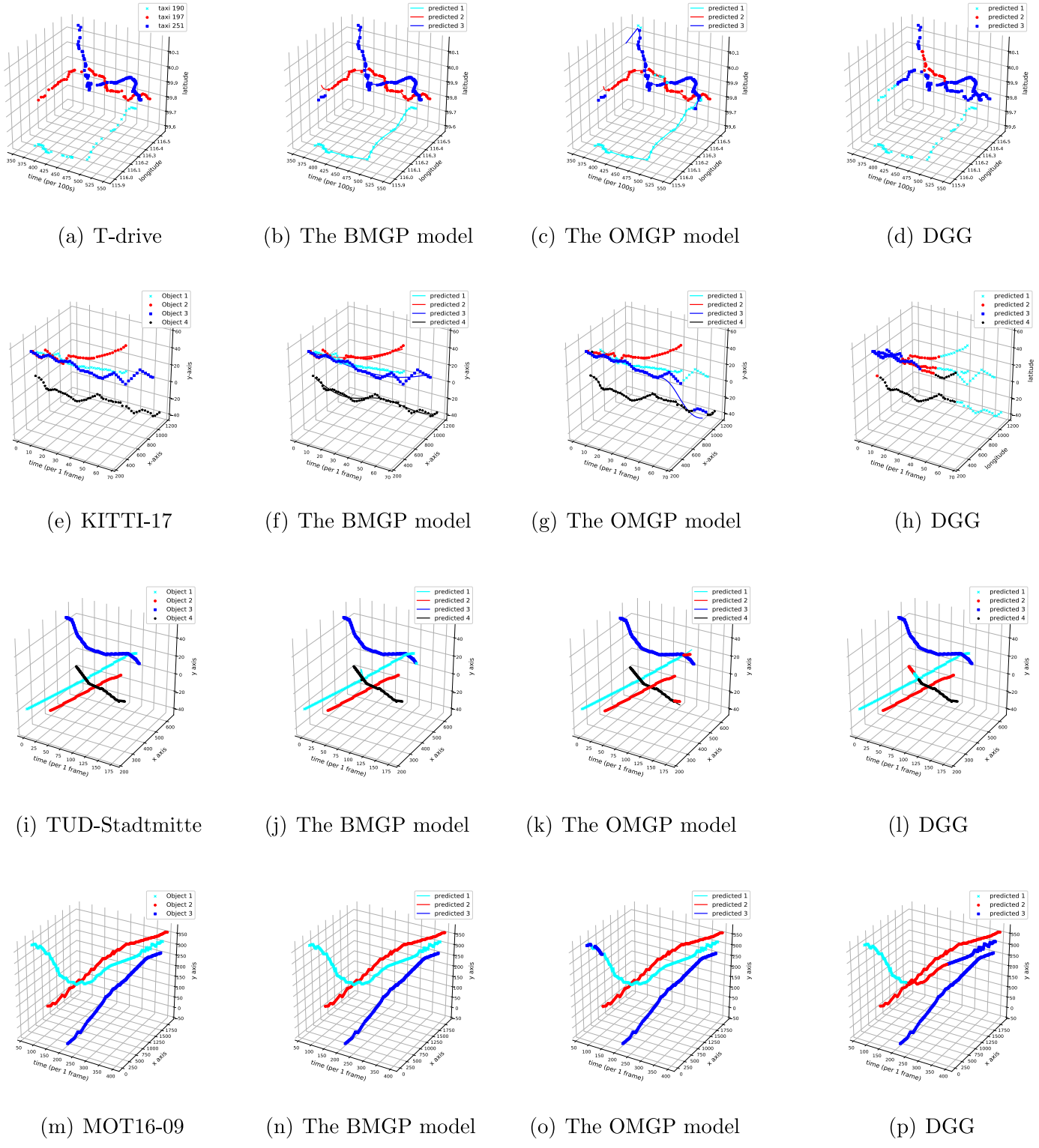
5.3. Real data: trajectory association tasks

In this subsection we use some real 2D trajectory data sets such as GPS trajectories and several pedestrians' paths. We apply the OMGP and BMGP models to the real data sets. We also apply the VAE based method called DGG [19] to the same data sets. While [18] is also promising, we use DGG because of the following two reasons: (1) it is applicable to non-video data sets while [18] is only applicable to videos; and, (2) it has some similarities with the OMGP and BMGP models in the sense that DGG also models the assignment probabilities of observations and updates them via functional derivative.

Note that input points and outputs for the OMGP and BMGP models are the observed times $x_i \in \mathbb{R}^1$ and the observed positions $\mathbf{y}_i \in \mathbb{R}^2$, respectively. On the other hand, the input points in DGG are $\{\mathbf{t}_i := (x_i, \mathbf{y}_i) \in \mathbb{R}^3\}_{i=1}^n$ as DGG is a clustering algorithm. Note that DGG can only obtain the posterior distribution of observations $\{\mathbf{t}_i\}_{i=1}^n$ in the observation space and hence cannot predict the position at a given time. It is thus impossible for DGG to compute the average NMSE or RMSE.

We first consider the T-Drive data [31], which contain the GPS trajectories of the taxes. In our experiment, we select three trajectories observed on the same day. Next, we consider some clipped video sequences in KITTI-17, TUD-Stadtmitte of MOT15 [32], and MOT16-09 of MOT16 [33]. Here, we use four trajectories for the videos in MOT15 and three trajectories for the videos in MOT16 consisting of the positions of the centers of bounding boxes for pedestrians. The observations and association results are provided in Fig. 5. As all three models require the number of mixtures, we set it to the number of trajectories for all models.

We compute n_{err}'s of all models and the RMSEs of the OMGP and BMGP models between the approximated predictive means for observations and their trajectories to evaluate the performance. The results are reported in Table 3. We observe from Fig. 5 and Table 3 that the BMGP model outperforms the others for the real trajectory data sets. Moreover, from the figure we see that the approximated predictive means of the OMGP model for each GP appear to be overfitted to the observations by choosing very small length-scale hyper-parameters, due to its complicated hyper-parameter space containing $\boldsymbol{\theta}, \sigma^2$, and $\{\pi_{nm}\}_{n,m=1}^{N,M}$. DGG shows worse performance compared with the GP based methods, especially for the KITTI-17 data set. We think that such bad association results on the KITTI-17 data set for DGG derive from the fact that it cannot capture the latent functions of observations. Another plausible reason is that we have an insufficient number of observations per each trajectory to train DGG. Note that DGG has more parameters than the GP based methods due to neural networks in VAE



(a) T-drive (b) The BMGP model (c) The OMGP model (d) DGG

(e) KITTI-17 (f) The BMGP model (g) The OMGP model (h) DGG

(i) TUD-Stadtmitte (j) The BMGP model (k) The OMGP model (l) DGG

(m) MOT16-09 (n) The BMGP model (o) The OMGP model (p) DGG

Fig. 5. The total numbers of observations for T-drive, KITTI-17, TUD-Stadtmitte, and MOT16-09 data set are 296, 192, 567, and 772, respectively. Different trajectories are marked by different markers ×, ●, ■ and ⋆. The figures in the first column are true trajectories.

and hence obviously requires more observations to train than the GP based methods, which is a drawback of DGG.

6. Conclusions

In this paper, we have presented a Bayesian approach based on a mixture of GPs. In the proposed method, the assignment probabilities and the GPs are simultaneously updated through a new EM algorithm by separating the assignment probabilities and the kernel-hyper parameters and hence it is more adaptive to the observations. Through analysis we have derived an inequality that shows the effectiveness of our Bayesian update. We have also validated the proposed approach on real trajectory data sets as well as two synthetic data sets.

As future work, we consider two issues: the scalability problem and the assumption of independence with respect to each dimension of observations. The first comes from the computation of the inverse of Gram matrix and the variational parameters. We try to solve it based on the inducing point method [34], and we believe that the independence issue in the second can be solved by using a multi-output Gaussian process [35].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Disaster-Safety Platform Technology Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (Grant no. NRF-2019M3D7A1095629) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (Grant no. NRF-2019R1A5A1028324).

Appendix A. Proof of Theorem 1

Note first that $\mathrm{KL}(q(\mathbf{G})||P(\mathbf{G}))$ becomes the KL divergence term in (8). Next, we compute two terms in (7). From (1) to (5), it follows that

$$E_{q(\mathbf{G})q(\mathbf{Z})}[\log P(Y|\mathbf{Z},\mathbf{G},\boldsymbol{\pi})] = \sum_{d,n,m=1}^{D,N,M} \phi_{nm} E_{q(\mathbf{G})}[\log \mathcal{N}(Y_{nd}|g_d^{(m)}(\mathbf{x}_n),\sigma^2)] = -\frac{DN}{2}\log 2\pi\sigma^2 + \mathrm{Tr}(A\Phi^\top).$$

In addition, we see that

$$E_{q(\mathbf{Z})}[\log P(\mathbf{Z}|\boldsymbol{\pi})] = \sum_{n=1}^{N} E_{q(\mathbf{z}_n)}[\log P(\mathbf{z}_n|\boldsymbol{\pi})] = \sum_{n=1}^{N}\sum_{m=1}^{M} \phi_{nm}\log\pi_m =: \sum_{m=1}^{M} \hat{\phi}_m \log\pi_m$$

Since we set $\boldsymbol{\pi} \sim \mathrm{Dir}(\alpha,\dots,\alpha)$, we obtain the second term in (7) as

$$\int \exp(E_{q(\mathbf{Z})}[\log P(\mathbf{Z}|\boldsymbol{\pi})])P(\boldsymbol{\pi})d\boldsymbol{\pi} = \int e^{\sum_{m=1}^{M}\hat{\phi}_m\log\pi_m}\frac{\Gamma(M\alpha)}{\Gamma(\alpha)^M}\prod_{m=1}^{M}\pi_m^{\alpha-1}d\boldsymbol{\pi} = \frac{\Gamma(M\alpha)}{\Gamma(M\alpha+N)}\cdot\prod_{m=1}^{M}\frac{\Gamma(\alpha+\hat{\phi}_m)}{\Gamma(\alpha)}. \tag{A.1}$$

By combining all results with (A.1), we obtain Eq. (8). □

Appendix B. Proof of Theorem 2

We want to find the optimal $\{\boldsymbol{\mu}_d^{(m)}, \Sigma_d^{(m)}\}_{d,m=1}^{D,M}$ that maximize $\tilde{\mathcal{L}}_E(q(\mathbf{Z},\mathbf{G}))$ for a fixed $\boldsymbol{\theta},\sigma^2$ and Φ. It is based on a baseline EM algorithm that uses a functional derivative. Note that $\mathcal{L}_{VB}(q(\mathbf{Z},\mathbf{G}))$ is the only term in $\tilde{\mathcal{L}}_E(q(\mathbf{Z},\mathbf{G}))$ that depends on $q(\mathbf{G})$. Hence, it suffices to solve $\mathrm{argmax}_{q(\mathbf{G})}\mathcal{L}_{VB}(q(\mathbf{Z},\mathbf{G}))$. From the Lagrange multiplier method, we define an objective function $\mathcal{L}_{obj} := \mathcal{L}_{VB}(q(\mathbf{Z},\mathbf{G})) + \lambda\left(\int q(\mathbf{G})d\mathbf{G} - 1\right)$ for $\lambda > 0$. By taking derivatives with respect to $q(\mathbf{G})$ and λ, we obtain

$$\frac{\delta\mathcal{L}_{obj}}{\delta q(\mathbf{G})} = E_{q(\mathbf{Z})}[\log P(Y|\mathbf{Z},\mathbf{G},\boldsymbol{\pi})] + \log P(\mathbf{G}) - \log q(\mathbf{G}) - 1 + \lambda = 0,$$
$$\frac{\partial\mathcal{L}_{obj}}{\partial\lambda} = \int q(\mathbf{G})d\mathbf{G} - 1 = 0.$$

By solving the above equations, we get $q^*(\mathbf{G}) = \frac{1}{C}P(\mathbf{G})\exp\left(E_{q(\mathbf{Z})}[\log P(Y|\mathbf{Z},\mathbf{G},\boldsymbol{\pi})]\right)$ where C is the normalizing constant. From (1), it follows that

$$\exp(E_{q(\mathbf{Z})}[\log P(Y|\mathbf{Z},\mathbf{G},\boldsymbol{\pi})]) \propto \prod_{d,m=1}^{D,M}\mathcal{N}(Y_{(d)}|g_d^{(m)}(X),B_m^{-1}). \tag{B.1}$$

Finally, from $P(\mathbf{G}) = \prod_{d,m=1}^{D,M}\mathcal{N}(g_d^{(m)}(X)|\mathbf{0},K_m)$ we have

$$P(\mathbf{G})\exp\left(E_{q(\mathbf{Z})}[\log P(Y|\mathbf{Z},\mathbf{G},\boldsymbol{\pi})]\right) \propto \prod_{d,m=1}^{D,M}\mathcal{N}\left(g_d^{(m)}(X)\middle|\mathbf{0},K_m\right)\mathcal{N}\left(Y_{(d)}\middle|g_d^{(m)}(X),B_m^{-1}\right) \propto \prod_{d,m=1}^{D,N}\mathcal{N}\left(g_d^{(m)}(X)\middle|\boldsymbol{\mu}_d^{(m)*},\Sigma^{(m)*}\right).$$

Hence, $q^*(\mathbf{G})$ is given by $q^*(\mathbf{G}) = \prod_{d,m=1}^{D,N}\mathcal{N}(g_d^{(m)}(X)|\boldsymbol{\mu}_d^{(m)*},\Sigma^{(m)*})$ where $\Sigma^{(m)*} = (K_m^{-1}+B_m)^{-1}$, $\boldsymbol{\mu}_d^{(m)*} = \Sigma^{(m)*}B_mY_{(d)}$.□

Appendix C. Proof of Theorem 3

Assume that an optimal distribution $Q^* = q^*(\mathbf{G})q^*(\mathbf{Z})$ is obtained from the E-step. As

$$\tilde{\mathcal{L}}_E(Q^*) = \log\frac{\Gamma(M\alpha)}{\Gamma(M\alpha+N)} + \sum_{m=1}^{M}\log\frac{\Gamma(\alpha+\hat{\phi}_m)}{\Gamma(\alpha)} + H(q^*(\mathbf{Z})) + E_{q^*(\mathbf{G})q^*(\mathbf{Z})}[\log P(Y|\mathbf{Z},\mathbf{G},\boldsymbol{\pi})] - \int q^*(\mathbf{G})\log\frac{q^*(\mathbf{G})}{P(\mathbf{G})}d\mathbf{G}, \tag{C.1}$$

by plugging $q^*(\mathbf{G})$ in Appendix B into the last term in (C.1), we obtain the following:

$$\tilde{\mathcal{L}}_E(Q^*) = \log\frac{\Gamma(M\alpha)}{\Gamma(M\alpha+N)} + \sum_{m=1}^{M}\log\frac{\Gamma(\alpha+\hat{\phi}_m)}{\Gamma(\alpha)} + H(q^*(\mathbf{Z})) + \log C. \tag{C.2}$$

Hence, it is enough to compute the constant C. From (B.1), $C = \int P(\mathbf{G})\exp(E_{q^*(\mathbf{Z})}[\log P(Y|\mathbf{Z},\mathbf{G},\boldsymbol{\pi})])d\mathbf{G}$. For simplicity, define $C_0 := E_{q^*(\mathbf{Z})}[\log P(Y|\mathbf{Z},\mathbf{G},\boldsymbol{\pi})]$ (i.e, $C = E_{p(\mathbf{G})}[\exp(C_0)]$). We then have

$$C_0 = -\frac{ND}{2}\log(2\pi\sigma^2) - \frac{1}{2}\sum_{d,m=1}^{D,N}(Y_{(d)} - g_d^{(m)}(X))^\top B_m(Y_{(d)} - g_d^{(m)}(X)),$$

$$\exp(C_0) = \exp(-\frac{ND}{2}\log(2\pi\sigma^2))(2\pi)^{\frac{NDM}{2}}\prod_{d,m=1}^{D,M}\left(|B_m^{-1}|^{1/2}\mathcal{N}(Y_{(d)}|g_d^{(m)}(X),B_m^{-1})\right)$$

If we let $C_1 := \exp(-\frac{ND}{2}\log(2\pi\sigma^2))(2\pi)^{\frac{NDM}{2}}\prod_{d,m=1}^{D,M}|B_m^{-1}|^{1/2}$, then

$$C = C_1\int\prod_{d,m=1}^{D,M}\mathcal{N}(g_d^{(m)}(X)|\mathbf{0},K_m)\mathcal{N}(Y_{(d)}|g_d^{(m)}(X),B_m^{-1})d\mathbf{G} = C_1\prod_{d,m=1}^{D,M}\left(Z_{dm}\int\mathcal{N}(g_d^{(m)}|(K_m^{-1}+B_m)^{-1}B_mY_{(d)},(K_m^{-1}+B_m)^{-1})d\mathbf{G}\right) = \exp(-\frac{ND}{2}\log(2\pi\sigma^2))(2\pi)^{\frac{NDM}{2}}\prod_{d,m=1}^{D,M}\left(|B_m^{-1}|^{1/2}Z_{dm}\right)$$

where $Z_{dm} = \mathcal{N}(Y_{(d)}|\,\mathbf{0}, K_m + B_m^{-1})$. So,

$$\log C = -\frac{ND}{2}\log 2\pi\sigma^2 + \frac{NDM}{2}\log 2\pi - \frac{D}{2}\sum_{m=1}^{M}\log|B_m| + \sum_{d,m=1}^{D,M}\log Z_{dm}$$
$$= \frac{ND(M-1)}{2}\log 2\pi\sigma^2 - \frac{D}{2}\sum_{n,m=1}^{N,M}\log\phi_{nm} + \sum_{d,m=1}^{D,M}\log Z_{dm}.$$

By plugging the above $\log C$ into (C.2), we get $\tilde{\mathcal{L}}_E(Q^*) = \tilde{\mathcal{L}}_M(Q^*)$.□

Appendix D. Proof of Proposition 1

By applying the Weyl's inequality to $\Sigma^{(m)*} - K_m$ and K_m, we get

$$\lambda_N(K_m) + \lambda_N(\Sigma^{(m)*} - K_m) \le \lambda_N(\Sigma^{(m)*}) \le \lambda_N(K_m) + \lambda_1(\Sigma^{(m)*} - K_m)$$

which yields

$$\lambda_N(\Sigma^{(m)*} - K_m) \le \lambda_N(\Sigma^{(m)*}) - \lambda_N(K_m). \tag{D.1}$$

Similarly, by applying the Weyl's inequality to K_m^{-1} and B_m, we get

$$\lambda_1(K_m^{-1} + B_m) \ge \lambda_1(B_m) + \lambda_N(K_m^{-1}). \tag{D.2}$$

On the other hand, since we assume $\phi_{nm} > 0$ for all n and m, B_m is invertible. So, by using the Woodbury matrix formula on $\Sigma^{(m)*} = (K_m^{-1} + B_m)^{-1}$,

$$K_m - \Sigma^{(m)*} = K_m - K_m + K_m(B_m^{-1} + K_m)^{-1}K_m = K_m(B_m^{-1} + K_m)^{-1}K_m$$

which shows that $K_m - \Sigma_m$ is a positive definite matrix. By combining this with (D.1) and (D.2), we obtain

$$\begin{aligned}
&\|K_m - \Sigma^{(m)*}\|_2 \\
&= \lambda_1(K_m - \Sigma^{(m)*}) = -\lambda_N(\Sigma^{(m)*} - K_m) \\
&\ge \lambda_N(K_m) - \lambda_N(\Sigma^{(m)*}) = \lambda_N(K_m) - \frac{1}{\lambda_1(K_m^{-1} + B_m)} \\
&\ge \lambda_N(K_m) - \frac{1}{\lambda_N(K_m^{-1}) + \lambda_1(B_m)} = \lambda_N(K_m) - \frac{\sigma^2\lambda_1(K_m)}{\sigma^2 + \lambda_1(K_m)\phi_{*m}}.
\end{aligned}$$

□

Appendix E. Proof of Proposition 2

First observe that

$$P(y_d^*|\mathbf{x}^*, X, Y) = \sum_{m=1}^{M} P(y_d^*|\mathbf{x}^*, X, Y, z_{\mathbf{x}^*m} = 1)P(z_{\mathbf{x}^*m} = 1|\mathbf{x}^*, X, Y). \tag{E.1}$$

From $\boldsymbol{\pi}|X, Y \sim \mathrm{Dir}(\alpha + c_1, \ldots, \alpha + c_M)$, it follows that

$$\begin{aligned}
P(z_{\mathbf{x}^*m} = 1|\mathbf{x}^*, X, Y) &= \int P(z_{\mathbf{x}^*m} = 1|\mathbf{x}^*, X, Y, \boldsymbol{\pi})P(\boldsymbol{\pi}|\mathbf{x}^*, X, Y)\,d\boldsymbol{\pi} \\
&= \int \pi_m P(\boldsymbol{\pi}|X, Y)\,d\boldsymbol{\pi} = \frac{\alpha + c_m}{M\alpha + N} =: \hat{\pi}_m.
\end{aligned}$$

Next, by using $q^*(g_d^{(m)}|X, Y)$ which estimates unknown $P(g_d^{(m)}|X, Y)$, we get

$$\begin{aligned}
P(y_d^*|\mathbf{x}^*, X, Y, z_{\mathbf{x}^*m} = 1) &= \int P(y_d^*|\mathbf{x}^*, X, g_d^{(m)})P(g_d^{(m)}|X, Y)dg_d^{(m)} \\
&\simeq \int P(y_d^*|\mathbf{x}^*, X, g_d^{(m)})q^*(g_d^{(m)}|X, Y)dg_d^{(m)} \\
&= \mathcal{N}(y_d^*\,|\,\mu_d^{(m)*}(\mathbf{x}^*), \sigma_d^{2(m)*}(\mathbf{x}^*))
\end{aligned} \tag{E.2}$$

Note that the last equality in (E.2) is the result of the product of two Gaussian distributions in Petersen and Pedersen [10]. By combining (E.1) and (E.2), we obtain the above results.□

References

[1] C.K. Williams, C.E. Rasmussen, Gaussian Processes for Machine Learning, vol. 2, MIT Press Cambridge, MA, 2006.
[2] P. Li, S. Chen, Gaussian process approach for metric learning, Pattern Recognit. 87 (2019) 17–28.
[3] M. Kandemir, T. Kekeç, R. Yeniterzi, Supervising topic models with Gaussian processes, Pattern Recognit. 77 (2018) 226–236.
[4] P. Ruiz, P. Morales-Álvarez, R. Molina, A.K. Katsaggelos, Learning from crowds with variational Gaussian processes, Pattern Recognit. 88 (2019) 298–311.
[5] I.J. Cox, A review of statistical data association techniques for motion correspondence, Int. J. Comput. Vis. 10 (1) (1993) 53–66.
[6] M. Lázaro-Gredilla, S. Van Vaerenbergh, N.D. Lawrence, Overlapping mixtures of Gaussian processes for the data association problem, Pattern Recognit. 45 (4) (2012) 1386–1395.
[7] J. Hensman, M. Rattray, N.D. Lawrence, Fast nonparametric clustering of structured time-series, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2) (2014) 383–393.
[8] L.-L. Li, J. Sun, C.-H. Wang, Y.-T. Zhou, K.-P. Lin, Enhanced Gaussian process mixture model for short-term electric load forecasting, Inf. Sci. 477 (2019) 386–398.
[9] M. Kuusela, T. Raiko, A. Honkela, J. Karhunen, A gradient-based algorithm competitive with variational Bayesian em for mixture of Gaussians, in: 2009 International Joint Conference on Neural Networks, IEEE, 2009, pp. 1688–1695.
[10] K.B. Petersen, M.S. Pedersen, The matrix cookbook, Nov 2012, URL http://www2.imm.dtu.dk/pubdb/p.php 3274 (2012).
[11] R. Bellman, On the approximation of curves by line segments using dynamic programming, Commun. ACM 4 (6) (1961) 284.
[12] J. Martens, New insights and perspectives on the natural gradient method, arXiv preprint arXiv:1412.1193 (2014).
[13] T. Fortmann, Y. Bar-Shalom, M. Scheffe, Sonar tracking of multiple targets using joint probabilistic data association, IEEE J. Ocean. Eng. 8 (3) (1983) 173–184.
[14] S. Oh, Bayesian formulation of data association and Markov chain Monte Carlo data association, in: Robotics: Science and Systems Conference (RSS) Workshop Inside Data Association, vol. 2, Citeseer, 2008.
[15] C.J. Veenman, M.J. Reinders, E. Backer, Resolving motion correspondence for densely moving points, IEEE Trans. Pattern Anal. Mach. Intell. 23 (1) (2001) 54–72.
[16] L. Vaquero, V.M. Brea, M. Mucientes, Tracking more than 100 arbitrary objects at 25 fps through deep learning, Pattern Recognit. 121 (2022) 108205.
[17] K. Yoon, D.Y. Kim, Y.-C. Yoon, M. Jeon, Data association for multi-object tracking via deep neural networks, Sensors 19 (3) (2019) 559.
[18] C.-C. Lin, Y. Hung, R. Feris, L. He, Video instance segmentation tracking with a modified VAE architecture, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13147–13157.
[19] L. Yang, N.-M. Cheung, J. Li, J. Fang, Deep clustering by Gaussian mixture variational autoencoders with graph embedding, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6440–6449.
[20] V. Tresp, Mixtures of Gaussian processes, in: Advances in Neural Information Processing Systems, 2001, pp. 654–660.
[21] M. Lázaro-Gredilla, S. Van Vaerenbergh, A Gaussian process model for data association and a semidefinite programming solution, IEEE Trans. Neural Netw. Learn. Syst. 25 (11) (2014) 1967–1979.
[22] M. Kaiser, C. Otte, T.A. Runkler, C.H. Ek, Data association with Gaussian processes, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2019, pp. 548–564.
[23] H. Liu, Y.-S. Ong, X. Jiang, X. Wang, Modulating scalable Gaussian processes for expressive statistical learning, Pattern Recognit. 120 (2021) 108121.
[24] GPy, GPy: a Gaussian process framework in python, since 2012, (http://github.com/SheffieldML/GPy).
[25] D. Duvenaud, J. Lloyd, R. Grosse, J. Tenenbaum, G. Zoubin, Structure discovery in nonparametric regression through compositional kernel search, in: International Conference on Machine Learning, 2013, pp. 1166–1174.
[26] Z. Chen, B. Wang, How priors of initial hyperparameters affect Gaussian process regression models, Neurocomputing 275 (2018) 1702–1710.
[27] A.G. Wilson, Covariance Kernels for Fast Automatic Pattern Discovery and Extrapolation with Gaussian Processes, University of Cambridge, 2014 Ph.D. thesis.
[28] C.E. Rasmussen, H. Nickisch, Gaussian processes for machine learning (GPML) toolbox, J. Mach. Learn. Res. 11 (Nov) (2010) 3011–3015.
[29] C. Zhu, R.H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-BFGS-B: fortran subroutines for large-scale bound-constrained optimization, ACM Trans. Math. Softw. (TOMS) 23 (4) (1997) 550–560.
[30] R. Karlsson, F. Gustafsson, Monte Carlo data association for multiple target tracking, in: Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE, vol. 1, 2001, pp. 1–13.
[31] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, Y. Huang, T-drive: driving directions based on taxi trajectories, in: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2010, pp. 99–108.
[32] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, K. Schindler, Motchallenge 2015: towards a benchmark for multi-target tracking, arXiv preprint arXiv:1504.01942(2015).
[33] A. Milan, L. Leal-Taixé, I.D. Reid, S. Roth, K. Schindler, Mot16: a benchmark for multi-object tracking, arXiv abs/1603.00831(2016).
[34] J. Hensman, A. Matthews, Z. Ghahramani, Scalable variational Gaussian process classification, in: Artificial Intelligence and Statistics, PMLR, 2015, pp. 351–360.

[35] H. Liu, J. Cai, Y.-S. Ong, Remarks on multi-output Gaussian process regression, Knowledge-Based Syst. 144 (2018) 102–121.

Younghwan Jeon received the B.Sc. degree in mathematical sciences from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2017, currently pursuing the Ph.D. degree with the Department of Mathematical Sciences. His research interests include machine learning with Gaussian processes and the Bayesian inference.

Ganguk Hwang received his Ph.D. degree in Mathematics from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 1997. He has been with the Department of Mathematical Sciences, KAIST since 2002. His research interests include machine learning with Gaussian processes, kernel methods, and deep learning.