Query-Aware Bayesian Committee Machine for Scalable Gaussian Process Regression

Jiayuan He* Jianzhong Qi*† Kotagiri Ramamohanarao*

Abstract

The Gaussian process (GP) model is a powerful tool for regression problems. However, the high computational costs of the GP model has constrained its applications over large-scale data sets. To overcome this limitation, aggregation models employ distributed GP submodels (experts) for parallel training and predicting, and then merge the predictions of all submodels to produce an approximated result. The state-of-the-art aggregation models are based on Bayesian committee machines, where a prior is assumed at the start and then updated by each submodel. In this paper, we investigate the impact of the prior on the accuracy of aggregations. We propose a *query-aware Bayesian committee machine* (QBCM). The QBCM model partitions the testing data (i.e., queries) into subsets, and incorporates a query-aware prior when merging the predictions of submodels. This model improves the prediction accuracy, while retaining the advantages of aggregation models, i.e., closed-form inference and parallelizability. We conduct both theoretical analysis and empirical experiments on real data. The results confirm the effectiveness and efficiency of the proposed model QBCM.

Keywords Gaussian process, Aggregation methods, Bayesian committee machine

1 Introduction.

The *Gaussian process* (GP) [1, 2] is an important and powerful non-parametric learning model. This model has shown strong capability in various applications, e.g., regression [3], classification [4], optimization [5, 6], time-series data modelling [7], anomaly detection [8, 9], and visualization [10]. However, the standard GP model needs to compute a covariance matrix over the input data set of n data points. This leads to high computational and memory costs, i.e., $O(n^3)$ and $O(n^2)$, respectively. These high costs have hindered the applicability of GP to large-scale datasets.

To scale GP to large-scale datasets, approximation methods have been proposed [11, 12, 13, 14]. One stream of approximation methods are the inducing points methods which replace the full covariance matrix with a size-m ($m \ll n$) set of induced points [12, 13]. The computational complexity is then reduced to $O(nm^2)$ for training and $O(nm)$ for predicting. However, the inducing points methods reduce the *full GP* (i.e., performing GP based on the full covariance matrix over n data points) to a degenerate model where only m linearly independent functions are assumed. Another stream of methods exploit the existing structure of the covariance matrix for fast training and predicting [15]. Although efficient, these methods usually require the input of GP to conform a special distribution (e.g., grid structure), making them inapplicable for most real-world datasets.

This paper focuses on a third stream of GP approximation methods, i.e., *aggregation models* [16, 17, 18]. Such models provide a practical and parallelizable solution to scalable GP. Aggregation models partition the input dataset into subsets and allocate them to a set of submodels. When predicting, each submodel provides an independent prediction, which is then merged with the predictions of other submodels to form a final prediction. The aggregation criteria used in merging predictions is crucial to the accuracy of aggregation models. The state-of-the-art aggregation criteria are based on Bayesian committee machines (BCM) [17, 19, 20]. BCM uses a typical Bayesian framework, which assumes a prior distribution at the start, and then updates the posterior using the prediction of each submodel. Different priors have been used in the literature. For example, Tresp et al. [17] and Deisenroth et al. [19] use the original prior of testing points as the prior; Liu et al. [20] use the posterior of one submodel as the prior. These studies use the same prior for all testing points.

In this paper, we show that (i) using different prior for different testing points significantly improves the accuracy of the aggregation; and (ii) the best prior is query dependent. Then, we propose a efficient *Query-aware BCM* (QBCM) that improves the aggregation accuracy via incorporating a query-aware prior. Specifically, the proposed QBCM partition the testing points into dis-

*School of Computing and Information Systems, The University of Melbourne, Australia

†Corresponding author: jianzhong.qi@unimelb.edu.au

joint clusters. Then for each cluster of testing points, it uses the submodel that provides the most confident prediction as the prior, which is then updated by other submodels. The contributions of this paper are threefolds.

1. We perform a theoretical analysis on the impact of prior on the accuracy of aggregation models for scalable GP.
2. We propose a novel query-aware Bayesian committee machine (QBCM) that provides a more accurate approximation than the state-of-the-art aggregation models.
3. We conduct extensive experiments on both synthetic and real datasets. Experimental results show that our proposed QBCM model improves the prediction accuracy by up to 23.3% comparing with the state-of-the-art GP approximation models and is more robust to the growth in the number of submodels.

2 Preliminaries and related work.

We start with a brief review over Gaussian process regression and existing aggregation models for approximate Gaussian process.

2.1 Gaussian process. A Gaussian process is a collection of random variables such that any finite subset of these variables conform a joint Gaussian distribution. Let $\mathcal{D}$ be a dataset that consists of input feature vectors $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ and its corresponding real-valued targets $\boldsymbol{y} = \{y_1, \ldots, y_n\}$. Without loss of generality, we assume zero mean of the GP, then we can model a distribution over functions $f(\boldsymbol{x}) \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x'}))$, where any set of function values forms a joint Gaussian distribution characterized by the covariance mapping function $k(\cdot, \cdot)$. One of the most well-known covariance functions is the squared exponential (SE) covariance function:

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \lambda^2 \exp(-\frac{1}{2}\sum_{i=1}^{d}\frac{(x_i - x_i')^2}{l_i^2}) \tag{2.1}$$

where λ is the output scale amplitude, l_i is an input length-scale along the i-th dimension and d is data dimensionality. Assuming noisy observation $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \epsilon$ where the *i.i.d.* noise follows $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$, the predictive distribution for a testing point $\boldsymbol{x}_*$ conditioned on the dataset $\mathcal{D}$, i.e., $p(y_*|\mathcal{D}, \boldsymbol{x}_*)$, conforms the Gaussian distribution $\mathcal{N}(\mu(\boldsymbol{x}_*), \sigma^2(\boldsymbol{x}_*))$, where

$$\mu(\boldsymbol{x}_*) = \boldsymbol{k}_{f*}^\top [\boldsymbol{K}_{ff} + \sigma_\epsilon^2 \boldsymbol{I}]^{-1} \boldsymbol{y}_f \tag{2.2}$$

$$\sigma^2(\boldsymbol{x}_*) = \sigma_{**}^2 - \boldsymbol{k}_{f*}^\top [\boldsymbol{K}_{ff} + \sigma_\epsilon^2 \boldsymbol{I}]^{-1} \boldsymbol{k}_{f*} + \sigma_\epsilon^2 \tag{2.3}$$

Here, we use σ_{**}^2 to present $k(\boldsymbol{x}_*, \boldsymbol{x}_*)$, $\boldsymbol{y}_f$ to represent the full input target, $\boldsymbol{k}_{f*}$ to denote $k(\boldsymbol{x}_*, \boldsymbol{X})$, and $\boldsymbol{K}_{ff}$ to denote $k(\boldsymbol{X}, \boldsymbol{X})$ for simplicity.

2.2 Aggregation models. The aggregation models form an important family of approximate methods for scalable GP [16, 17, 19, 20, 21, 22]. In general, such models employ a set of submodels (GP experts), each of which is trained on a subset of the full training dataset. For a testing point to be inferred, an aggregation model produces the final prediction by aggregating the predictions from all submodels according to certain criterion.

Most aggregation models share the same training process, named *factorized training*. Assuming m submodels, an aggregation model divides the complete dataset $\mathcal{D}$ into m disjoint subsets, $\mathcal{D}_1, \ldots, \mathcal{D}_m$. Let $\mathcal{D}_i = \{\boldsymbol{X}_i, \boldsymbol{y}_i\}$ be the subset allocated to submodel $\mathcal{M}_i$. The marginal likelihood of $\mathcal{D}_i$, $p(\boldsymbol{y}_i|\theta, \boldsymbol{X}_i) = \mathcal{N}(0, \boldsymbol{K}_{ii} + \sigma_\epsilon^2 \boldsymbol{I}_i)$. Here, $\boldsymbol{K}_{ii}$ represents the covariance matrix of $\boldsymbol{X}_i$, $\boldsymbol{I}_i$ the order-n_i identity matrix, and n_i the number of instances in $\mathcal{D}_i$, respectively. The factorized training process assumes independence between submodels (zero covariance between the subsets). The exact marginal likelihood of $\mathcal{D}$ can be approximated by the product of the marginal likelihood of all submodels:

$$p(\boldsymbol{y}_f|\theta, \boldsymbol{X}_f) \approx \prod_{i=i}^{m} p(\boldsymbol{y}_i|\theta, \boldsymbol{X}_i) \tag{2.4}$$

where $\boldsymbol{y}_f$ and $\boldsymbol{X}_f$ represent the original input $\boldsymbol{y}$ and $\boldsymbol{X}$ in full GP. Thus, the factorized training approximates the training process as maximizing the summation of the log-likelihood functions of all submodels, where the hyperparameters θ for all submodels are assumed to be the same. Assuming equal partition of $\mathcal{D}$, the complexity of the training process is reduced to $O(m(\frac{n}{m})^3)$, which can be significantly lower than that of the full GP (i.e., $O(n^3)$) in practice.

To make predictions, aggregation models first use each submodel to generate a prediction based on its training dataset. Let $\mathcal{M}_i$ be a submodel and $\boldsymbol{x}_*$ be a testing point to be inferred. The predicted distribution of $\boldsymbol{x}_*$ produced by $\mathcal{M}_i$ has $p(y_*|\mathcal{D}_i, \boldsymbol{x}_*) = \mathcal{N}(\mu_i(\boldsymbol{x}_*), \sigma_i^2(\boldsymbol{x}_*))$ where

$$\mu_i(\boldsymbol{x}_*) = \boldsymbol{k}_{i*}^\top [\boldsymbol{K}_{ii} + \sigma_\epsilon^2 \boldsymbol{I}_i]^{-1} \boldsymbol{y}_i \tag{2.5}$$

$$\sigma_i^2(\boldsymbol{x}_*) = \sigma_{**}^2 - \boldsymbol{k}_{i*}^\top [\boldsymbol{K}_{ii} + \sigma_\epsilon^2 \boldsymbol{I}_i]^{-1} \boldsymbol{k}_{i*} + \sigma_\epsilon^2 \tag{2.6}$$

Here, $\boldsymbol{k}_{i*}$ represents the covariance vector of $\boldsymbol{x}_*$ and $\boldsymbol{X}_i$. The main difference among existing aggregation models lies in their criteria for combining the predictions provided by all submodels.

Next, we briefly discuss six state-of-the-art aggregation models: *Product-of-Experts* (PoE) [16] and its generalized form (GPoE) [21], *Bayesian committee machine* (BCM) [17], *robust Bayesian committee machine*

(RBCM) and its generalized form (GRBCM) [20], and *nested pointwise aggregation of experts* (NPAE) [22].

PoE models. PoE assumes that the final prediction of a testing point $\boldsymbol{x}_*$ is the product of the predictions of $\boldsymbol{x}_*$ provided by each submodel:

$$p(y_*|\mathcal{D}, \boldsymbol{x}_*) \approx \prod_{i=1}^{m} p^{\beta_i}(y_*|\mathcal{D}_i, \boldsymbol{x}_*) \tag{2.7}$$

Thus, the aggregated predicted mean and variance can be written as follows.

$$\mu_{poe}(\boldsymbol{x}_*) = \sigma^2_{poe}(\boldsymbol{x}_*) \sum_{i=1}^{m} \beta_i \sigma_i^{-2}(\boldsymbol{x}_*) \mu_i(\boldsymbol{x}_*) \tag{2.8}$$

$$\sigma^{-2}_{poe}(\boldsymbol{x}_*) = \sum_{i=1}^{m} \beta_i \sigma_i^{-2}(\boldsymbol{x}_*) \tag{2.9}$$

The original model PoE assumes $\beta_i = 1, \forall i \in [1, m]$. The generalized model GPoE introduces a varing β_i so as to weigh the importance of different submodels in aggregation. The weight β_i for submodel $\mathcal{M}_i$ is computed as the difference in the differential entropy between the prior of $\boldsymbol{x}_*$ and the posterior of $\boldsymbol{x}_*$ given $\mathcal{D}_i$, i.e., $\beta_i = 0.5(\log \sigma^2_{**} - \log \sigma^2_i(\boldsymbol{x}_*))$. Thus, the more informative submodel contributes more to the final prediction, while a distant submodel contributes little ($\beta_i \to 0$ as $\sigma^2_i \to \sigma^2_{**}$). PoE becomes over-confident (too small variance) if the number of submodels increases, e.g., $\sigma^2(\boldsymbol{x}_*) \to 0$ as $m \to \infty$. For GPoE, if $\boldsymbol{x}_*$ is far away from all submodels ($\sigma^2_i(\boldsymbol{x}_*) \to 0$), the prediction variance will not converge to its prior ($\sigma^2(\boldsymbol{x}_*) \to \infty$). As such, it has been suggested that $\beta_i = 1/m$ yields the best prediction result.

BCM models. Based on the Bayesian inference framework, BCM models incorporate the prior of $\boldsymbol{x}_*$ in its aggregation criteria. The original BCM model [17] assumes conditional independence of any two submodels given the target value of $\boldsymbol{x}_*$. Thus, the closed-form aggregated distribution $p(y_*|\mathcal{D}, \boldsymbol{x}_*)$ can be written as:

$$p(y_*|\mathcal{D}, \boldsymbol{x}_*) \approx \frac{\prod_{i=1}^{m} p^{\beta_i}(y_*|\mathcal{D}_i, \boldsymbol{x}_*)}{p^{\sum_i \beta_i - 1}(y_*|\boldsymbol{x}_*)} \tag{2.10}$$

The original BCM model assumes $\beta_i = 1$ for all submodels, whereas its variant, RBCM [19], varies β_i to weigh the importance of submodels. However, both models have been shown to produce over-confident predictions when the data area is dense ($n \to \infty$) [20]. To address this issue, Liu et al. [20] proposes GRBCM. GRBCM uses a global submodel (prior submodel) $\mathcal{M}_g$ and then enhances each other submodel ($\mathcal{M}_i$) by combining $\mathcal{D}_i$ and $\mathcal{D}_g$. The predictions of these enhanced submodels are then merged assuming conditional independence of the non-global submodels given $\mathcal{M}_g$ and y_*. Thus, the aggregated distribution of $\boldsymbol{x}_*$ in GRBCM can be written as:

$$p(y_*|\mathcal{D}, \boldsymbol{x}_*) \approx \frac{\prod_{i=1}^{m-1} p^{\beta_i}(y_*|\mathcal{D}_i, \mathcal{D}_g, \boldsymbol{x}_*)}{p^{\sum_i \beta_i - 1}(y_*|\mathcal{D}_g, \boldsymbol{x}_*)} \tag{2.11}$$

NPAE model. Rullière et al. [22] propose a nested pointwise aggregation experts (NPAE) model, which exploits a hierarchical structure. In NPAE, each node aggregates the predictions of its child nodes, and the root node makes the final prediction. Suppose $v_1, \ldots, v_k$ are the k child nodes of v_p. To aggregate the predictions of child nodes, v_p treats their predictions as well as the testing point $\boldsymbol{x}_*$ as random variables that conform a nested GP. Although NPAE incorporates the dependency among the submodels when producing the final prediction, it requires a different GP for every testing point, leading to expensive prediction costs. Assuming a two-layer structure with $\sqrt{n}$ submodels and n' testing points, its time complexity for prediction is $O(n^2 n')$.

3 Query-aware Bayesian committee machine.

The key advantage of BCM models over PoE models is the incorporation of prior knowledge of testing points into the aggregation process. In this section, we first investigate how the prediction accuracy of BCM models is affected by the choice of priors (Section 3.1). Then, we introduce our query-aware BCM (QBCM) model (Section 3.2).

3.1 Impact of prior. We use the relative entropy (Kullback-Leibler divergence) as the metric to analyze the deviation of an aggregation model from the full GP. Given two probability distributions $p_1(x)$ and $p_2(x)$, the relative entropy of p_1 w.r.t. p_2 is formulated as:

$$\boldsymbol{D}_{KL} = -\int_x p_2(x) \log \frac{p_1(x)}{p_2(x)} dx \tag{3.12}$$

Usually, $\boldsymbol{D}_{KL} \to 0$ if $p_1(x)$ and $p_2(x)$ have very similar distributions, while $\boldsymbol{D}_{KL} \to \infty$ if $p_1(x)$ and $p_2(x)$ behave in such a different manner that the expectation given $p_1(x)$ becomes zero. Note that $\boldsymbol{D}_{KL} \geqslant 0$ for any two distributions.

Let $\boldsymbol{x}_*$ be a testing point, an effective aggregation model should yield a posterior distribution of $\boldsymbol{x}_*$ that has small deviation from the posterior of full GP ($\boldsymbol{D}_{KL} \to 0$). In what follows, we measure the approximation effectiveness of an aggregation model using the relative entropy of its predicted posterior distribution for $\boldsymbol{x}_*$ w.r.t. the posterior distribution produced by a full GP.

Let $\mathcal{D}$ be a training dataset that is partitioned into m submodels. Based on Eqs. 2.11 and 3.12, the relative entropy of a GRBCM model can be formulated as:

$$\boldsymbol{D}_{KL} = -\int_{y_*} p(y_*|\mathcal{D}, \boldsymbol{x}_*) \log \frac{\prod_{i=1}^{m} p^{\beta_i}(y_*|\mathcal{D}_i, \mathcal{D}_g, \boldsymbol{x}_*)}{p(y_*|\mathcal{D}, \boldsymbol{x}_*) p^{\sum_i \beta_i - 1}(y_*|\mathcal{D}_g, \boldsymbol{x}_*)} dy_* \tag{3.13}$$

where $p(y_*|\mathcal{D}, \boldsymbol{x}_*)$ is the posterior distribution of full GP. Submodel $\mathcal{D}_g$ is the global submodel (prior submodel) that is used to enhance other submodels. The above equation can be rewritten as:

(3.14)
$$\boldsymbol{D}_{KL} = -\int_{y_*} p(y_*|\mathcal{D}, \boldsymbol{x}_*) \log \Big(\frac{p^{\sum_i \beta_i - 1}(y_*|\mathcal{D}, \boldsymbol{x}_*)}{p^{\sum_i \beta_i - 1}(y_*|\mathcal{D}_g, \boldsymbol{x}_*)} \cdot \prod_{i=1}^{m} \frac{p^{\beta_i}(y_*|\mathcal{D}_i, \mathcal{D}_g, \boldsymbol{x}_*)}{p^{\beta_i}(y_*|\mathcal{D}, \boldsymbol{x}_*)} \Big) dy_*$$

Expanding the logarithmic term with the product rule, the above equation becomes:

$$\boldsymbol{D}_{KL} = \sum_{i=1}^{m} \beta_i \boldsymbol{D}_{KL}^{\mathcal{M}_{i,g}} + (1 - \sum_{i=1}^{m} \beta_i) \boldsymbol{D}_{KL}^{\mathcal{M}_g} \tag{3.15}$$

where $\boldsymbol{D}_{KL}^{\mathcal{M}_g}$ represents the relative entropy of the prediction produced by submodel of $\mathcal{M}_g$ w.r.t. that of full GP and $\boldsymbol{D}_{KL}^{\mathcal{M}_{i,g}}$ represents the relative entropy of the posterior distribution conditioned on $\mathcal{D}_i \cap \mathcal{D}_g$ w.r.t. that of full GP. For ease of manipulation, we rewrite the above equation as:

$$\boldsymbol{D}_{KL} = \sum_{i=1}^{m} \beta_i (\boldsymbol{D}_{KL}^{\mathcal{M}_{i,g}} - \boldsymbol{D}_{KL}^{\mathcal{M}_g}) + \boldsymbol{D}_{KL}^{\mathcal{M}_g} \tag{3.16}$$

One important characteristic of GP is that if the data allocated with $\mathcal{M}_i$ is a subset of the data with $\mathcal{M}_j$ ($\mathcal{D}_i \subset \mathcal{D}_j$), then $\mathcal{M}_j$ is a more informative submodel than $\mathcal{M}_i$. This leads to a more accurate prediction of $\mathcal{M}_j$ than $\mathcal{M}_i$ ($\boldsymbol{D}_{KL}^{\mathcal{M}_j} < \boldsymbol{D}_{KL}^{\mathcal{M}_i}$), and in the meantime, a more confident prediction of $\mathcal{M}_j$ than $\mathcal{M}_i$ ($\sigma_j^2 < \sigma_i^2$). Here, σ_j^2 represents the predicted variance of $\boldsymbol{x}_*$ by $\mathcal{M}_j$. Thus, in the above equation, we have $\boldsymbol{D}_{KL}^{\mathcal{M}_{i,g}} - \boldsymbol{D}_{KL}^{\mathcal{M}_g} < 0$, since $\mathcal{M}_{i,g}$ is trained on $\mathcal{D}_i \cap \mathcal{D}_g$. Next, we show how the global submodel $\mathcal{M}_g$ affects the accuracy of the aggregation model. Specifically, we assume two cases: (i) Among all submodels, at least one submodel (e.g., $\mathcal{M}_1$) is sufficiently close to $\boldsymbol{x}_*$ such that $\boldsymbol{D}_{KL}^{\mathcal{M}_1} \to 0$; and (ii) All submodels are a weaker model than the full GP such that $\boldsymbol{D}_{KL}^{\mathcal{M}_i} > 0, \forall i \in [1, M]$.

In Case (i), consider two aggregation models, Model 1 and Model 2. Model 1 uses $\mathcal{M}_1$ as the global submodel, i.e., $\mathcal{M}_g = \mathcal{M}_1$. Model 2 uses a weaker submodel (e.g., $\mathcal{M}_2$ such that $\boldsymbol{D}_{KL}^{\mathcal{M}_2} > 0$) as the global submodel, i.e., $\mathcal{M}_g = \mathcal{M}_2$. We formulate the relative entropies of the two models, $\boldsymbol{D}_{KL}^1$ and $\boldsymbol{D}_{KL}^2$, as follows.

$$\boldsymbol{D}_{KL}^1 = \sum_{i \neq 1} \beta_{i,1} (\boldsymbol{D}_{KL}^{\mathcal{M}_{i,1}} - \boldsymbol{D}_{KL}^{\mathcal{M}_1}) + \boldsymbol{D}_{KL}^{\mathcal{M}_1} \tag{3.17}$$

$$\boldsymbol{D}_{KL}^2 = \sum_{i \neq 2} \beta_{i,2} (\boldsymbol{D}_{KL}^{\mathcal{M}_{i,2}} - \boldsymbol{D}_{KL}^{\mathcal{M}_2}) + \boldsymbol{D}_{KL}^{\mathcal{M}_2} \tag{3.18}$$

We use $\beta_{i,1}$ and $\beta_{i,2}$ to differentiate the values of β_i when $\mathcal{M}_1$ and $\mathcal{M}_2$ are used as global submodel, respectively. Since $\mathcal{M}_{i,1}$ is more informative than $\mathcal{M}_1$, we have $\boldsymbol{D}_{KL}^{\mathcal{M}_{i,1}} \to 0$ as $\boldsymbol{D}_{KL}^{\mathcal{M}_1} \to 0$. Based on Eq. 3.17, we have $\boldsymbol{D}_{KL}^1 \to 0$, which means that Model 1 is as accurate as full GP.

For Model 2, we have $\boldsymbol{D}_{KL}^{\mathcal{M}_{1,2}} \to 0$, since $\boldsymbol{D}_{KL}^{\mathcal{M}_1} \to 0$. Thus, Eq. 3.18 can be rewritten as:

$$\boldsymbol{D}_{KL}^2 = \sum_{i=3}^{m} \beta_{i,2} (\boldsymbol{D}_{KL}^{\mathcal{M}_{i,2}} - \boldsymbol{D}_{KL}^{\mathcal{M}_2}) + (1 - \beta_{1,2}) \boldsymbol{D}_{KL}^{\mathcal{M}_2} \tag{3.19}$$

Since $\beta_{i,2} = 0.5(\log \sigma_2^2 - \log \sigma_{i,2}^2)$ and $\sigma_2^2 \geqslant \sigma_{i,2}^2$, we have $\beta_{i,2} \geqslant 0$, where the equality only holds when $\sigma_2^2 = \sigma_{i,2}^2$. Meanwhile, we have $\boldsymbol{D}_{KL}^{\mathcal{M}_{i,2}} \leqslant \boldsymbol{D}_{KL}^{\mathcal{M}_2}$ since submodel $\mathcal{M}_{i,2}$ is a more informative submodel than $\mathcal{M}_2$. Therefore, the first term of Eq. 3.19 is always no larger than 0. Next, we consider two scenarios: (1) $\beta_{1,2} > 1$; and (2) $\beta_{1,2} \leqslant 1$. The first scenario represents that $\mathcal{M}_2$ is a much weaker model than $\mathcal{M}_1$ ($\beta_{1,2} = \log \frac{\sigma_2}{\sigma_{1,2}} > 1$), while the second scenario represents that $\mathcal{M}_2$ is only slightly weaker than $\mathcal{M}_1$ ($\beta_{1,2} = \log \frac{\sigma_2}{\sigma_{1,2}} \leqslant 1$). In the first scenario, since $\boldsymbol{D}_{KL}^{\mathcal{M}_2} > 0$, we have $\boldsymbol{D}_{KL}^2 < 0$. In the second scenario, a prerequisite for $\boldsymbol{D}_{KL}^2 = 0$ is at least one of the two following conditions is satisfied: (1) $\beta_{1,2} = 1$, which means $\sigma_2 = 2 \cdot \sigma_{1,2}$; and (2) $\boldsymbol{D}_{KL}^{\mathcal{M}_2} = \frac{1}{\beta_{1,2}-1} [\sum_{i=3}^{m} \beta_{i,2} (\boldsymbol{D}_{KL}^{\mathcal{M}_{i,2}} - \boldsymbol{D}_{KL}^{\mathcal{M}_2})]$. Both conditions are very hard to be satisfied in real applications.

The above analysis proves that Model 1 is superior to Model 2. One important observation is that the relative entropy of Model 2 goes negative given $\beta_{1,2} > 1$. Although the relative entropy for any two distributions is presumed to be non-negative, here we are investigating the relative entropy of an aggregation of multiple distributions to the predicted distribution of full GP. In fact, this negativity reflects that Model 2 makes too aggressive aggregation in such scenario, i.e., over-estimating the contributions (e.g., $\beta_{1,2} > 1$) of informative submodels (e.g., $\boldsymbol{D}_{KL}^{\mathcal{M}_1} = 0$), when the prior submodel is imperfect (e.g., $\boldsymbol{D}_{KL}^{\mathcal{M}_2} > 0$). This finding is a generalized form of that in [20]. In [20],

Liu et al. show that when all submodels make perfect predictions ($\forall i \in [1,m], \sigma_i^2 \to \sigma_f^2$), the predicted variance of (R)BCM becomes zero ($\sigma^2_{(r)bcm} \to 0$) even if $\sigma_f^2 > 0$, where σ_f^2 represents the predicted variance of full GP. Here, we prove that if there is at least one submodel that makes approximately perfect prediction ($\exists i \in [1,m], \sigma_i^2 \to \sigma_f^2$) and if the global submodel that serves as the prior submodel is a much weaker submodel (e.g., $\beta_{1,2} > 1$), BCM models produce over-confident predictions ($\sigma^2_{(gr)bcm} < \sigma_f^2$).

In Case (ii) where each submodel is a weaker model than the full GP, no submodel is able to make perfect prediction solely on its own. Thus, BCM models need to combine the information provided by more than one submodels to make accurate predictions. BCM models assume conditional independence of submodels given the global submodel and the test input. This assumption, on one hand, reduces the overall computation cost, while on the other hand, incurs error as it does not consider the conditional covariance of non-global submodels. Next, we examine how this assumption impact the result prediction, based on which, we propose a low-cost solution to reduce the impact.

Let $\boldsymbol{X}_g$ be the input of the global submodel and $\boldsymbol{X}_{g'}$ be $[\boldsymbol{X}_g, \boldsymbol{x}_*]$. The covariance matrix for $\boldsymbol{X}_{g'}$ can be written as:

$$\boldsymbol{K}_{g'g'} = \begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{k}_{1*} \\ \boldsymbol{k}_{1*}^\top & \boldsymbol{k}_{**} \end{bmatrix} \tag{3.20}$$

The covariance of two non-global submodels $\mathcal{M}_i$ and $\mathcal{M}_j$ given y_* and $\mathcal{M}_g$ can be written as:

$$cov(\mathcal{M}_i, \mathcal{M}_j | \mathcal{M}_g, y_*) = \boldsymbol{K}_{ij} - \boldsymbol{k}_{ig'} \boldsymbol{K}^{-1}_{g'g'} \boldsymbol{k}^\top_{jg'} \tag{3.21}$$

The conditional independence assumption of $\mathcal{M}_i \perp \mathcal{M}_j | \mathcal{M}_g, \boldsymbol{y}_*$ results in $cov(\mathcal{M}_i, \mathcal{M}_j | \mathcal{M}_g, y_*) = 0$. This further leads to $\boldsymbol{K}_{ij} = \boldsymbol{k}_{ig'} \boldsymbol{K}^{-1}_{g'g'} \boldsymbol{k}^\top_{jg'}$. Without loss of generality, let $\mathcal{M}_1$ be the global submodel and $\mathcal{M}_2$ to $\mathcal{M}_m$ be the set of non-global submodels, we have

$$\boldsymbol{K}'_{ff} = \boldsymbol{K}_{ff} + \Delta \tag{3.22}$$

where $\boldsymbol{K}_{ff}$ represents the original covariance matrix for all training data and $\boldsymbol{K}'_{ff}$ represents the covariance matrix with the conditional independence assumption. Term Δ represents the change in covariance matrix:

$$\Delta = \begin{bmatrix} \Delta_{1,1} & \cdots & \Delta_{1,m} \\ \vdots & \ddots & \vdots \\ \Delta_{m,1} & \cdots & \Delta_{m,m} \end{bmatrix} \tag{3.23}$$

where

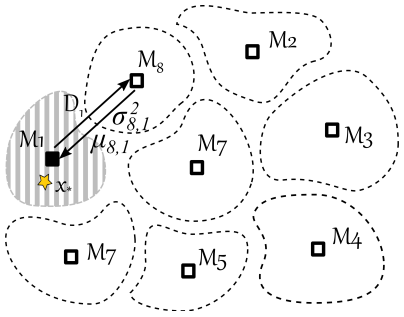


Figure 1: Illustration of QBCM model

$$\Delta_{i,j} = \begin{cases} 0 & i = j \text{ or } i = 1 \text{ or } j = 1 \\ \boldsymbol{k}_{ig'} \boldsymbol{K}^{-1}_{g'g'} \boldsymbol{k}^\top_{jg'} - \boldsymbol{K}_{ij} & \text{otherwise} \end{cases} \tag{3.24}$$

Here, $\Delta_{i,j} = 0$ if $i = j$, since the internal covariance is captured by each submodel. Similarly, $\Delta_{i,j} = 0$ if $i = 1$ or $j = 1$, since the external covariances of the global submodel with every non-global submodel are captured in GRBCM. Finally, the predicted mean and variance for $\boldsymbol{x}_*$ can be given as:

$$\mu(\boldsymbol{x}_*) = \boldsymbol{k}_{f*}^\top [\boldsymbol{K}_{ff} + \Delta + \sigma_\epsilon^2 I]^{-1} \boldsymbol{y}_f \tag{3.25}$$

$$\sigma^2(\boldsymbol{x}_*) = \sigma_{**}^2 - \boldsymbol{k}_{f*}^\top [\boldsymbol{K}_{ff} + \Delta + \sigma_\epsilon^2 I]^{-1} \boldsymbol{k}_{f*} \tag{3.26}$$

Comparing Eqs. 3.25 and 3.26 with Eqs. 2.2 and 2.3, we can see that error induced by the conditional independence assumption is proportional to (1) the actual external covariance between two non-global submodels (differences in predicted mean and variance increases as $|\Delta_{i,j}|$ increases); and (2) the covariance between $\boldsymbol{x}_*$ and the non-global submodels (differences in predicted mean and variance increases as $|\boldsymbol{k}_{i*}|$ increases). Based on these two findings, we propose to reduce the error induced by the conditional independence assumption by: (1) using locality-aware clustering (e.g., *kmeans*) to partition the training dataset so as to reduce the external covariances between submodels; and (2) using the submodel with the highest covariance with $\boldsymbol{x}_*$ as the global submodel so as to reduce the covariances of $\boldsymbol{x}_*$ with non-global submodels.

3.2 QBCM model. The above discussions show that (i) BCM models make over-confident predictions if one submodel is able to make an accurate prediction; and (ii) if each submodel is a significantly weaker model than full GP, using the submodel with the most confident prediction as the prior submodel improves the aggregation accuracy. Both findings show that the prior submodel, which yields the best aggregation accuracy,

is the submodel that provides the most confident prediction for the testing point among all submodels. Thus, we propose a query-aware BCM model, which chooses the best prior submodel for testing points adaptively.

We depict the QBCM model in Fig. 1. To construct the submodels, we partition the training data into equisized subsets and allocate each submodel (e.g., $\mathcal{M}_1$ to $\mathcal{M}_8$ in the figure) with a subset (the space enclosed by dashed circles). The partition is done using a locality-aware clustering algorithm (e.g., *kmeans*). We then store the geometric centers of all submodels in $\boldsymbol{c}$. We also add these centres to all submodels. The training process of QBCM follows the factorized training adopted by all aggregation models as discussed in Section 2.2. In the stage of prediction, let $\boldsymbol{X}_*$ be the set of testing points. We first allocate the testing points into different clusters: each testing point is allocated to its nearest submodel. This can be completed by comparing each testing point with each geometric center stored in $\boldsymbol{c}$. Suppose $\boldsymbol{X}_*^1$ is a cluster of testing points allocated to $\mathcal{M}_1$. To make predictions of $\boldsymbol{X}_*^1$, we send the allocated training data in $\mathcal{M}_1$ (i.e., $\mathcal{D}_1$) to all the other submodels. Then, each submodel (e.g., $\mathcal{M}_8$) provides an independent prediction based on its allocated data and the prior dataset: $\mathcal{M}_1 \cap \mathcal{M}_8$. The predicted results from each submodel (e.g., $\mu_{8,1}$ and $\sigma_{8,1}^2$) are sent back to $\mathcal{M}_1$ for aggregation. Based on Bayesian committee machine, we assume conditional independence of the submodels given y_i^* and $\mathcal{M}_i$. Let $\boldsymbol{x}_*$ be a testing point in $\boldsymbol{X}_*^1$, we get the final prediction for $\boldsymbol{x}_*$ as:

$$p(y_*|\mathcal{D}, \boldsymbol{x}_*) \approx \frac{\prod_{j\neq 1} p^{\beta_{j,1}}(y_*|\mathcal{D}_1, \mathcal{D}_j, \boldsymbol{x}_*)}{p^{\sum_{j\neq 1}\beta_{j,1}-1}(y_*|\mathcal{D}_1, \boldsymbol{x}_*)} \tag{3.27}$$

In section 3.1, we have shown that QBCM is able to provide more accurate result with the query-aware prior selection. Next, we show that QBCM will also outperform PoE and GPoE for completeness. Based on Eq. 2.7, we can write the relative entropy of (G)PoE as

$$\boldsymbol{D}_{KL}^{poe} = -\int_{y_*} p(y_*|\mathcal{D}, \boldsymbol{x}_*) \log \frac{\prod_{i=1}^m p^{\beta_i}(y_*|\mathcal{D}_i, \boldsymbol{x}_*)}{p(y_*|\mathcal{D}, \boldsymbol{x}_*)} dy_* \tag{3.28}$$

The above equation can be further written as

$$\boldsymbol{D}_{KL}^{poe} = -\int_{y_*} p(y_*|\mathcal{D}, \boldsymbol{x}_*) \log \Big(p^{\sum_i \beta_i - 1}(y_*|\mathcal{D}_i, \boldsymbol{x}_*) \frac{\prod_{i=1}^m p^{\beta_i}(y_*|\mathcal{D}_i, \boldsymbol{x}_*)}{p^{\sum_i \beta_i}(y_*|\mathcal{D}, \boldsymbol{x}_*)} \Big) dy_* \tag{3.29}$$

Expanding the logarithmic expression, the relative entropy of (G)PoE can be rewritten as

$$\boldsymbol{D}_{KL}^{poe} = \sum_{i=1}^m \beta_i \boldsymbol{D}_{KL}^{\mathcal{M}_i} + (\sum_{i=1}^m \beta_i - 1)\boldsymbol{H}(y_*|\mathcal{D}) \tag{3.30}$$

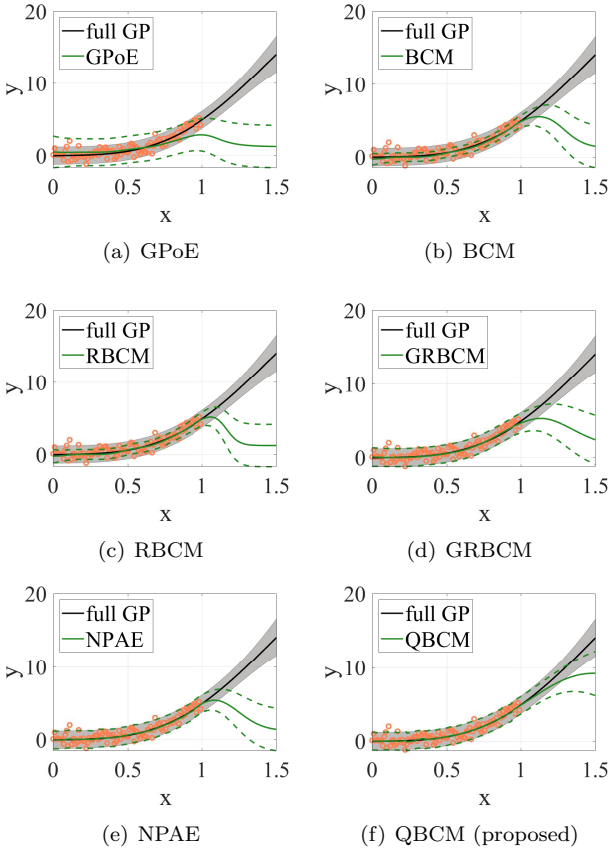
where $\boldsymbol{H}(y_*|\mathcal{D})$ represents the information entropy of y_* given the entire dataset $\mathcal{D}$. Since $\boldsymbol{H}(y_*|\mathcal{D}) \geqslant 0$ and $\boldsymbol{D}_{KL}^{\mathcal{M}_i} \geqslant 0$, PoE is unable to make correct predictions since it assumes $\beta_i = 1, \forall i \in [1, m]$. GPoE assumes $\beta_i = 1/m$ which cancels $\boldsymbol{H}(y_*|\mathcal{D})$ in Eq. 3.30. However, it can only make correct predictions when every submodel is perfect, i.e., $\boldsymbol{D}_{KL}^{\mathcal{M}_i} = 0, \forall i \in [1, m]$. Thus, QBCM is a more accurate model than both PoE and GPoE.

3.3 Complexity analysis. Suppose a QBCM model with n input training points, n' testing points, and m submodels. Assuming equisized partition of the training points, let s ($s = \lceil n/m \rceil$) be the number of training points on each submodel. The training process of QBCM requires $O(ns^2)$ time and $O(ns)$ memory. When making predictions, QBCM first clusters the testing points, which requires $O(n'm)$. The process for all submodels to make predictions requires $O(8ns^2 - 7s^3)$ (one inverse of the prior submodel requires $O(s^3)$ and $(m-1)$ inverses of the other submodels requires $O(8(m-1)s^3)$). We assume that the testing points are equally partitioned into k subsets. Then, the process of aggregating predictions for one cluster of testing points requires $O((4ns - 3s^2)n'/k)$. Thus, the overall predicting cost is $O(k(8ns^2 - 7s^3) + (4ns - 3s^2)n')$. Note that the proposed QBCM model allows parallel computation which can further reduce the response time. If each submodel is assigned to a distributed machine, the training cost per machine can be reduced to $O(s^3)$. The predicting cost per machine can be reduced to $O(8ks^2 + 4n's^2)$. Compared to BCM, RBCM, and GRBCM, QBCM has equal training cost and slightly higher predicting cost. Compared to NPAE, QBCM has equal training cost but is much more efficient as NPAE needs to perform a nested GP for each testing point.

3.4 A numerical example. We use a one-dimensional example to show the characteristics of the aggregation models, where

$$f(x) = 5x^3 + \epsilon \tag{3.31}$$

where the $i.i.d$ noise $\epsilon \sim \mathcal{N}(0, 0.36)$. We generate 2×10^3 training points within the range of $[0, 1]$, and 5×10^3 testing points within the range of $[0, 1.5]$, respectively. We compare the proposed model, QBCM, with aggregation models discussed in Section 2.2. For all compared models except GRBCM, we partition the training dataset into 16 subsets using the k-means algorithm. In GRBCM, we obtain the global subset using random sampling and the remaining subsets using k-means [20]. We follow previous studies [19] and [20] to set β_i in GPoE as $1/m$ where $m = 16$.



(a) GPoE (b) BCM (c) RBCM (d) GRBCM (e) NPAE (f) QBCM (proposed)

Figure 2: A numerical example of aggregation models.

4 Experiments and results.

In this section, we first use an example to illustrate the characteristics of the aggregation models discussed and the proposed model. Then we conduct experiments on real datasets to evaluate the accuracy and efficiency of the proposed model. We implement all codes in Matlab and perform the experiments on a workstation with 27 Intel (R) Xeon (R) E5-2697 2.60 GHz CPUs. We implement the submodels using the GPML toolbox [1] with the SE kernel as shown in Eq. 2.1. We optimize the hyperparameters using the conjugate gradients code, where the maximum iterations is 100.

We present the results in Fig. 2. We use the coral-colored circles to represent the training data points, the black solid lines and the grey regions to represent the mean predictions and 95% confidence intervals of full GP, the green solid lines and the green dashed lines to represent the mean predictions and 95% confidence intervals of aggregation models. For clarity of presentation, we downsample the training data by 20 times. Due to the page limit, we omit the result of PoE in this figure since GPoE has been shown to outperform PoE. In terms of variance prediction, GPoE makes conservative predictions (too large variance) all the time. BCM and RBCM make over-confident predictions (too small variance) compared with full GP. GRBCM, NPAE, and QBCM are all able to make correct variance prediction when the testing data is tightly enclosed by training data. However, GRBCM and NPAE make conservative predictions as soon as the testing data leaves the training data area. In terms of mean prediction, GPoE has the worst performance. All BCM models and NPAE can predict the correct mean values in the training data area. However, when the testing data leaves the training data area, the mean predictions of BCM and RBCM fall quickly to the prior of the model. GRBCM and NPAE are slightly better, but are still significantly outperformed by QBCM. In particular, the predicted mean values for testing point at $x = 1.5$ produced by GRBCM, NPAE, and QBCM are 3.17, 2.82, and 9.22, respectively. This confirms that the proposed model QBCM can make more accurate predictions even if no submodel makes perfect prediction.

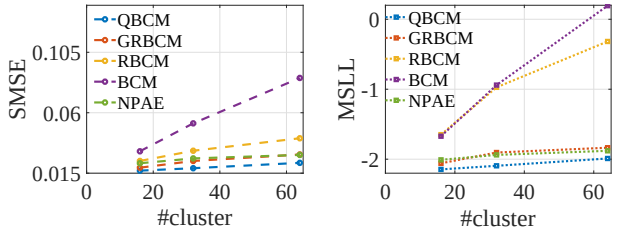
4.1 Experiments on real datasets. We conduct experiments on three real datasets: KIN8NM [23, 24] (8 dimensions, 8,192 data points), KIN40K [20, 18] (8 dimensions, 1×10^4 training points, and 3×10^4 testing points), POL [18] (26 dimensions, 1×10^4 training points, and 5×10^3 testing points). For dataset KIN8NM, we use random sampling to split the dataset into two partitions for training and testing, respectively. For all other datasets, we use the original split of the datasets. We compare the proposed model QBCM with aggregation models discussed in Section 2.2 and full GP. In addition, we compare the proposed model with the state-of-the-art inducing points method: fully independent training conditional (FITC) approximation [12]. We follow [20] and set the number of induced points of FITC as 10% of the total training points in all three datasets. For aggregation models, we set the number of submodels as 16, 32, and 64 on KIN40K, 4, 8, and 16 on KIN8NM, and 16, 32, and 64 on POL, repsectively. To evaluate the accuracy of the compared models, we adopt two metrics, namely standardized mean square error (SMSE) and mean standardized log loss (MSLL) [1], which represent the accuracy of mean prediction and the accuracy of variance prediction, respectively. To evaluate the efficiency of the models, we record their run time for training and predicting, respectively. We present the results of QBCM, GRBCM, NPAE and FITC on KIN40K and POL in Table 1, where the best result is highlighted in **bold**. We put the result of full GP on the first line as reference. In this table, we show the results of aggregation models where the numbers of

[1] http://www.gaussianprocess.org/gpml/code/matlab/doc/

submodels are set as 16 on KIN40K and POL. Results of these models with different numbers of submodels are shown in Fig. 3(a)- 4(b). The complete experiment results are provided in supplementary information, including all the results (G)PoE and (R)BCM and the results on dataset KIN8NM. The results in Table 1 confirm that QBCM is able to provide better approximation of full GP compared to the state-of-the-art models. On all datasets, QBCM consistently outperform all baseline models in both SMSE and MSLL. In particular, compared to GRBCM, QBCM reduces SMSE by 11.5%, and 23.3%. In terms of variance prediction, QBCM outperforms GRBCM by 4.3%, and 6%.

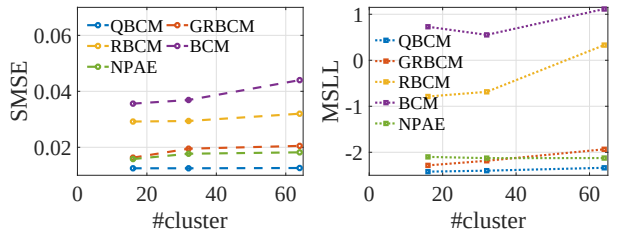
Table 1: Comparison in SMSE and MSLL

Dataset	KIN40K		POL	
Model	SMSE	MSLL	SMSE	MSLL
full GP	0.0117	-2.3557	0.0121	-2.5228
QBCM	**0.0169**	**-2.1458**	**0.0125**	**-2.4209**
GRBCM [20]	0.0191	-2.0586	0.0163	-2.2846
NPAE [22]	0.0224	-2.0093	0.0182	-2.1002
FITC [12]	0.0543	-1.575	0.0319	-2.1452



(a) SMSE (b) MSLL

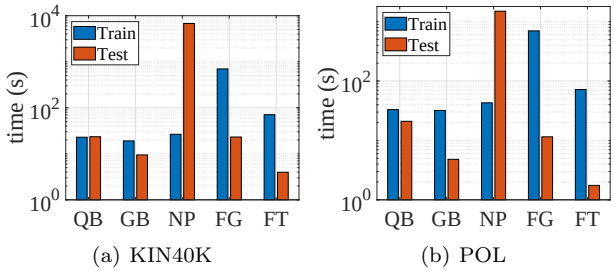
Figure 3: Comparison of accuracy by varying number of clusters on KIN40K



(a) SMSE (b) MSLL

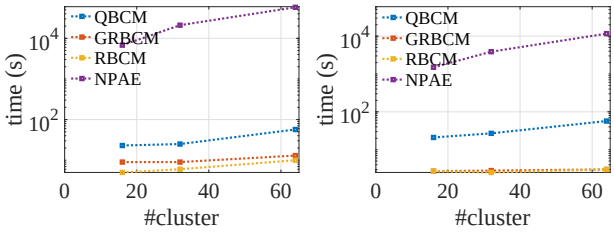
Figure 4: Comparison of accuracy by varying number of clusters on POL

We compare the robustness of the aggregation models by increasing the number of clusters in Fig. 3 and 4. The results on both datasets show that QBCM is the most robust than all baseline models. The accuracy degrades very slowly with the increase in the number of clusters. NPAE and GRBCM are very close in their accuracies, while NPAE is slightly more robust than GRBCM. BCM has the worst performance, due to its constant β_i in its aggregation process.



(a) KIN40K (b) POL

Figure 5: Comparison of training and predicting time



(a) KIN40K (b) POL

Figure 6: Comparison of predicting time by varying number of clusters

We compare the time cost for training and predicting between aggregation models, FITC, and full GP in Fig. 5, where QB, GB, NP, FG and FT represent QBCM, GRBCM, NPAE, Full GP and FITC, respectively. In terms of training cost, full GP and FT have much higher time cost than aggregation models, note the log scale of the figure. Benefiting from factorized training, all aggregation models require much less training time. In terms of predicting cost, NPAE is most expensive. QBCM requires much less time than NPAE in predicting, but has slightly more cost than GRBCM. FITC is most efficient in predicting. However, this is at the cost of much more expensive training. Overall, QBCM strikes a good balance in training and predicting costs, compared to NPAE and FITC. It has slightly higher cost than GRBCM, however, it provides more accurate prediction than GRBCM, as shown in the above results.

Finally, we vary the number of clusters and compare the time cost for predicting between aggregation models in Fig. 6. We do not compare the training time cost as all aggregation models use factorized training, leading to almost the same time costs. The results show that the predicting times of all aggregation models increase when the number of clusters increases. RBCM is most efficient in predicting, at the cost of poor

prediction accuracy. GRBCM has slightly higher time cost. NPAE is most expensive in predicting, since it has to do a nested GP for each testing point. QBCM has slightly higher cost than GRBCM but is much more efficient than NPAE. In the meantime, QBCM is able to provide more accurate prediction than both NPAE and GRBCM.

5 Conclusion

In this paper, we propose a query-aware Bayesian committee machine for scalable Gaussian process regression. The proposed model introduces a query-oriented submodel, which improves the aggregation process of BCM models by (i) alleviating over-confident prediction when testing point is tightly enclosed by training data; and (ii) improving the accuracy of submodels when testing point leaves the training data area. We perform empirical experiments on both synthetic and real datasets. The results confirm that the proposed model consistently outperforms the state-of-the-art approximation models for GP.

Acknowledgment

This work is partially supported by Australian Research Council Discovery Project DP180103332.

References

[1] Christopher K Williams and Carl Edward Rasmussen. Gaussian processes for machine learning. *MIT Press*, 2(3):4, 2006.

[2] Ralf Zimmermann. On the maximum likelihood training of gradient-enhanced spatial gaussian processes. *SIAM Journal on Scientific Computing*, 35(6):A2554–A2574, 2013.

[3] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Numerical gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 40(1):A172–A198, 2018.

[4] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008.

[5] Warren Scott, Peter Frazier, and Warren Powell. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization*, 21(3):996–1026, 2011.

[6] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *NeurIPS*, pages 2951–2959, 2012.

[7] Zitao Liu, Lei Wu, and Milos Hauskrecht. Modeling clinical time series using gaussian process sequences. In *SDM*, pages 623–631, 2013.

[8] Varun Chandola and Ranga Raju Vatsavai. A gaussian process based online change detection algorithm for monitoring periodic time series. In *SDM*, pages 95–106, 2011.

[9] Kai-Wen Cheng, Yie-Tarng Chen, and Wen-Hsien Fang. Video anomaly detection and localization using hierarchical feature representation and gaussian process regression. In *CVPR*, pages 2909–2917, 2015.

[10] Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *NeurIPS*, pages 329–336, 2004.

[11] Arindam Choudhury, Prasanth B Nair, and Andy J Keane. A data parallel approach for large-scale gaussian process modeling. In *SDM*, pages 95–111. SIAM, 2002.

[12] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

[13] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *NeurIPS*, pages 1257–1264, 2006.

[14] Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse gaussian process approximations. In *NeurIPS*, pages 1533–1541, 2016.

[15] Andrew G Wilson, Elad Gilboa, Arye Nehorai, and John P Cunningham. Fast kernel learning for multidimensional pattern extrapolation. In *NeurIPS*, pages 3626–3634, 2014.

[16] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[17] Volker Tresp. A bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

[18] Trung Nguyen and Edwin Bonilla. Fast allocation of gaussian process experts. In *ICML*, pages 145–153, 2014.

[19] Marc Peter Deisenroth and Jun Wei Ng. Distributed gaussian processes. In *ICML*, pages 1481–1490, 2015.

[20] Haitao Liu, Jianfei Cai, Yi Wang, and Yew-Soon Ong. Generalized robust bayesian committee machine for large-scale gaussian process regression. In *ICML*, 2018.

[21] Yanshuai Cao and David J Fleet. Generalized product of experts for automatic and principled fusion of gaussian process predictions. *arXiv preprint arXiv:1410.7827*, 2014.

[22] Didier Rullière, Nicolas Durrande, François Bachoc, and Clément Chevalier. Nested kriging predictions for datasets with a large number of observations. *Statistics and Computing*, 28(4):849–867, 2018.

[23] Carl Edward Rasmussen. *Evaluation of Gaussian processes and other methods for non-linear regression.* University of Toronto, 1999.

[24] Anton Schwaighofer and Volker Tresp. Transductive and inductive methods for approximate gaussian process regression. In *NeurIPS*, pages 977–984, 2003.