Generalized Robust Bayesian Committee Machine for Large-scale Gaussian Process Regression

Haitao Liu [1]   Jianfei Cai [2]   Yi Wang [3]   Yew-Soon Ong [2 4]

Abstract

In order to scale standard Gaussian process (GP) regression to large-scale datasets, aggregation models employ factorized training process and then combine predictions from distributed experts. The state-of-the-art aggregation models, however, either provide inconsistent predictions or require time-consuming aggregation process. We first prove the inconsistency of typical aggregations using disjoint or random data partition, and then present a consistent yet efficient aggregation model for large-scale GP. The proposed model inherits the advantages of aggregations, e.g., closed-form inference and aggregation, parallelization and distributed computing. Furthermore, theoretical and empirical analyses reveal that the new aggregation model performs better due to the consistent predictions that converge to the true underlying function when the training size approaches infinity.

1. Introduction

Gaussian process (GP) (Rasmussen & Williams, 2006) is a well-known statistical learning model extensively used in various scenarios, e.g., regression, classification, optimization (Shahriari et al., 2016), visualization (Lawrence, 2005), active learning (Fu et al., 2013; Liu et al., 2017) and multi-task learning (Alvarez et al., 2012; Liu et al., 2018). Given the training set $\boldsymbol{X} = \{\boldsymbol{x}_i \in R^d\}_{i=1}^n$ and the observation set $\boldsymbol{y} = \{y(\boldsymbol{x}_i) \in R\}_{i=1}^n$, as an approximation of the underlying function $\eta : R^d \to R$, GP provides informative predictive distributions at test points.

However, the most prominent weakness of the full GP is that it scales poorly with the training size. Given n data points, the time complexity of a standard GP paradigm scales as $\mathcal{O}(n^3)$ in the training process due to the inversion of an $n \times n$ covariance matrix; it scales as $\mathcal{O}(n^2)$ in the prediction process due to the matrix-vector operation. This weakness confines the full GP to training data of size $\mathcal{O}(10^4)$.

To cope with large-scale regression, various computationally efficient approximations have been presented. The sparse approximations reviewed in (Quiñonero-Candela & Rasmussen, 2005) employ m ($m \ll n$) inducing points to summarize the whole training data (Seeger et al., 2003; Snelson & Ghahramani, 2006; 2007; Titsias, 2009; Bauer et al., 2016), thus reducing the training complexity of full GP to $\mathcal{O}(nm^2)$ and the predicting complexity to $\mathcal{O}(nm)$. The complexity can be further reduced through distributed inference, stochastic variational inference or Kronecker structure (Hensman et al., 2013; Gal et al., 2014; Wilson & Nickisch, 2015; Hoang et al., 2016; Peng et al., 2017). A main drawback of sparse approximations, however, is that the representational capability is limited by the number of inducing points (Moore & Russell, 2015). For example, for a quick-varying function, the sparse approximations need many inducing points to capture the local structures. That is, this kind of scheme has not reduced the scaling of the complexity (Bui & Turner, 2014).

The method exploited in this article belongs to the aggregation models (Hinton, 2002; Tresp, 2000; Cao & Fleet, 2014; Deisenroth & Ng, 2015; Rullière et al., 2017), also known as consensus statistical methods (Genest & Zidek, 1986; Ranjan & Gneiting, 2010). This kind of scheme produces the final predictions by the aggregation of M submodels (GP experts) respectively trained on the subsets $\{\mathcal{D}_i = \{\boldsymbol{X}_i, \boldsymbol{y}_i\}\}_{i=1}^M$ of $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$, thus distributing the computations to "local" experts. Particularly, due to the product of experts, the aggregation scheme derives a factorized marginal likelihood for efficient training; and then it combines the experts' posterior distributions according to a certain aggregation criterion. In comparison to sparse approximations, the aggregation models (i) operate directly on the full training data, (ii) require no additional

inducing or variational parameters and (iii) distribute the computations on individual experts for straightforward parallelization (Tavassolipour et al., 2017), thus scaling them to arbitrarily large training data. In comparison to typical local GPs (Snelson & Ghahramani, 2007; Park et al., 2011), the aggregations smooth out the ugly discontinuity by the product of posterior distributions from GP experts. Note that the aggregation methods are different from the mixture-of-experts (Rasmussen & Ghahramani, 2002; Yuan & Neubauer, 2009), which suffers from intractable inference and is mainly developed for non-stationary regression.

However, it has been pointed out (Rullière et al., 2017) that there exists a particular type of training data such that typical aggregations, e.g., product-of-experts (PoE) (Hinton, 2002; Cao & Fleet, 2014) and Bayesian committee machine (BCM) (Tresp, 2000; Deisenroth & Ng, 2015), cannot offer *consistent* predictions, where "*consistent*" means the aggregated predictive distribution can converge to the true underlying predictive distribution when the training size n approaches infinity.

The major contributions of this paper are three-fold. We first prove the inconsistency of typical aggregation models, e.g., the overconfident or conservative prediction variances illustrated in Fig. 3, using conventional disjoint or random data partition. Thereafter, we present a consistent yet efficient aggregation model for large-scale GP regression. Particularly, the proposed generalized robust Bayesian committee machine (GRBCM) selects a global subset to communicate with the remaining subsets, leading to the consistent aggregated predictive distribution derived under the Bayes rule. Finally, theoretical and empirical analyses reveal that GRBCM outperforms existing aggregations due to the consistent yet efficient predictions. We release the demo codes in https://github.com/LiuHaiTao01/GRBCM.

2. Aggregation models revisited

2.1. Factorized training

A GP usually places a probability distribution over the latent function space as $f(\boldsymbol{x}) \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}'))$, which is defined by the zero mean and the covariance $k(\boldsymbol{x}, \boldsymbol{x}')$. The well-known squared exponential (SE) covariance function is

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{d}\frac{(x_i - x_i')^2}{l_i^2}\right), \quad (1)$$

where σ_f^2 is an output scale amplitude, and l_i is an input length-scale along the ith dimension. Given the noisy observation $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \epsilon$ where the *i.i.d.* noise follows $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ and the training data $\mathcal{D}$, we have the marginal likelihood $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{0}, k(\boldsymbol{X}, \boldsymbol{X}) + \sigma_\epsilon^2\boldsymbol{I})$ where $\boldsymbol{\theta}$ represents the hyperparameters to be inferred.

In order to train the GP on large-scale datasets, the aggregation models introduce a factorized training process. It first partitions the training set $\mathcal{D}$ into M subsets $\mathcal{D}_i = \{\boldsymbol{X}_i, \boldsymbol{y}_i\}$, $1 \leq i \leq M$, and then trains GP on $\mathcal{D}_i$ as an expert $\mathcal{M}_i$. In data partition, we can assign the data points randomly to the experts (*random partition*), or assign disjoint subsets obtained by clustering techniques to the experts (*disjoint partition*). Ignoring the correlation between the experts $\{\mathcal{M}_i\}_{i=1}^{M}$ leads to the factorized approximation as

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) \approx \prod_{i=1}^{M} p_i(\boldsymbol{y}_i|\boldsymbol{X}_i, \boldsymbol{\theta}_i), \quad (2)$$

where $p_i(\boldsymbol{y}_i|\boldsymbol{X}_i, \boldsymbol{\theta}_i) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}_i + \sigma_{\epsilon,i}^2\boldsymbol{I}_i)$ with $\boldsymbol{K}_i = k(\boldsymbol{X}_i, \boldsymbol{X}_i) \in R^{n_i \times n_i}$ and n_i being the training size of $\mathcal{M}_i$. Note that for simplicity all the M GP experts in (2) share the same hyperparameters as $\boldsymbol{\theta}_i = \boldsymbol{\theta}$ (Deisenroth & Ng, 2015). The factorization (2) degenerates the full covariance matrix $\boldsymbol{K} = k(\boldsymbol{X}, \boldsymbol{X})$ into a diagonal block matrix $\mathrm{diag}[\boldsymbol{K}_1, \cdots, \boldsymbol{K}_M]$, leading to $\boldsymbol{K}^{-1} \approx \mathrm{diag}[\boldsymbol{K}_1^{-1}, \cdots, \boldsymbol{K}_M^{-1}]$. Hence, compared to the full GP, the complexity of the factorized training process is reduced to $\mathcal{O}(nm_0^2)$ given $n_i = m_0 = n/M$, $1 \leq i \leq M$.

Conditioned on the related subset $\mathcal{D}_i$, the predictive distribution $p_i(y_*|\mathcal{D}_i, \boldsymbol{x}_*) \sim \mathcal{N}(\mu_i(\boldsymbol{x}_*), \sigma_i^2(\boldsymbol{x}_*))$ of $\mathcal{M}_i$ has[1]

$$\mu_i(\boldsymbol{x}_*) = \boldsymbol{k}_{i*}^{\mathsf{T}}[\boldsymbol{K}_i + \sigma_\epsilon^2\boldsymbol{I}]^{-1}\boldsymbol{y}_i, \quad (3a)$$

$$\sigma_i^2(\boldsymbol{x}_*) = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_{i*}^{\mathsf{T}}[\boldsymbol{K}_i + \sigma_\epsilon^2\boldsymbol{I}]^{-1}\boldsymbol{k}_{i*} + \sigma_\epsilon^2, \quad (3b)$$

where $\boldsymbol{k}_{i*} = k(\boldsymbol{X}_i, \boldsymbol{x}_*)$. Thereafter, the experts' predictions $\{\mu_i, \sigma_i^2\}_{i=1}^{M}$ are combined by the following aggregation methods to perform the final predicting.

2.2. Prediction aggregation

The state-of-the-art aggregation methods include PoE (Hinton, 2002; Cao & Fleet, 2014), BCM (Tresp, 2000; Deisenroth & Ng, 2015), and nested pointwise aggregation of experts (NPAE) (Rullière et al., 2017).

For the PoE and BCM family, the aggregated prediction mean and precision are generally formulated as

$$\mu_{\mathcal{A}}(\boldsymbol{x}_*) = \sigma_{\mathcal{A}}^2(\boldsymbol{x}_*)\sum_{i=1}^{M}\beta_i\sigma_i^{-2}(\boldsymbol{x}_*)\mu_i(\boldsymbol{x}_*), \quad (4a)$$

$$\sigma_{\mathcal{A}}^{-2}(\boldsymbol{x}_*) = \sum_{i=1}^{M}\beta_i\sigma_i^{-2}(\boldsymbol{x}_*) + (1 - \sum_{i=1}^{M}\beta_i)\sigma_{**}^{-2}, \quad (4b)$$

where the prior variance $\sigma_{**}^2 = k(\boldsymbol{x}_*, \boldsymbol{x}_*) + \sigma_\epsilon^2$, which is a correction term to $\sigma_{\mathcal{A}}^{-2}$, is only available for the BCM family; and β_i is the weight of the expert $\mathcal{M}_i$ at $\boldsymbol{x}_*$.

---

[1]Instead of using $p_i(f_*|\mathcal{D}_i, \boldsymbol{x}_*)$ in (Deisenroth & Ng, 2015), we here consider the aggregations in a general scenario where each expert has all its belongings at hand.

The predictions of the PoE family, which omit the prior precision σ_{**}^{-2} in (4b), are derived from the product of M experts as

$$p_{\mathcal{A}}(y_*|\mathcal{D}, \boldsymbol{x}_*) = \prod_{i=1}^{M} p_i^{\beta_i}(y_*|\mathcal{D}_i, \boldsymbol{x}_*). \qquad (5)$$

The original PoE (Hinton, 2002) employs the constant weight $\beta_i = 1$, resulting in the aggregated prediction variances that vanish with increasing M. On the contrary, the generalized PoE (GPoE) (Cao & Fleet, 2014) considers a varying $\beta_i = 0.5(\log \sigma_{**}^2 - \log \sigma_i^2(\boldsymbol{x}_*))$, which represents the difference in the differential entropy between the prior $p(y_*|\boldsymbol{x}_*)$ and the posterior $p(y_*|\mathcal{D}_i, \boldsymbol{x}_*)$, to weigh the contribution of $\mathcal{M}_i$ at $\boldsymbol{x}_*$. This varying β_i brings the flexibility of increasing or reducing the importance of experts based on the predictive uncertainty. However, the varying β_i may produce undesirable errors for GPoE. For instance, when $\boldsymbol{x}_*$ is far away from the training data such that $\sigma_i^2(\boldsymbol{x}_*) \rightarrow \sigma_{**}^2$, we have $\beta_i \rightarrow 0$ and $\sigma_{\mathrm{GPoE}}^2 \rightarrow \infty$.

The BCM family, which is opposite to the PoE family, explicitly incorporates the GP prior $p(y_*|\boldsymbol{x}_*)$ when combining predictions. For two experts $\mathcal{M}_i$ and $\mathcal{M}_j$, BCM introduces a conditional independence assumption $\mathcal{D}_i \perp \mathcal{D}_j|y_*$, leading to the aggregated predictive distribution as

$$p_{\mathcal{A}}(y_*|\mathcal{D}, \boldsymbol{x}_*) = \frac{\prod_{i=1}^{M} p_i^{\beta_i}(y_*|\mathcal{D}_i, \boldsymbol{x}_*)}{p^{\sum_i \beta_i - 1}(y_*|\boldsymbol{x}_*)}. \qquad (6)$$

The original BCM (Tresp, 2000) employs $\beta_i = 1$ but its predictions suffer from weak experts when leaving the data. Hence, inspired by GPoE, the robust BCM (RBCM) (Deisenroth & Ng, 2015) uses a varying β_i to produce robust predictions by reducing the weights of weak experts. When $\boldsymbol{x}_*$ is far away from the training data $\boldsymbol{X}$, the correction term brought by the GP prior in (4b) helps the (R)BCM's prediction variance recover σ_{**}^2. However, given $M = 1$, the predictions of RBCM as well as GPoE cannot recover the full GP predictions because usually $\beta_1 = 0.5(\log \sigma_{**}^2 - \log \sigma_1^2(\boldsymbol{x}_*)) = 0.5(\log \sigma_{**}^2 - \log \sigma_{full}^2(\boldsymbol{x}_*)) \neq 1$.

To achieve computation gains, the above aggregations introduce additional independence assumption for the experts' predictions, which however is often violated in practice and yields poor results. Hence, in the aggregation process, NPAE (Rullière et al., 2017) regards the prediction mean $\mu_i(\boldsymbol{x}_*)$ in (3a) as a random variable by assuming that $\boldsymbol{y}_i$ has not yet been observed, thus allowing for considering the covariances between the experts' predictions. Thereafter, for the random vector $[\mu_1, \cdots, \mu_M, y_*]^\mathsf{T}$, the covariances are derived as

$$\mathrm{cov}[\mu_i, y_*] = \boldsymbol{k}_{i*}^\mathsf{T} \boldsymbol{K}_{i,\epsilon}^{-1} \boldsymbol{k}_{i*}, \qquad (7a)$$

$$\mathrm{cov}[\mu_i, \mu_j] = \begin{cases} \boldsymbol{k}_{i*}^\mathsf{T} \boldsymbol{K}_{i,\epsilon}^{-1} \boldsymbol{K}_{ij} \boldsymbol{K}_{j,\epsilon}^{-1} \boldsymbol{k}_{j*}, & i \neq j, \\ \boldsymbol{k}_{i*}^\mathsf{T} \boldsymbol{K}_{i,\epsilon}^{-1} \boldsymbol{K}_{ij,\epsilon} \boldsymbol{K}_{j,\epsilon}^{-1} \boldsymbol{k}_{j*}, & i = j, \end{cases} \qquad (7b)$$

where $\boldsymbol{K}_{ij} = k(\boldsymbol{X}_i, \boldsymbol{X}_j) \in R^{n_i \times n_j}$, $\boldsymbol{K}_{i,\epsilon} = \boldsymbol{K}_i + \sigma_\epsilon^2 \boldsymbol{I}$, $\boldsymbol{K}_{j,\epsilon} = \boldsymbol{K}_j + \sigma_\epsilon^2 \boldsymbol{I}$, and $\boldsymbol{K}_{ij,\epsilon} = \boldsymbol{K}_{ij} + \sigma_\epsilon^2 \boldsymbol{I}$. With these covariances, a nested GP training process is performed to derive the aggregated prediction mean and variance as

$$\mu_{\mathrm{NPAE}}(\boldsymbol{x}_*) = \boldsymbol{k}_{\mathcal{A}*}^\mathsf{T} \boldsymbol{K}_{\mathcal{A}}^{-1} \boldsymbol{\mu}, \qquad (8a)$$

$$\sigma_{\mathrm{NPAE}}^2(\boldsymbol{x}_*) = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_{\mathcal{A}*}^\mathsf{T} \boldsymbol{K}_{\mathcal{A}}^{-1} \boldsymbol{k}_{\mathcal{A}*} + \sigma_\epsilon^2, \qquad (8b)$$

where $\boldsymbol{k}_{\mathcal{A}*} \in R^{M \times 1}$ has the ith element as $\mathrm{cov}[\mu_i, y_*]$, $\boldsymbol{K}_{\mathcal{A}} \in R^{M \times M}$ has $\boldsymbol{K}_{\mathcal{A}}^{ij} = \mathrm{cov}[\mu_i, \mu_j]$, and $\boldsymbol{\mu} = [\mu_1(\boldsymbol{x}_*), \cdots, \mu_M(\boldsymbol{x}_*)]^\mathsf{T}$. The NPAE is capable of providing *consistent* predictions at the cost of implementing a much more time-consuming aggregation because of the inversion of $\boldsymbol{K}_{\mathcal{A}}$ at each test point.

2.3. Discussions of existing aggregations

Though showcasing promising results (Deisenroth & Ng, 2015), given that $n \rightarrow \infty$ and the experts are noise-free GPs, (G)PoE and (R)BCM have been proved to be *inconsistent*, since there exists particular triangular array of data points that are dense in the input domain Ω such that the prediction variances do not go to zero (Rullière et al., 2017).

Particularly, we further show below the inconsistency of (G)PoE and (R)BCM using two typical data partitions (random and disjoint partition) in the scenario where the observations are blurred with noise. Note that since GPoE using a varying β_i may produce undesirable errors, we adopt $\beta_i = 1/M$ as suggested in (Deisenroth & Ng, 2015). Now the GPoE's prediction mean is the same as that of PoE; but the prediction variance blows up as M times that of PoE.

Definition 1. *When $n \rightarrow \infty$, let $\boldsymbol{X} \in R^{n \times d}$ be dense in $\Omega \in [0, 1]^d$ such that for any $\boldsymbol{x} \in \Omega$ we have $\lim_{n \rightarrow \infty} \min_{1 \leq i \leq n} \|\boldsymbol{x}_i - \boldsymbol{x}\| = 0$. Besides, the underlying function to be approximated has true continuous response $\mu_\eta(\boldsymbol{x})$ and true noise variance σ_η^2.*

Firstly, for the disjoint partition that uses clustering techniques to partition the data $\mathcal{D}$ into disjoint local subsets $\{\mathcal{D}_i\}_{i=1}^{M}$, The proposition below reveals that when $n \rightarrow \infty$, PoE and (R)BCM produce overconfident prediction variance that shrinks to zero; on the contrary, GPoE provides conservative prediction variance.

Proposition 1. *Let $\{\mathcal{D}_i\}_{i=1}^{M_n}$ be a disjoint partition of the training data $\mathcal{D}$. Let the expert $\mathcal{M}_i$ trained on $\mathcal{D}_i$ be GP with zero mean and stationary covariance function $k(.) > 0$. We further assume that (i) $\lim_{n \rightarrow \infty} M_n = \infty$ and (ii) $\lim_{n \rightarrow \infty} n/M_n^2 > 0$, where the second condition implies that the subset size $m_0 = n/M_n$ and the number of experts M_n are comparable such that too weak experts are not preferred. Besides, from the second condition we have $m_0 \rightarrow_{n \rightarrow \infty} \infty$, which implies that the experts become more informative with increasing n. Then, PoE and (R)BCM*

produce overconfident prediction variance at $\boldsymbol{x}_* \in \Omega$ as

$$\lim_{n \to \infty} \sigma_{\mathcal{A},n}^2(\boldsymbol{x}_*) = 0, \tag{9}$$

whereas GPoE yields conservative prediction variance

$$\sigma_\eta^2 < \lim_{n \to \infty} \sigma_{\mathcal{A},n}^2(\boldsymbol{x}_*) < \sigma_{b_n}^2(\boldsymbol{x}_*) < \sigma_{**}^2, \tag{10}$$

where $\sigma_{b_n}^2(\boldsymbol{x}_*)$ is offered by the farthest expert $\mathcal{M}_{b_n}$ ($1 \leq b_n \leq M_n$) whose prediction variance is closet to σ_{**}^2.

The detailed proof is given in Appendix A. Moreover, we have the following findings.

Remark 1. For the averaging $\sigma_{\mathrm{GPoE}}^{-2} = \frac{1}{M} \sum_{i=1}^{M} \sigma_i^{-2}$ and $\mu_{(\mathrm{G})\mathrm{PoE}} = \sum_{i=1}^{M} \frac{\sigma_i^{-2}}{\sum \sigma_i^{-2}} \mu_i$ using disjoint partition, more and more experts become relatively far away from $\boldsymbol{x}_*$ when $n \to \infty$, i.e., the prediction variances at $\boldsymbol{x}_*$ approach σ_{**}^2 and the prediction means approach the prior mean μ_{**}. Hence, empirically, when $n \to \infty$, the conservative σ_{GPoE}^2 approaches $\sigma_{b_n}^2$, and the $\mu_{(\mathrm{G})\mathrm{PoE}}$ approaches μ_{**}.

Remark 2. The BCM's prediction variance is always larger than that of PoE since

$$a_* = \frac{\sigma_{\mathrm{PoE}}^{-2}(\boldsymbol{x}_*)}{\sigma_{\mathrm{BCM}}^{-2}(\boldsymbol{x}_*)} = \frac{\sum_{i=1}^{M} \sigma_i^{-2}(\boldsymbol{x}_*)}{\sum_{i=1}^{M} \sigma_i^{-2}(\boldsymbol{x}_*) - (M-1)\sigma_{**}^{-2}} > 1$$

for $M > 1$. This means σ_{PoE}^2 deteriorates faster to zero when $n \to \infty$. Besides, it is observed that μ_{BCM} is a_* times that of PoE, which alleviates the deterioration of prediction mean when $n \to \infty$. However, when $\boldsymbol{x}_*$ is leaving $\boldsymbol{X}$, $a_* \to M$ since $\sigma_i^{-2}(\boldsymbol{x}_*) \to \sigma_{**}^{-2}$. That is why BCM suffers from undesirable prediction mean when leaving $\boldsymbol{X}$.

Secondly, for the random partition that assigns the data points randomly to the experts without replacement, The proposition below implies that when $n \to \infty$, the prediction variances of PoE and (R)BCM will shrink to zero; the PoE's prediction mean will recover $\mu_\eta(\boldsymbol{x})$, but the (R)BCM's prediction mean cannot; interestingly, the simple GPoE can converge to the underlying true predictive distribution.

Proposition 2. *Let $\{\mathcal{D}_i\}_{i=1}^{M_n}$ be a random partition of the training data $\mathcal{D}$ with (i) $\lim_{n \to \infty} M_n = \infty$ and (ii) $\lim_{n \to \infty} n/M_n^2 > 0$. Let the experts $\{\mathcal{M}_i\}_{i=1}^{M_n}$ be GPs with zero mean and stationary covariance function $k(.) > 0$. Then, for the aggregated predictions at $\boldsymbol{x}_* \in \Omega$ we have*

$$\begin{cases} \lim_{n \to \infty} \mu_{\mathrm{PoE}}(\boldsymbol{x}_*) = \mu_\eta(\boldsymbol{x}_*), \ \lim_{n \to \infty} \sigma_{\mathrm{PoE}}^2(\boldsymbol{x}_*) = 0, \\ \lim_{n \to \infty} \mu_{\mathrm{GPoE}}(\boldsymbol{x}_*) = \mu_\eta(\boldsymbol{x}_*), \ \lim_{n \to \infty} \sigma_{\mathrm{GPoE}}^2(\boldsymbol{x}_*) = \sigma_\eta^2, \\ \lim_{n \to \infty} \mu_{(\mathrm{R})\mathrm{BCM}}(\boldsymbol{x}_*) = a\mu_\eta(\boldsymbol{x}_*), \ \lim_{n \to \infty} \sigma_{(\mathrm{R})\mathrm{BCM}}^2(\boldsymbol{x}_*) = 0, \end{cases} \tag{11}$$

where $a = \sigma_\eta^{-2}/(\sigma_\eta^{-2} - \sigma_{**}^{-2}) \geq 1$ and the equality holds when $\sigma_\eta^2 = 0$.

The detailed proof is provided in Appendix B. Propositions 1 and 2 imply that no matter what kind of data partition has been used, the prediction variances of PoE and (R)BCM will shrink to zero when $n \to \infty$, which strictly limits their usability since no benefits can be gained from such useless uncertainty information.

As for data partition, intuitively, the random partition provides overlapping and coarse global information about the target function, which limits the ability to describe quick-varying characteristics. On the contrary, the disjoint partition provides separate and refined local information, which enables the model to capture the variability of target function. The superiority of disjoint partition has been empirically confirmed in (Rullière et al., 2017). Therefore, unless otherwise indicated, we employ disjoint partition for the aggregation models throughout the article.

As for time complexity, the five aggregation models have the same training process, and they only differ in how to combine the experts' predictions. For (G)PoE and (R)BCM, their time complexity in prediction scales as $\mathcal{O}(nm_0^2) + \mathcal{O}(n'nm_0)$ where n' is the number of test points.[2] For the complicated NPAE, it however needs to invert an $M \times M$ matrix $\boldsymbol{K}_\mathcal{A}$ at each test point, leading to a greatly increased time complexity in prediction as $\mathcal{O}(n'n^2)$.[3]

The inconsistency of (G)PoE and (R)BCM and the extremely time-consuming process of NPAE impose the demand of developing a consistent yet efficient aggregation model for large-scale GP regression.

3. Generalized robust Bayesian committee machine

3.1. GRBCM

Our proposed GRBCM divides M experts into two groups. The first group has a *global communication expert* $\mathcal{M}_c$ trained on the subset $\mathcal{D}_c = \mathcal{D}_1$, and the second group contains the remaining $M-1$ global or local experts[4] $\{\mathcal{M}_i\}_{i=2}^{M}$ trained on $\{\mathcal{D}_i\}_{i=2}^{M}$, respectively. The training process of GRBCM is identical to that of typical aggregations in section 2.1. The prediction process of GRBCM, however, is different. Particularly, GRBCM assigns the global communication expert with the following properties:

- (*Random selection*) The communication subset $\mathcal{D}_c$ is a random subset wherein the points are randomly se-

---

[2] $\mathcal{O}(nm_0^2)$ is induced by the update of M GP experts after optimizing hyperparameters.

[3] The predicting complexity of NPAE can be reduced by employing various hierarchical computing structure (Rullière et al., 2017), which however cannot provide identical predictions.

[4] "Global" means the expert is trained on a random subset, whereas "local" means it is trained on a disjoint subset.

lected without replacement from $\mathcal{D}$. It indicates that the points in $\boldsymbol{X}_c$ spread over the entire domain, which enables $\mathcal{M}_c$ to capture the main features of the target function. Note that there is no limit to the partition type for the remaining $M-1$ subsets.

- (*Expert communication*) The expert $\mathcal{M}_c$ with predictive distribution $p_c(y_*|\mathcal{D}_c, \boldsymbol{x}_*) \sim \mathcal{N}(\mu_c, \sigma_c^2)$ is allowed to communicate with each of the remaining experts $\{\mathcal{M}_i\}_{i=2}^M$. It means we can utilize the augmented data $\mathcal{D}_{+i} = \{\mathcal{D}_c, \mathcal{D}_i\}$ to improve over the base expert $\mathcal{M}_c$, leading to a new expert $\mathcal{M}_{+i}$ with the improved predictive distribution as $p_{+i}(y_*|\mathcal{D}_{+i}, \boldsymbol{x}_*) \sim \mathcal{N}(\mu_{+i}, \sigma_{+i}^2)$ for $2 \le i \le M$.

- (*Conditional independence*) Given the communication subset $\mathcal{D}_c$ and y_*, the independence assumption $\mathcal{D}_i \perp \mathcal{D}_j|\mathcal{D}_c, y_*$ holds for $2 \le i \ne j \le M$.

Given the conditional independence assumption and the weights $\{\beta_i\}_{i=2}^M$, we approximate the exact predictive distribution $p(y_*|\mathcal{D}, \boldsymbol{x}_*)$ using the Bayes rule as

$$
p(y_*|\mathcal{D}, \boldsymbol{x}_*)
$$
$$
\propto p(y_*|\boldsymbol{x}_*)p(\mathcal{D}_c|y_*, \boldsymbol{x}_*)\prod_{i=2}^M p(\mathcal{D}_i|\{\mathcal{D}_j\}_{j=1}^{i-1}, y_*, \boldsymbol{x}_*)
$$
$$
\approx p(y_*|\boldsymbol{x}_*)p(\mathcal{D}_c|y_*, \boldsymbol{x}_*)\prod_{i=2}^M p^{\beta_i}(\mathcal{D}_i|\mathcal{D}_c, y_*, \boldsymbol{x}_*)
$$
$$
= \frac{p(y_*|\boldsymbol{x}_*)\prod_{i=2}^M p^{\beta_i}(\mathcal{D}_{+i}|y_*, \boldsymbol{x}_*)}{p^{\sum_{i=2}^M \beta_i - 1}(\mathcal{D}_c|y_*, \boldsymbol{x}_*)}.
$$
$$(12)$$

Note that $p(\mathcal{D}_2|\mathcal{D}_c, y_*, \boldsymbol{x}_*)$ is exact with no approximation in (12). Hence, we set $\beta_2 = 1$.

With (12), GRBCM's predictive distribution is

$$
p_{\mathcal{A}}(y_*|\mathcal{D}, \boldsymbol{x}_*) = \frac{\prod_{i=2}^M p_{+i}^{\beta_i}(y_*|\mathcal{D}_{+i}, \boldsymbol{x}_*)}{p_c^{\sum_{i=2}^M \beta_i - 1}(y_*|\mathcal{D}_c, \boldsymbol{x}_*)}. \tag{13}
$$

with

$$
\mu_{\mathcal{A}}(\boldsymbol{x}_*) = \sigma_{\mathcal{A}}^2(\boldsymbol{x}_*)\left[\sum_{i=2}^M \beta_i \sigma_{+i}^{-2}(\boldsymbol{x}_*)\mu_{+i}(\boldsymbol{x}_*)\right.
$$
$$
\left. - \left(\sum_{i=2}^M \beta_i - 1\right)\sigma_c^{-2}(\boldsymbol{x}_*)\mu_c(\boldsymbol{x}_*)\right], \tag{14a}
$$
$$
\sigma_{\mathcal{A}}^{-2}(\boldsymbol{x}_*) = \sum_{i=2}^M \beta_i \sigma_{+i}^{-2}(\boldsymbol{x}_*) - \left(\sum_{i=2}^M \beta_i - 1\right)\sigma_c^{-2}(\boldsymbol{x}_*). \tag{14b}
$$

Different from (R)BCM, GRBCM employs the informative σ_c^{-2} rather than the prior σ_{**}^{-2} to correct the prediction

precision in (14b), leading to consistent predictions when $n \to \infty$, which will be proved below. Also, the prediction mean of GRBCM in (14a) now is corrected by $\mu_c(\boldsymbol{x}_*)$. Fig. 1 depicts the structure of the GRBCM aggregation model.
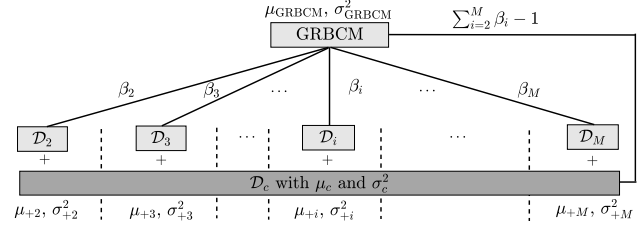

Figure 1. The GRBCM aggregation model.

In (14a) and (14b), the parameter β_i ($i > 2$) akin to that of RBCM is defined as the difference in the differential entropy between the base predictive distribution $p_c(y_*|\mathcal{D}_c, \boldsymbol{x}_*)$ and the enhanced predictive distribution $p_{+i}(y_*|\mathcal{D}_{+i}, \boldsymbol{x}_*)$ as

$$
\beta_i = \begin{cases} 1, & i = 2, \\ 0.5(\log \sigma_c^2(\boldsymbol{x}_*) - \log \sigma_{+i}^2(\boldsymbol{x}_*)), & 3 \le i \le M. \end{cases} \tag{15}
$$

It is found that after adding a subset $\mathcal{D}_i$ ($i \ge 2$) into the communication subset $\mathcal{D}_c$, if there is little improvement of $p_{+i}(y_*|\mathcal{D}_{+i}, \boldsymbol{x}_*)$ over $p_c(y_*|\mathcal{D}_c, \boldsymbol{x}_*)$, we weak the vote of $\mathcal{M}_{+i}$ by assigning a small β_i that approaches zero.

As for the size of $\boldsymbol{X}_c$, more data points bring more informative $\mathcal{M}_c$ and better GRBCM predictions at the cost of higher computing complexity. In this article, we assign all the experts with the same training size as $n_c = n_i = m_0$ and $n_{+i} = 2m_0$ for $2 \le i \le M$.

Next, we show that the GRBCM's predictive distribution will converge to the underlying true predictive distribution when $n \to \infty$.

Proposition 3. *Let $\{\mathcal{D}_i\}_{i=1}^{M_n}$ be a partition of the training data $\mathcal{D}$ with (i) $\lim_{n\to\infty} M_n = \infty$ and (ii) $\lim_{n\to\infty} n/M_n^2 > 0$. Besides, among the M subsets, there is a global communication subset $\mathcal{D}_c$, the points in which are randomly selected from $\mathcal{D}$ without replacement. Let the global expert $\mathcal{M}_c$ and the enhanced experts $\{\mathcal{M}_{+i}\}_{i=2}^{M_n}$ be GPs with zero mean and stationary covariance function $k(.) > 0$. Then, GRBCM yields consistent predictions as*

$$
\begin{cases} \lim_{n\to\infty} \mu_{\mathrm{GRBCM}}(\boldsymbol{x}_*) = \mu_\eta(\boldsymbol{x}_*), \\ \lim_{n\to\infty} \sigma_{\mathrm{GRBCM}}^2(\boldsymbol{x}_*) = \sigma_\eta^2. \end{cases} \tag{16}
$$

The detailed proof is provided in Appendix C. It is found in Proposition 3 that apart from the requirement that the communication subset $\mathcal{D}_c$ should be a random subset, the consistency of GRBCM holds for any partition of the remaining data $\mathcal{D}\backslash\mathcal{D}_c$. Besides, according to Propositions 2

and 3, both GPoE and GRBCM produce consistent predictions using random partition. It is known that the GP model $\mathcal{M}$ provides more confident predictions, i.e., lower uncertainty $U(\mathcal{M}) = \int \sigma^2(\boldsymbol{x})d\boldsymbol{x}$, with more data points. Since GRBCM trains experts on more informative subsets $\{\mathcal{D}_{+i}\}_{i=2}^{M}$, we have the following finding.

Remark 3. When using random subsets, the GRBCM's prediction uncertainty is always lower than that of GPoE, since the discrepancy $\delta_{U^{-1}} = U_{\mathrm{GRBCM}}^{-1} - U_{\mathrm{GPoE}}^{-1}$ satisfies

$$\delta_{U^{-1}} = \left[U^{-1}(\mathcal{M}_{+2}) - \frac{1}{M_n} \sum_{i=1}^{M_n} U^{-1}(\mathcal{M}_i) \right]$$
$$+ \int \sum_{i=3}^{M_n} \beta_i \left(\sigma_{+i}^{-2}(\boldsymbol{x}_*) - \sigma_c^{-2}(\boldsymbol{x}_*) \right) d\boldsymbol{x}_* > 0$$

for a large enough n. It means compared to GPoE, GRBCM converges faster to the underlying function when $n \to \infty$.

Finally, similar to RBCM, GRBCM can be executed in multi-layer computing architectures with identical predictions (Deisenroth & Ng, 2015; Ionescu, 2015), which allow to run optimally and efficiently with the available computing infrastructure for distributed computing.

3.2. Complexity

Assuming that the experts $\{\mathcal{M}_i\}_{i=1}^{M}$ have the same training size $n_i = m_0 = n/M$ for $1 \le i \le M$. Compared to (G)PoE and (R)BCM, the proposed GRBCM has a higher time complexity in prediction due to the construction of new experts $\{\mathcal{M}_{+i}\}_{i=2}^{M}$. In prediction, it first needs to calculate the inverse of $k(\boldsymbol{X}_c, \boldsymbol{X}_c)$ and $M - 1$ augmented covariance matrices $\{k(\{\boldsymbol{X}_i, \boldsymbol{X}_c\}, \{\boldsymbol{X}_i, \boldsymbol{X}_c\})\}_{i=2}^{M}$, which scales as $\mathcal{O}(8nm_0^2 - 7m_0^3)$, in order to obtain the predictions μ_c, $\{\mu_{+i}\}_{i=2}^{M}$ and σ_c^2, $\{\sigma_{+i}^2\}_{i=2}^{M}$. Then, it combines the predictions of $\mathcal{M}_c$ and $\{\mathcal{M}_{+i}\}_{i=2}^{M}$ at n' test points. Therefore, the time complexity of the GRBCM prediction process is $\mathcal{O}(\alpha n m_0^2) + \mathcal{O}(\beta n' n m_0)$, where $\alpha = (8M - 7)/M$ and $\beta = (4M - 3)/M$.

4. Numerical experiments

4.1. Toy example

We employ a 1D toy example

$$f(x) = 5x^2 \sin(12x) + (x^3 - 0.5)\sin(3x - 0.5)$$
$$+ 4\cos(2x) + \epsilon, \qquad (17)$$

where $\epsilon \sim \mathcal{N}(0, 0.25)$, to illustrate the characteristics of existing aggregation models.

We generate $n = 10^4, 5 \times 10^4, 10^5, 5 \times 10^5$ and 10^6 training points, respectively, in $[0, 1]$, and select $n' = 0.1n$ test points randomly in $[-0.2, 1.2]$. We pre-normalize each column of $\boldsymbol{X}$ and $\boldsymbol{y}$ to zero mean and unit variance. Due to

the global expert $\mathcal{M}_c$ in GRBCM, we slightly modify the disjoint partition: we first generate a random subset and then use the k-means technique to generate $M-1$ disjoint subsets. Each expert is assigned with $m_0 = 500$ data points. We implement the aggregations by the GPML toolbox[5] using the SE kernel in (1) and the conjugate gradients algorithm with the maximum number of evaluations as 500, and execute the code on a workstation with four 3.70 GHz cores and 16 GB RAM (multi-core computing in Matalb is employed). Finally, we use the Standardized Mean Square Error (SMSE) to evaluate the accuracy of prediction mean, and the Mean Standardized Log Loss (MSLL) to quantify the quality of predictive distribution (Rasmussen & Williams, 2006).
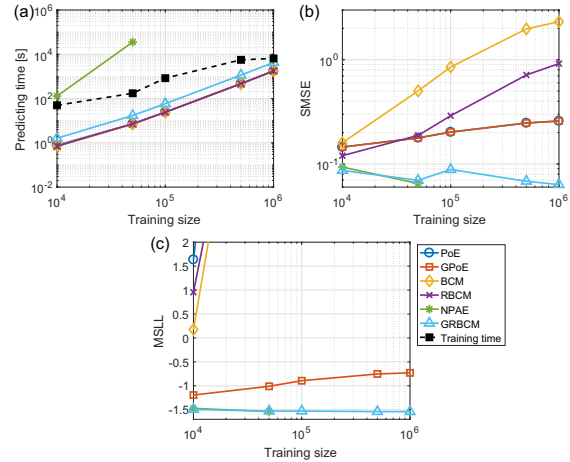

Figure 2. Comparison of different aggregation models on the toy example in terms of (a) computing time, (b) SMSE and (c) MSLL.

Fig. 2 depicts the comparative results of six aggregation models on the toy example. Note that NPAE using $n > 5 \times 10^4$ is unavailable due to the time-consuming prediction process. Fig. 2(a) shows that these models require the same training time, but they differ in the predicting time. Due to the communication expert, the GRBCM's predicting time slightly offsets the curves of (G)PoE and (R)BCM. The NPAE however exhibits significantly larger predicting time with increasing M and n'. Besides, Fig. 2(b) and (c) reveal that GRBCM and NPAE yield better predictions with increasing n, which confirm their consistency when $n \to \infty$.[6] As for NPAE, though performing slightly better than GRBCM using $n = 5 \times 10^4$, it requires several orders of magnitude larger predicting time, rendering it unsuitable for cases with many test points and subsets.

Fig. 3 illustrates the six aggregation models using $n = 10^4$ and $n = 5 \times 10^5$, respectively, in comparison to the full GP (ground truth) using $n = 10^4$.[7] It is observed that in terms

---

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

[6] Further discussions of GRBCM is shown in Appendix D.

[7] The full GP is intractable using our computer for $n = 5 \times 10^5$.
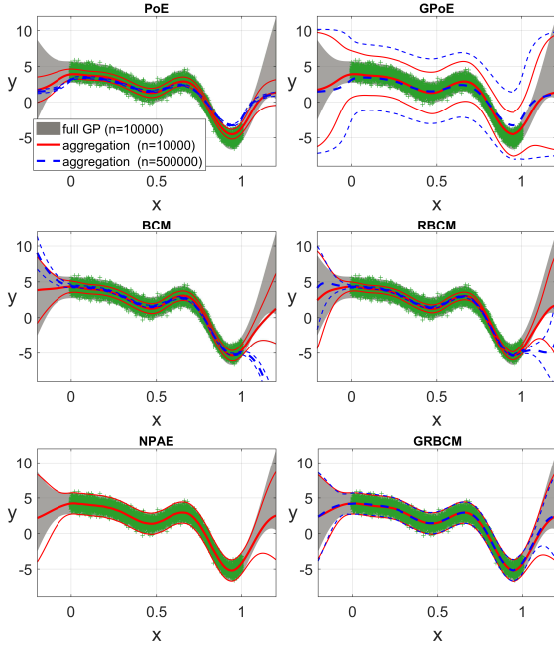
Figure 3. Illustrations of the aggregation models on the toy example. The green "+" symbols represent the 10^4 data points. The shaded area indicates 99% confidence intervals of the full GP predictions using $n = 10^4$.

of prediction mean, as discussed in remark 1, PoE and GPoE provide poorer results in the entire domain with increasing n. On the contrary, BCM and RBCM provide good predictions in the range $[0, 1]$. As discussed in remark 2, BCM however yields unreliable predictions when leaving the training data. RBCM alleviates the issue by using a varying β_i. In terms of prediction variance, with increasing n, PoE and (R)BCM tend to shrink to zero (overconfident), while GPoE tends to approach σ_{**}^2 (too conservative). Particularly, PoE always has the largest MSLL value in Fig. 2(b), since as discussed in remark 2, its prediction variance approaches zero faster.

4.2. Medium-scale datasets

We use two realistic datasets, *kin40k* (8D, 10^4 training points, 3×10^4 test points) (Seeger et al., 2003) and *sarcos* (21D, 44484 training points, 4449 test points) (Rasmussen & Williams, 2006), to assess the performance of our approach.

The comparison includes all the aggregations except the expensive NPAE.[8] Besides, we employ the fully independent training conditional (FITC) (Snelson & Ghahramani, 2006), the GP using stochastic variational inference (SVI)[9] (Hensman et al., 2013), and the subset-of-data (SOD) (Chalupka et al., 2013) for comparison. We select the inducing size m for FITC and SVI, the batch size m_b for SVI, and the

---

[8]The comparison of NPAE and GRBCM are separately provided in Appendix E.

[9]https://github.com/SheffieldML/GPy

subset size m_{sod} for SOD, such that the computing time is similar to or a bit larger than that of GRBCM. Particularly, we choose $m = 200$, $m_b = 0.1n$ and $m_{\text{sod}} = 2500$ for *kin40k*, and $m = 300$, $m_b = 0.1n$ and $m_{\text{sod}} = 3000$ for *sarcos*. Differently, SVI employs the stochastic gradients algorithm with $t_{\text{sg}} = 1200$ iterations. Finally, we adopt the disjoint partition used before to divide the *kin40k* dataset into 16 subsets, and the *sarcos* dataset into 72 subsets for the aggregations. Each experiment is repeated ten times.
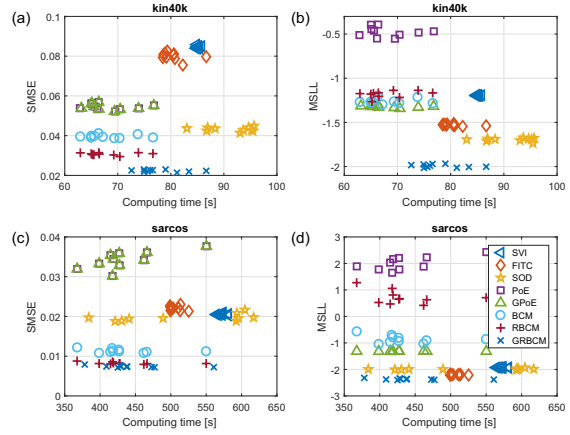

Figure 4. Comparison of the approximation models on the *kin40k* and *sarcos* datasets.

Fig. 4 depicts the comparative results of different approximation models over 10 runs on the *kin40k* and *sarcos* datasets. The horizontal axis represents the sum of training and predicting time. It is first observed that GRBCM provides the best performance on the two datasets in terms of both SMSE and MSLL at the cost of requiring a bit more computing time than (G)PoE and (R)BCM. As for (R)BCM, the small SMSE values reveal that they provide better prediction mean than FITC and SOD; but the large MSLL values again confirm that they provide overconfident prediction variance. As for (G)PoE, they suffer from poor prediction mean, as indicated by the large SMSE; but GPoE performs well in terms of MSLL. Finally, the simple SOD outperforms FITC and SVI on the *kin40k* dataset, and performs similarly on the *sarcos* dataset, which are consistent with the findings in (Chalupka et al., 2013).

Next, we explore the impact of the number M of experts on the performance of aggregations. To this end, we run them on the *kin40k* dataset with M respectively being 8, 16 and 64, and we run on the *sarcos* dataset with M respectively being 36, 72 and 288. The results in Fig. 5 turn out that all the aggregations perform worse with increasing M, since the experts become weaker; but GRBCM still yields the best performance with different M. Besides, with increasing M, the poor prediction mean and the vanishing prediction variance of PoE result in the sharp increase of MSLL values.
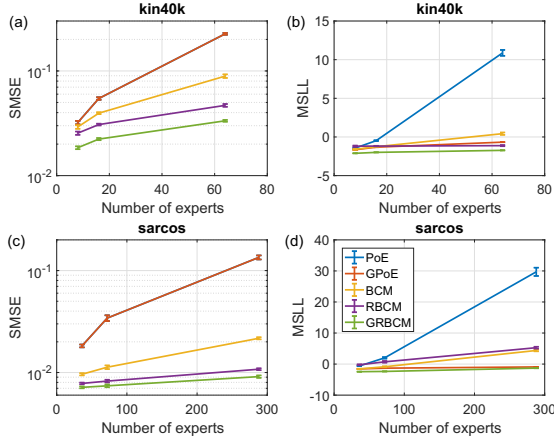
Figure 5. Comparison of the aggregation models using different numbers of experts on the *kin40k* and *sarcos* datasets.
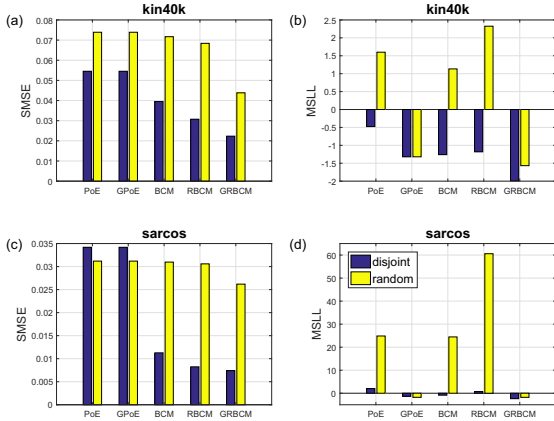

Figure 6. Comparison of the aggregation models using disjoint and random partitions on the *kin40k* dataset ($M = 16$) and the *sarcos* dataset ($M = 72$).

Finally, we investigate the impact of data partition (disjoint or random) on the performance of aggregations. The average results in Fig. 6 turn out that the disjoint partition is more beneficial for the aggregations. The results are expectable since the disjoint subsets provide separate and refined local information, whereas the random subsets provide overlapping and coarse global information. But we observe that GPoE performs well on the *sarcos* dataset using random partition, which confirms the conclusions in Proposition 2. Besides, as revealed in remark 3, even using random partition, GRBCM outperforms GPoE.

4.3. Large-scale datasets

This section explores the performance of aggregations and SVI on two large-scale datasets. We first assess them on the 90D *song* dataset, which is a subset of the million song dataset (Bertin-Mahieux et al., 2011). The *song* dataset is partitioned into 450000 training points and 65345 test

Table 1. Comparative results of the aggregation models and SVI on the *song* and *electric* datasets.

	song (450K)		*electric* (1.8M)	
	SMSE	MSLL	SMSE	MSLL
PoE	0.8527	328.82	0.1632	1040.3
GPoE	0.8527	0.1159	0.1632	24.940
BCM	2.6919	156.62	0.0073	51.081
RBCM	1.3383	24.930	0.0027	85.657
SVI	0.7909	**-0.1885**	0.0042	-1.1410
GRBCM	**0.7321**	-0.1571	**0.0024**	**-1.3161**

points. We then assess the models on the 11D *electric* dataset that is partitioned into 1.8 million training points and 249280 test points. We follow the normalization and data pre-processing in (Wilson et al., 2016) to generate the two datasets.[10] For the *song* dataset, we use the foregoing disjoint partition to divide it into $M = 720$ subsets, and use $m = 800$, $m_b = 5000$ and $t_{sg} = 1300$ for SVI; for the *electric* dataset, we divide it into $M = 2880$ subsets, and use $m = 1000$, $m_b = 5000$ and $t_{sg} = 1500$ for SVI. As a result, each expert is assigned with $m_0 = 625$ data points for the aggregations.

Table 1 reveals that the (G)PoE's SMSE value is smaller than that of (R)BCM on the *song* dataset. The poor prediction mean of BCM is caused by the fact that the *song* dataset is highly clustered such that BCM suffers from weak experts in regions with scarce points. On the contrary, due to the almost uniform distribution of the *electric* data points, the (R)BCM's SMSE is much smaller than that of (G)PoE. Besides, unlike the vanishing prediction variances of PoE and (R)BCM when $n \to \infty$, GPoE provides conservative prediction variance, resulting in small MSLL values on the two datasets. The proposed GRBCM always outperforms the other aggregations in terms of both SMSE and MSLL on the two datasets due to the consistency. Finally, GRBCM performs similarly to SVI on the *song* dataset; but GRBCM outperforms SVI on the *electric* dataset.

5. Conclusions

To scale the standard GP to large-scale regression, we present the GRBCM aggregation model, which introduces a global communication expert to yield consistent yet efficient predictions when $n \to \infty$. Through theoretical and empirical analyses, we demonstrated the superiority of GRBCM over existing aggregations on datasets with up to 1.8M training points.

The superiority of local experts is the capability of capturing local patterns. Hence, further works will consider the experts with individual hyperparameters in order to capture non-stationary and heteroscedastic features.

---

[10]The datasets and the pre-processing scripts are available in https://people.orie.cornell.edu/andrew/.

Acknowledgements

References

Alvarez, Mauricio A, Rosasco, Lorenzo, Lawrence, Neil D, et al. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3): 195–266, 2012.

Bauer, Matthias, van der Wilk, Mark, and Rasmussen, Carl Edward. Understanding probabilistic sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems*, pp. 1533–1541. Curran Associates, Inc., 2016.

Bertin-Mahieux, Thierry, Ellis, Daniel PW, Whitman, Brian, and Lamere, Paul. The million song dataset. In *ISMIR*, pp. 1–6, 2011.

Bui, Thang D and Turner, Richard E. Tree-structured Gaussian process approximations. In *Advances in Neural Information Processing Systems*, pp. 2213–2221. Curran Associates, Inc., 2014.

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

Chalupka, Krzysztof, Williams, Christopher KI, and Murray, Iain. A framework for evaluating approximation methods for Gaussian process regression. *Journal of Machine Learning Research*, 14(Feb):333–350, 2013.

Choi, Taeryon and Schervish, Mark J. Posterior consistency in nonparametric regression problems under Gaussian process priors. Technical report, Carnegie Mellon University, 2004.

Deisenroth, Marc Peter and Ng, Jun Wei. Distributed Gaussian processes. In *International Conference on Machine Learning*, pp. 1481–1490. PMLR, 2015.

Fu, Yifan, Zhu, Xingquan, and Li, Bin. A survey on instance selection for active learning. *Knowledge and Information Systems*, 35(2):249–283, 2013.

Gal, Yarin, van der Wilk, Mark, and Rasmussen, Carl Edward. Distributed variational inference in sparse Gaussian process regression and latent variable models. In *Advances in Neural Information Processing Systems*, pp. 3257–3265. Curran Associates, Inc., 2014.

Genest, Christian and Zidek, James V. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135, 1986.

Hensman, James, Fusi, Nicolò, and Lawrence, Neil D. Gaussian processes for big data. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, pp. 282–290. AUAI Press, 2013.

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

Hoang, Trong Nghia, Hoang, Quang Minh, and Low, Bryan Kian Hsiang. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *International Conference on Machine Learning*, pp. 382–391. PMLR, 2016.

Ionescu, Radu Cristian. Revisiting large scale distributed machine learning. *arXiv preprint arXiv:1507.01461*, 2015.

Lawrence, Neil. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6(Nov): 1783–1816, 2005.

Liu, Haitao, Cai, Jianfei, and Ong, Yew-Soon. An adaptive sampling approach for Kriging metamodeling by maximizing expected prediction error. *Computers & Chemical Engineering*, 106(Nov):171–182, 2017.

Liu, Haitao, Cai, Jianfei, and Ong, Yew-Soon. Remarks on multi-output Gaussian process regression. *Knowledge-Based Systems*, 144(March):102–121, 2018.

Moore, David and Russell, Stuart J. Gaussian process random fields. In *Advances in Neural Information Processing Systems*, pp. 3357–3365. Curran Associates, Inc., 2015.

Park, Chiwoo, Huang, Jianhua Z, and Ding, Yu. Domain decomposition approach for fast Gaussian process regression of large spatial data sets. *Journal of Machine Learning Research*, 12(May):1697–1728, 2011.

Peng, Hao, Zhe, Shandian, Zhang, Xiao, and Qi, Yuan. Asynchronous distributed variational Gaussian process for regression. In *International Conference on Machine Learning*, pp. 2788–2797. PMLR, 2017.

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

Ranjan, Roopesh and Gneiting, Tilmann. Combining probability forecasts. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(1):71–91, 2010.

Rasmussen, Carl E and Ghahramani, Zoubin. Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems*, pp. 881–888. Curran Associates, Inc., 2002.

Rasmussen, Carl Edward and Williams, Christopher K. I. *Gaussian processes for machine learning*. MIT Press, 2006.

Rullière, Didier, Durrande, Nicolas, Bachoc, François, and Chevalier, Clément. Nested Kriging predictions for datasets with a large number of observations. *Statistics and Computing*, pp. 1–19, 2017.

Seeger, Matthias, Williams, Christopher, and Lawrence, Neil. Fast forward selection to speed up sparse Gaussian process regression. In *Artificial Intelligence and Statistics*, pp. EPFL–CONF–161318. PMLR, 2003.

Shahriari, Bobak, Swersky, Kevin, Wang, Ziyu, Adams, Ryan P, and de Freitas, Nando. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Snelson, Edward and Ghahramani, Zoubin. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pp. 1257–1264. MIT Press, 2006.

Snelson, Edward and Ghahramani, Zoubin. Local and global sparse Gaussian process approximations. In *Artificial Intelligence and Statistics*, pp. 524–531. PMLR, 2007.

Tavassolipour, Mostafa, Motahari, Seyed Abolfazl, and Shalmani, Mohammad-Taghi Manzuri. Learning of Gaussian processes in distributed and communication limited systems. *arXiv preprint arXiv:1705.02627*, 2017.

Titsias, Michalis K. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pp. 567–574. PMLR, 2009.

Tresp, Volker. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

Vazquez, Emmanuel and Bect, Julien. Pointwise consistency of the Kriging predictor with known mean and covariance functions. In *9th International Workshop in Model-Oriented Design and Analysis*, pp. 221–228. Springer, 2010.

Wilson, Andrew and Nickisch, Hannes. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *International Conference on Machine Learning*, pp. 1775–1784. PMLR, 2015.

Wilson, Andrew Gordon, Hu, Zhiting, Salakhutdinov, Ruslan, and Xing, Eric P. Deep kernel learning. In *Artificial Intelligence and Statistics*, pp. 370–378. PMLR, 2016.

Yuan, Chao and Neubauer, Claus. Variational mixture of Gaussian process experts. In *Advances in Neural Information Processing Systems*, pp. 1897–1904. Curran Associates, Inc., 2009.