OPT2021: 13th Annual Workshop on Optimization for Machine Learning

Gaussian Graphical Models as an Ensemble Method for Distributed Gaussian Processes

Hamed Jalali HAMED.JALALI@WSII.UNI-TUEBINGEN.DE
Gjergji Kasneci GJERGJI.KASNECI@UNI-TUEBINGEN.DE
University of Tuebingen, Germany

Abstract

Distributed Gaussian process (DGP) is a popular approach to scale GP to big data which divides the training data into some subsets, performs local inference for each partition, and aggregates the results to acquire global prediction. To combine the local predictions, the *conditional independence assumption* is used which basically means there is a perfect diversity between the subsets. Although it keeps the aggregation tractable, it is often violated in practice and generally yields poor results. In this paper, we propose a novel approach for aggregating the Gaussian experts' predictions by Gaussian graphical model (GGM) where the target aggregation is defined as an unobserved latent variable and the local predictions are the observed variables. We first estimate the joint distribution of latent and observed variables using the Expectation-Maximization (EM) algorithm. The interaction between experts can be encoded by the precision matrix of the joint distribution and the aggregated predictions are obtained based on the property of conditional Gaussian distribution. Using both synthetic and real datasets, our experimental evaluations illustrate that our new method outperforms other state-of-the-art DGP approaches.

1. Introduction

Gaussian processes (GPs) are powerful non-parametric statistical methods based on Bayes' theorem. Without the need for restrictive assumptions, they are capable to estimate complex models with a low amount of uncertainty. They have been widely used in practice, e.g. optimization [24], data visualization [13], reinforcement learning [6], multitask learning [1], online streaming models [14], and time series analysis [26]. Despite many advantages, GPs suffer from their computational costs where they poorly scale with the size of the dataset. The prominent distributed Gaussian processes (also called local approximation GPs) are based on the divide-and-conquer approach. It means the training data is divided into some partitions (called experts), the local inference is done for each partition separately, and at the end, these local estimations are combined using an ensemble method. All experts share the same hyper-parameters, which leads to automatic regularisation and the model tends to prevent the overfitting of individual experts [5].

In a DGP, the *conditional independence* assumption (CI) between partitions allows factorizing the global posterior distribution as a product of local distributions. Although this assumption reduces the computational cost, it is often violated in practice. However, solutions that deal with the dependency problem (e.g. NPAE method [23]) suffer from extra computational costs and therefore, are impractical for large data sets.

The key contribution of our work lies in aggregating the local experts' predictions considering their dependencies. Unlike conventional DGPs, here the CI assumption is violated to improve the

prediction quality. The conditional dependency is inferred as the interactions between nodes in a continuous form of a Markov random field (MRF). We consider the local and latent experts as nodes of an undirected graph. Then, the Gaussian graphical model (GGM) is used to construct the undirected graph between Gaussian experts and their interactions. Since the latent expert is unobserved, we use the latent variable Gaussian graphical model (LVGGM) to estimate the joint distribution of observed and latent experts. The final predictions are the mean of the conditional distribution of the latent expert given observed experts. Relative to the available baselines, our approach substantially provides competitive prediction performance than other state-of-the-art (SOTA) approaches, which use the CI assumption. The structure of the paper is as follows. Section 2 introduces the problem formulation and related works. In Section 3 the proposed model and the inference process are presented. Section 4 shows the experimental results and we conclude in Section 5.

2. Background and Problem Set-up

2.1. Background

Let us consider the regression problem $y = f(x) + \epsilon$, where $x \in R^d$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$, and the Gaussian likelihood is $p(y|f) = \mathcal{N}(f, \sigma^2 I)$. The objective is to learn the latent function f from a training set $\mathcal{D} = \{X, y\}$ of size n. The Gaussian process regression is a collection of random variables of which any finite subset has a joint Gaussian distribution. The GP then describes a prior distribution over the latent functions as $f \sim GP(0, k(x, x'))$, where $k(x, x')$ is the covariancee function (kernel) with hyperparameters ψ. To train the GP, the hyperparameters $\theta = \{\sigma^2, \psi\}$ should be determined such that they maximise the log-marginal likelihood,

$$\log p(y|X) = -\frac{1}{2} y^T \mathcal{C}^{-1} y - \frac{1}{2} \log |\mathcal{C}| - \frac{n}{2} \log 2\pi, \tag{1}$$

where $\mathcal{C} = K + \sigma^2 I$ and $K = k(X, X)$. According to (1), the training step scales as $\mathcal{O}(n^3)$ because it is affected by the inversion and determinant of the $n \times n$ matrix $\mathcal{C}$. Therefore, for large data sets, GP training is a time-consuming task and imposes limitations on the scalability of GPs.

2.2. Distributed Gaussian Process

The term distributed Gaussian process was proposed by [5] uses the fact that the computations of the standard GP can be distributed among individual computing units. To do that, one divides the full training data set $\mathcal{D}$ into M partitions (called experts) and trains standard GPs on these partitions. Let $\mathcal{D}^{'} = \{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$ be the partitions, and X_i and y_i be the input and output of partition $\mathcal{D}_i$. All GP experts are trained jointly and share a single set of hyper-parameters $\theta = \{\sigma^2, \psi\}$. For a test set X^* of size n_t, the local prediction of the i-th GP expert $\mathcal{M}_i$ is:

$$\mu_i^* = k_{i*}^T (K_i + \sigma^2 I)^{-1} y_i, \tag{2}$$

where $K_i = k(X_i, X_i)$, and $k_{i*} = k(X_i, X^*)$.

Aggregating the experts in DGP is based on the assumption that they are conditionally independent. For a test input x^*, the posterior distribution of DGP is given as the product of multiple local densities, i.e. $p(y^*|\mathcal{D}, x^*) \propto \prod_{i=1}^{M} p_i(y^*|\mathcal{D}_i, x^*)$. The most popular aggregations are generalised product of experts (GPoE) [3], robust Bayesian committee machine (RBCM) [5] and generalized robust Bayesian committee machine (GRBCM) [17], see Appendix A.

2.3. Dependency

The CI assumption is used widely in ensemble methods for both regression and classification problems [19, 21]. The DGPs use CI to reduce the computational costs of the training and prediction processes. However, their predictions are not accurate enough and CI-based aggregation generally returns sub-optimal solution [9, 10]. In local approximation GPs, the dependency between experts has been discussed in few works. For instance, the nested pointwise aggregation of experts (NPAE) method [23] uses the internal correlation between local experts and the dependency between local experts and target variable y^*. However, this pointwise aggregation suffers from high time complexity which cubically depends on the number of experts at each test point, i.e. $\mathcal{O}(n_t M^3)$, and therefore, it is not an efficient solution for large datasets. Figure 2 in Appendix B shows the computational graphs of CI-based and dependency-based aggregation strategies.

3. Aggregating Conditionally dependent Experts with an Undirected Graph

At the heart of our work is the following ingredient. First, we assume that y_i in (2) has not yet been observed, see [23]. Then the experts' predictions μ_i^* can be considered as a *random variable*. This allows us to leverage correlations between experts. Then, we exert the Gaussian graphical model, where the nodes of the graph are the experts (local and latent) and the edges are the interactions between them.

3.1. Aggregating Dependent Experts' Predictions

Assume the Gaussian experts $\mathcal{M} = \{\mathcal{M}_1, \dots, \mathcal{M}_M\}$ have been trained on separated subsets and let $\mu^* = [\mu_1^*, \dots, \mu_M^*]^T$ be a $n_t \times M$ matrix that contains their centered predictions at n_t test points. As a consequence of the choice of the prior, the joint distribution of the local experts μ^* and target expert y^* is multivariate Gaussian distribution because any vector of linear combinations of observation is itself a Gaussian vector. Let $\Sigma_{y^*\mu^*}$ encodes the correlation between latent expert y^* and local experts μ^*, and $\Sigma_{\mu^*\mu^*}$ depicts the correlation between local experts. Employing the properties of conditional Gaussian distributions for the centered random vector allows for the following aggregation:

$$y_A^* = \Sigma_{y^*\mu^*}^T \Sigma_{\mu^*\mu^*}^{-1} \mu^*. \tag{3}$$

which is the mean of conditional distribution of y^* given μ^*, i.e. $p(y^*|\mu^*)$.

Proposition 1 (BLUP) *y_A^* is the best linear unbiased predictor of y^*, i.e. for linear estimators of the form $\beta\mu^* = \sum_{i=1}^M \beta_i \mu_i^*$, the mean square error $(y^* - \beta\mu^*)^2$ is minimized when $\beta = \Sigma_{y^*\mu^*}^T \Sigma_{\mu^*\mu^*}^{-1}$.*

The proof of Proposition 1 can be found in Appendix C. In the next subsection, we show how the GGM can be adapted to the local approximation problem with a latent target variable and suggest a new method to compute the aggregated estimator y_A^*.

3.2. Gaussian Graphical Models for Dependent Gaussian Experts

Gaussian graphical models [7, 22, 28] are continuous forms of pairwise MRFs which assume the variables in the network follow a multivariate Gaussian distribution. The distribution for a GGM is

$$p(\mu^*|\xi,\Omega) \propto \exp\left\{-\frac{1}{2}(\mu^*-\xi)^T\Omega(\mu^*-\xi)\right\}, \tag{4}$$

where $\mu^* = \{\mu_1^*,\ldots,\mu_M^*\}$ are the experts, and ξ and Ω are the mean and precision, respectively. The matrix Ω is also known as the potential or information matrix. In a GGM, if $\Omega_{ij}=0$, then μ_i^* ard μ_j^* are conditionally independent given all other variables, i.e. there is no edge between μ_i^* ard μ_j^* in the graph. GGMs use the common sparsity assumption, that is, there are only few edges in the network and thus the precision matrix is sparse. To this end, the graphical Lasso (GLasso) regression [8] is used to perform neighborhood selection for the network. It maximizes the log-likelihood subject to an element-wise $\mathcal{L}_1$ norm penalty on Ω. Precisely, for sample covariance S and Gaussian log-likelihood $\mathcal{L}(\Omega;S) = \log|\Omega| - trace(S\,\Omega)$, the objective function is

$$\widehat{\Omega}_\lambda = \arg\min_{\Omega}\ \left(-\mathcal{L}(\Omega;S) + \lambda\left\|\Omega\right\|_1\right). \tag{5}$$

The precision matrix Ω has been used before to find clusters of strongly dependent experts [10] and selecting most important experts in local approximation [11].

3.3. GGM-Based Aggregation using EM Algorithm

The main input of the GLasso method is the sample covariance of our observations. Since the targeted expert y^* is unobserved, one row (column) in S, related to y^* is unknown. Let $S_{\mu^*\mu^*}$ is a known $M\times M$ matrix of the sample covariance of the observed variables μ^*, $S_{y^*\mu^*}$ is an unknown $1\times M$ vector that shows the sample covariance between latent and observed expert, and $S_{y^*y^*}$ is the internal potential of a latent expert. To use the GLasso, it is needed to estimate unknown partitions of S, i.e. $S_{y^*\mu^*}$ and $S_{y^*y^*}$. Here, we explain how the expected-maximization algorithm can help us.

E-Step: The E-step Calculates $Q(\Omega \mid \Omega^{(t)})$, the expected value of the penalized negative log-likelihood function with respect to the conditional distribution of y^* given μ^* under the current estimate $\Omega^{(t)}$ of Ω:

$$Q(\Omega\mid\Omega^{(t)}) = \mathrm{E}_{y^*|\mu^*,\Omega^{(t)}}\left[-\mathcal{L}(\Omega;S)+\lambda\left\|\Omega\right\|_1\right] = -\log|\Omega| + trace\{E_{y^*|\mu^*,\Omega^{(t)}}(S)\Omega\} + \lambda\left\|\Omega\right\|_1.$$

Let $\Sigma^{(t)} = (\Omega^{(t)})^{-1}$, the conditional distribution of y^* given μ^* under the current estimate $\Omega^{(t)}$ follows

$$N\left(\Sigma^{(t)}_{y^*\mu^*}(\Sigma^{(t)}_{\mu^*\mu^*})^{-1}\mu^*, \Sigma^{(t)}_{y^*} - \Sigma^{(t)}_{y^*\mu^*}(\Sigma^{(t)}_{\mu^*\mu^*})^{-1}\Sigma^{(t)}_{\mu^*y^*}\right).$$

Therefore, unknown partitions of $\widehat{S} = E_{y^*|\mu^*,\Omega^{(t)}}(S)$ can be estimated as below:

$$\widehat{S}_{\mu^*y^*} = S_{\mu^*\mu^*}(\Sigma^{(t)}_{\mu^*\mu^*})^{-1}\Sigma^{(t)}_{\mu^*y^*}, \tag{6}$$

$$\widehat{S}_{y^*y^*} = \Sigma^{(t)}_{y^*y^*} - \Sigma^{(t)}_{y^*\mu^*}(\Sigma^{(t)}_{\mu^*\mu^*})^{-1}\Sigma^{(t)}_{\mu^*y^*} + \Sigma^{(t)}_{y^*\mu^*}(\Sigma^{(t)}_{\mu^*\mu^*})^{-1}S_{\mu^*\mu^*}(\Sigma^{(t)}_{\mu^*\mu^*})^{-1}\Sigma^{(t)}_{\mu^*y^*}. \tag{7}$$

M-Step : This step returns the updated precision matrix $\mathbf{\Omega}^{(t+1)}$ that maximize $Q(\Omega\mid\Omega^{(t)})$ over all $(M+1)\times(M+1)$ positive-definite matrices Ω. It is a *GLasso* problem and is equivalent to this minimization problem:

$$\Omega^{(t)} = \arg\min_{\Omega}\left(-\log|\Omega| + trace\{\widehat{S}\Omega\} + \lambda\left\|\Omega\right\|_1\right). \tag{8}$$

Algorithm 1 summarizes the whole procedure of the proposed ensemble, EMGGM.

Algorithm 1: GGM-Based Experts Aggregation (EMGGM)

Data: μ^*, λ, R (number of iterations)
Result: Aggregated predictions y_A^*
Initialize y^*;
Calculate sample covariance $S^{(0)}$ of (y^*, μ^*);
Estimate the initial parameter $\Omega^{(0)}$ using Equation (5);
$t \leftarrow 1$;
while $t \leq R$ **do**
  Estimate $E_{y^*|\mu^*,\Omega^{(t)}}(S_{\mu^* y^*})$ using Equation (6) ;
  Estimate $E_{y^*|\mu^*,\Omega^{(t)}}(S_{y^* y^*})$ using Equation (7);
  Update the sample covariance as $S^{(t)} = E_{y^*|\mu^*,\Omega^{(t)}}(S)$;
  Update the precision matrix $\Omega^{(t)}$ using Equation (8) ;
  $\Sigma^{(t)} \leftarrow (\Omega^{(t)})^{-1}$ and $t \leftarrow t + 1$;
end
Estimate the aggregated prediction y_A^* using Equation (3) ;

3.4. Discussion and Challenges

The proposed ensemble is capable to aggregate local experts considering their dependencies and its computational and storage costs are much smaller than NPAE which uses dependent experts, see Appendix D. Besides, the normality assumption for joint distribution is not a restrictive assumption and can be relaxed, see Appendix E. This gives the result that the proposed strategy can aggregate non-Gaussian experts.

An EM iteration does increase the likelihood function $\mathcal{L}(\Omega; S)$. However, no guarantee exists that the sequence converges to a maximum likelihood estimator. It is only guaranteed to converge to a point with zero gradient with respect to the parameters. So it can indeed get stuck at saddle points, see [18]. The converges property of the EM algorithm can be improved using a variety of heuristic or meta-heuristic approaches that enable EM to escape a local maximum, e.g. hill climbing and simulated annealing.

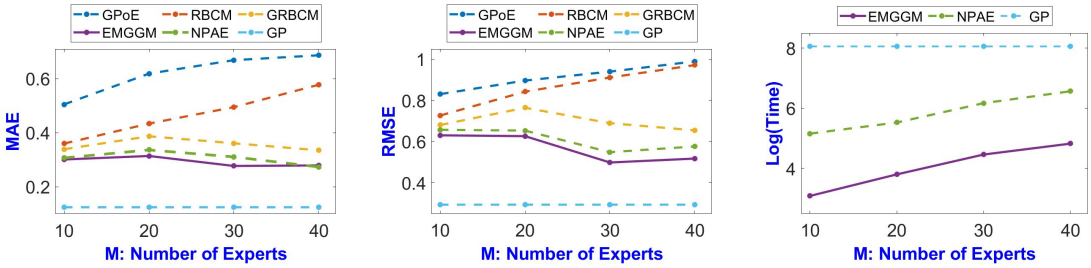
To avoid challenges in the convergence of EM, latent variable GGM (LVGGM) can be used. Maximizing the likelihood function of an LVGGM leads to a nonlinear optimization problem which is solved by convex or non-convex optimization methods. This strategy can be studied in future works to estimate the aggregated estimator y_A^* in (3), see Appendix F.

4. Experiments

In this section, we evaluate the prediction quality of the aggregated estimator using the conventional mean absolute error (MAE) and the root mean squared error (RMSE). We use the simulated data of a one-dimensional analytical function [10, 17],

$$f(x) = 5x^2 sin(12x) + (x^3 - 0.5)sin(3x - 0.5) + 4cos(2x) + \epsilon, \tag{9}$$

where $\epsilon \sim \mathcal{N}\left(0, (0.2)^2\right)$. We generate $n = 10^4$ training points in $[0, 1]$, and $n_t = 10^3$ test points in $[-0.2, 1.2]$. The data is normalized to zero mean and unit variance. We vary the number



(*a*) MAE (*b*) RMSE (*c*) Log(Time)

Figure 1: **Prediction quality** of DGP methods with respect for different number of experts M.

of experts, $M = \{10, 20, 30, 40\}$, to evaluate different partition sizes. The prediction quality of the proposed ensemble is compared with the other baselines: GPoE [3], RBCM [5], GRBCM [17], NPAE [23], and the full GP. We use the standard squared exponential kernel, a Gaussian likelihood and the K-means partitioning method.

Figure 1 (a) and Figure 1 (b) depict the prediction quality of different baselines. The ensemble methods that use dependency between experts, i.e. EMGGM and NPAE, outperform the CI-based baselines. However, the proposed method has slightly better predictions than NPAE. Figure 1 (c) presents the computation time of the ensembles that use dependency between experts. Remarkably, EMGGM provides predictions in just a fraction of NPAE's running time.

5. Conclusion

In this work, we have proposed a novel ensemble method, EMGGM, for distributed GPs which aggregate dependent local experts' predictions using GGMs. Our proposed approach uses undirected graphical models and EM algorithm to estimate the final predictions. Through empirical analyses, we illustrated the superiority of EMGGM over existing SOTA aggregation methods. Finally, we hope to use our insights to develop aggregations that provide the full predictive distribution.

References

[1] M. A. Alvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

[2] E.J. Candès, X. Li, y. MA, and j. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

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

[4] V. Chandrasekaran, P.A. Parrilo, and A. S. Willsky. Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40:1935–1967, 2012.

[5] M. P. Deisenroth and J. W. Ng. Distributed gaussian processes. *International Conference on Machine Learning*, pages 1481–1490, 2015.

[6] M. P. Deisenroth, D. Fox, and C. E. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37 (2):408–423, 2013.

[7] M. Drton and M.H. Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4:365–393, 2017.

[8] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[9] A. Jaffe, E. Fetaya, B. Nadler, T. Jiang, and Y. Kluger. Unsupervised ensemble learning with dependent classifiers. *In Artificial Intelligence and Statistics*, pages 351–360, 2016.

[10] H. Jalali and G. Kasneci. Aggregating dependent gaussian experts in local approximation. *arXiv preprint arXiv:2010.08873*, 2020.

[11] H. Jalali, M. Pawelczyk, and G. Kasneci. Gaussian experts selection using graphical models. *arXiv preprint arXiv:2102.01496*, 2021.

[12] J. Lafferty, H. Liu, and L. Wasserman. Sparse nonparametric graphical models. *Statistical Science*, 27(4):519–537, 2012.

[13] N. Lawrence. Taking the human out of the loop: A review of bayesian optimization. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

[14] T. Le, K. Nguyen, V. Nguyen, T. D. Nguyen, and D. Phung. Gogp: Fast online regression with gaussian processes. *IEEE International Conference on Data Mining*, pages 257–266, 2017.

[15] B. Li and E. Solea. A nonparametric graphical model for functional data with application to brain networks based on fmri. *Journal of the American Statistical Association*, 113(524): 1637–1655, 2018.

[16] H. Liu, J. Lafferty, and L. Wasserman. nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *The Journal of Machine Learning Research (JMLR)*, 10: 2295–2328, 2009.

[17] H. Liu, J. Cai, Y. Ong, and Y. Wang. Generalized robust bayesian committee machine for large-scale gaussian process regression. *International Conference on Machine Learning*, pages 1–10, 2018.

[18] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley-Interscience, 2 edition, 2008.

[19] J. Mendes-Moreira, C. Soares, A.M Jorge, and J.F.D Sousa. Ensemble approaches for regression: A survey. *Acm computing surveys (csur)*, 45(4):1–40, 2012.

[20] J. J. Mulgrave and S. Ghosal. Bayesian inference in nonparanormal graphical models. *Bayesian Analysis*, 15(2):449–475, 2020.

[21] F. Parisi, F. Strino, B. Nadler, and Y. Kluger. Ranking and combining multiple predictors without labeled data. *Proceedings of the National Academy of Sciences*, 111(4):1253–1258, 2014.

[22] H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005.

[23] D. Rullière, N. Durrande, F. Bachoc, and C. Chevalier. Nested kriging predictions for datasets with a large number of observations. *Statistics and Computing*, 28(4):849–867, 2018.

[24] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[25] E. Solea and H. Dette. Nonparametric and high-dimensional functional graphical models. *arXiv preprint arXiv:2103.10568s*, 2021.

[26] F. Tobar, T. D. Bui, and R. E. Turner. Learning stationary time series using gaussian processes with nonparametric kernels. *In Advances in Neural Information Processing Systems*, pages 3501–3509, 2015.

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

[28] C. Uhler. Gaussian graphical models: an algebraic and geometric perspective. *arXiv preprint arXiv:1707.04345*, 2017.

[29] T. Wang, Z. Renand Y. Ding, Z. Fang, Z. Sun M. L. MacDonald, R. A. Sweet, J. Wang, and W. Chen. Fastggm: An efficient algorithm for the inference of gaussian graphical model in biological networks. *PLOS Computational Biology*, 12(2):1–16, 2016.

[30] P. Xu, J. Ma, and Q. Gu. Speeding up latent variable gaussian graphical model estimation via nonconvex optimizations. *Advances in Neural Information Processing Systems*, 2017.

[31] Ming Yuan. Discussion: Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4):1968–1972, 2012.

[32] J. Zhang, M. Wang, Q. Li, S. Wang, X. Chang, and B. Wang. Quadratic sparse gaussian graphical model estimation method for massive variables. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 2964–2972, 2020.

Appendices

A. Distributed GP Models

In this section we review other divide-and-conquer distributed GP approaches in more detail, focusing on how other methods perform expert weighting. There are two main families of distributed GPs: product of experts and Bayesian committee machine.

Product of Experts. The posterior distribution of the PoE model is given by the product of multiple densities (i.e., the experts). Because of the product operation, the prediction quality of PoE suffers considerably from weak experts. To improve on this aspect, [3] proposed the GPoE model, which assigns importance weight to the experts.

For independent experts $\{\mathcal{M}\}_{i=1}^{M}$ trained on different partitions $\mathcal{D}_i$ the predictive distribution for a test input X^* is given by:

$$p(y^*|\mathcal{D}, X^*) = \prod_{i=1}^{M} p_i^{\beta_i}(y^*|\mathcal{D}_i, X^*), \tag{10}$$

where $\beta = \{\beta_1, \ldots, \beta_M\}$ controls the expert importance. The product distribution in (10) is proportional to a Gaussian distribution with mean and precision, respectively:

$$\mu_D^* = \Sigma_D^* \sum_{i=1}^{M} \beta_i (\Sigma_i^*)^{-1} \mu_i^*, \quad (\Sigma_D^*)^{-1} = \sum_{i=1}^{M} \beta_i (\Sigma_i^*)^{-1}.$$

The standard PoE can be recovered by setting $\beta_i = 1 \ \forall i$. The precision corresponding to the PoE prediction, i.e. $(\Sigma_D^*)^{-1}$, is a linear sum of individual precision values: hence, an increasing number of local GPs increases the precision and therefore it leads to a decrease in variance, which consequently returns overconfident predictions in areas with little data.

To choose the weights β_i in the PoE model several heuristics have been put forward. The authors of [3] suggested the difference in differential entropy between the prior and posterior distribution of each expert, i.e. $\beta_i = \frac{1}{2}(\log \Sigma^{**} - \log \Sigma_i^*)$ where the $(\Sigma^{**})^{-1}$ is the prior precision of $p(y^*)$. This leads to more conservative predictions. To fix this issue, [5] suggested to choose simple uniform weights $\beta_i = \frac{1}{M}$, which provides better predictions.

Bayesian Committee Machine. The Bayesian committee machine [27] uses the Gaussian process prior $p(y^*)$ for the aggregation step and assumes conditional independence between experts, i.e. $\mathcal{D}_i \perp\!\!\!\perp \mathcal{D}_j | y^*$ for two experts i and j. To mitigate the effect of weak experts on aggregation, especially in regions with few data points, [5] proposed the robust Bayesian committee machine (RBCM), which added importance weights β_i to the model. The distributed predictive distribution of this family of models can be written as:

$$p(y^*|\mathcal{D}, X^*) = \frac{\prod_{i=1}^{M} p_i^{\beta_i}(y^*|\mathcal{D}_i, X^*)}{p^{\sum_{i=1}^{M} \beta_i - 1}(y^*)}.$$

Its distribution is proportional to a Gaussian distribution with mean and precision, respectively:

$$\mu_D^* = \Sigma_D^* \sum_{i=1}^{M} \beta_i (\Sigma_i^*)^{-1} \mu_i^*, \quad (\Sigma_D^*)^{-1} = \sum_{i=1}^{M} \beta_i (\Sigma_i^*)^{-1} + (1 - \sum_{i=1}^{M} \beta_i)(\Sigma^{**})^{-1},$$

where the $(\Sigma^{**})^{-1}$ is the prior precision of $p(y^*)$. The general choice of the weights is the difference in differential entropy between the prior $p(y^*|X^*)$ and the posterior $p(y^*|\mathcal{D}, X^*)$, i.e. $\beta_i = \frac{1}{2}(\log \Sigma^{**} - \log \Sigma_i^*)$.

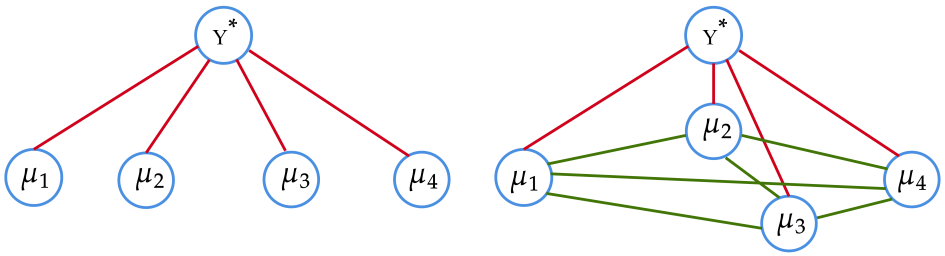
The most recent model in this family is the generalized robust Bayesian committee machine (GRBCM) [17]. It introduces a base (global) expert and considers the covariance between the base and other local experts. For a global expert M_b and a base partition D_b, the predictive distribution of GRBCM is

$$p(y^*|\mathcal{D}, X^*) = \frac{\prod_{i=2}^{M} p_{bi}^{\beta_i}(y^*|\mathcal{D}_{bi}, X^*)}{p_b^{\sum_{i=2}^{M}\beta_i - 1}(y^*|\mathcal{D}_b, X^*)}, \tag{11}$$

where the $p_b(y^*|\mathcal{D}_b, X^*)$ is the predictive distribution of M_b, and $p_{bi}(y^*|\mathcal{D}_{bi}, X^*)$ is the predictive distribution of an expert trained on the data set $\mathcal{D}_{bi} = \{\mathcal{D}_b, \mathcal{D}_i\}$. The base partition is randomly selected, while the remaining experts can be chosen through a random or disjoint partitioning strategy. It is noteworthy that, for M experts and m_0 data points per expert, the GRBCM operates based on $M - 1$ experts with $2m_0$ data points per expert. Therein lies the main difference between GRBCM and the other distributed GPs, which use m_0 data points per expert only. Since GRBCM assigns more data points to the experts, it trains experts on more informative subsets.

B. Computational Graphs of Aggregation Strategies

Figure 2 depicts the computational graphs of both strategies. Figure 2(a) reveals the aggregation based on conditional independence assumption between experts $\{\mu_1, \mu_2, \mu_3, \mu_4\}$. It means two local experts mu_i and mu_j are connected only via the target variable y^*, i.e. $mu_i \perp\!\!\!\perp mu_j \mid y^*$. However, this assumption is often violated in realistic conditions and the aggregation can lead to a sub-optimal solution. On the other hand, Figure 2(b) represents an aggregation with dependent experts where the interactions between experts show the dependencies.



(*a*) Independent Experts (*b*) Dependent Experts

Figure 2: **Computational graphs** of different aggregation strategies. (a) Conditional-Independent based aggregation where there is no interaction between the local experts. (b) An aggregation based on the conditional dependency between local experts where their interactions have not been excluded.

C. Proof of Proposition 1

Proof The proof is straightforward. We need to show that $Var(y-\beta\mu^*)-Var(y-y_A^*)$ is positive semi-definite for all linear unbiased predictors $\beta\mu^*$. To do that, we extend the $Var(y-\beta\mu^*)$:

$$Var(y-\beta\mu^*)=Var(y-y_A^*+y_A^*-\beta\mu^*)=Var(y-y_A^*)+var(y_A^*-\beta\mu^*)+2cov(y-y_A^*,y_A^*-\beta\mu^*).$$

Now, we show $cov(y-y_A^*,C\mu^*)=0,\forall C$.

$$cov(y-y_A^*,C\mu^*)=cov(y,C\mu^*)-cov(y_A^*,C\mu^*)=\Sigma_{y^*\mu^*}C^T-\Sigma_{y^*\mu^*}\Sigma_{\mu^*}^{-1}\Sigma_{\mu^*}C^T=0.$$

Therefore, $cov(y-y_A^*,y_A^*-\beta\mu^*)=0$ where in this case, $C=\Sigma_{y^*\mu^*}^T\Sigma_{\mu^*\mu^*}^{-1}-\beta$. It means

$$Var(y-\beta\mu^*)-Var(y-y_A^*)=var(y_A^*-\beta\mu^*)\geq 0$$

because $var(y_A^*-\beta\mu^*)$ is positive semi-definite variance matrix. ■

D. Computational cost of EMGGM and NPAE

Both EMGGM and NPAE use dependent experts. However, there are two major differences between them. First, NPAE needs all training and test data points during aggregation. Let $\Gamma_i=k_{i*}^T(K_i+\sigma^2I)^{-1}$. For a test point x^*, the pointwise covariance between experts i and j in NPAE, $K(x^*)_{ij}$, can be extended using (2) as

$$K(x^*)_{ij}=cov(\mu_i^*(x^*),\mu^*(x^*)_j)=Cov(\Gamma_iy_i,\Gamma_jy_j)=\Gamma_iCov(y_i,y_j)\Gamma_j^T=\Gamma_ik(x_i,x_j)\Gamma_j^T.$$

Therefore, all auto-covariance $k(x_i,x_i)$ and cross-covariance $k(x_i,x_j)$ matrices are required for NPAE aggregation which raises the storage costs.

Second, both aggregation methods have a $\mathcal{O}(M^3)$ calculation in each iteration, the inverse of $M\times M$ matrix in NPAE and GLasso in the proposed method. NPAE should do this costly calculation at each test point and therefore it is not efficient for large data sets. However, the proposed model can converge after a small number of iterations. When $R\ll n_t$, the proposed method is much faster than NPAE. Although the conventional GLasso for network learning is a costly method $\mathcal{O}(M^3)$, there are newer faster methods to learn a GGM that can be used instead of the GLasso, see [29, 30, 32]. For instance, the FST model [32] reduces the computational complexity of sparse Gaussian Graphical Model to a much lower order of magnitude $\big(\mathcal{O}(M^2)\big)$.

E. Gaussian Assumption

The normality assumption for joint distribution is not a restrictive assumption. In practice, we can relax this assumption and consider random variables without resorting to multi-dimensional Gaussian distribution. As a semiparametric generalization for continuous variables, authors in [12, 16] introduced the nonparanormal graphical model where it is assumed that the variables follow a Gaussian graphical model only after some unknown smooth monotone transformations on each of them. [20] considered Bayesian inference in nonparanormal graphical models by putting priors on the unknown transformations through a random series based on B-splines.

On the other hand, nonparametric methods can be used for functional graphical models. Authors in [15] and [25] exerted additive conditional independence and functional principal components to learn a graphical model when observations on vertices are functions. This gives the result that the proposed strategy can be considered as a general ensemble model, and not only for the local approximation GPs.

F. Latent Variable GGMs

Latent variable GGMs (LVGGMs) are used to estimate the distribution of the observed variables with respect to some latent variables. GGMs with latent variables have been widely considered over the past decade. Authors in [4] proposed *Low-Rank Plus Sparse Decomposition* (LR+SD), a regularized maximum likelihood approach to estimate Ω via convex optimization. The precision matrix in LR+SD contains two terms: sparse structure Ω_{μ^*} and the low-rank terms $L^* = \Omega_{\mu^* y^*}\Omega^{-1}_{y^* y^*}\Omega_{y^* \mu^*}$. The precision matrix in this form is $\Omega = \Omega_{\mu^*\mu^*} - L^*$. The log-likelihood can be expressed in terms of the $S_{\mu^*\mu^*}$, $\Omega_{\mu^*\mu^*}$, and L^*:

$$\mathcal{L}(\Omega_{\mu^*\mu^*}, L^*; S_{\mu^*\mu^*}) = log|\left(\Omega_{\mu^*\mu^*} - L^*\right)| - trace\left(S_{\mu^*\mu^*}\left(\Omega_{\mu^*\mu^*} - L^*\right)\right). \tag{12}$$

Essentially, it is a misspecified optimization problem because the precision matrix is the sum of two matrices. However, if $\Omega_{\mu^*\mu^*}$ is sparse and there are few latent variables, it is possible to decompose the precision matrix into its summands [2, 4].

$$\left(\hat{\Omega}_{\mu^*\mu^*}, \hat{L}^*\right) = \underset{\Omega_{\mu^*\mu^*}, L^* \in \mathcal{R}^{M\times M}}{\arg\min} - \mathcal{L}(\Omega_{\mu^*\mu^*}, L^*; S_{\mu^*\mu^*}) + \lambda\left(\gamma\left\|\Omega_{\mu^*\mu^*}\right\|_1 + \left\|L^*\right\|_*\right) \tag{13}$$

$$\text{such that} \quad \Omega_{\mu^*\mu^*} - L^* \succ 0,\ L^* \succeq 0 \tag{14}$$

such that $\Omega_{\mu^*} - L^* \succ 0,\ L^* \succeq 0$. Here, $\lambda > 0$ and $\gamma > 0$ are tuning parameters for sparsity and low rankness, and $\left\|L^*\right\|_*$ denotes the nuclear norm of L^* (i.e. the sum of its singular values).

To speeding up the LR+SD model, [30] proposed a non-convex optimization model and showed that it is orders of magnitude faster than the convex relaxation-based methods. Author in [31] proposed a direct approach via Expectation-Maximization algorithm which converts LR+SD model to a conventional GGM. Here, we modified this approach and proposed EMGGM.

However, LR+SD model has been developed to estimate the marginal distribution of observed variables, i.e. $p(\mu^*) = \int p(\mu^*, y^*)dy^*$, while the desired predictive distribution is the conditional distribution of y^* given local experts' predictions $p(y^*|\mu^*)$. Hence, further work could consider the modified form of the log-likelihood in (12) and the convex optimization in (13) to estimate the aggregate estimator y^*_A in (3) via a convex or non-convex optimization problem.