Scalable Gaussian Process Regression Networks

Shibo Li, Wei Xing, Robert Mike Kirby, Shandian Zhe

School of Computing, University of Utah

{shibo, kirby, zhe}@cs.utah.edu, wxing@sci.utah.edu

Abstract

Gaussian process regression networks (GPRN) are powerful Bayesian models for multi-output regression, but their inference is intractable. To address this issue, existing methods use a fully factorized structure (or a mixture of such structures) over all the outputs and latent functions for posterior approximation, which, however, can miss the strong posterior dependencies among the latent variables and hurt the inference quality. In addition, the updates of the variational parameters are inefficient and can be prohibitively expensive for a large number of outputs. To overcome these limitations, we propose a scalable variational inference algorithm for GPRN, which not only captures the abundant posterior dependencies but also is much more efficient for massive outputs. We tensorize the output space and introduce tensor/matrix-normal variational posteriors to capture the posterior correlations and to reduce the parameters. We jointly optimize all the parameters and exploit the inherent Kronecker product structure in the variational model evidence lower bound to accelerate the computation. We demonstrate the advantages of our method in several real-world applications.

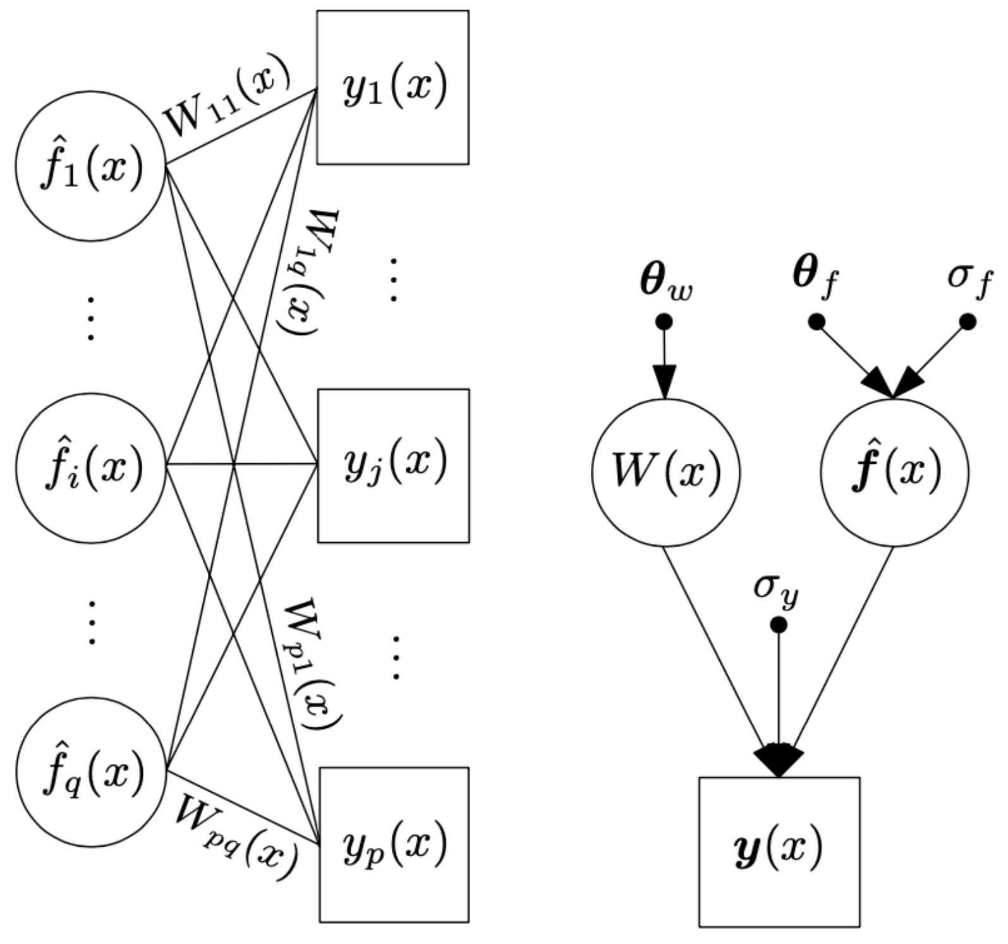
Introduction

High Dimensional Output Regression: to learn a regression function given training data,

$$\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots (\mathbf{x}_N, \mathbf{y}_N)\}$$

- Each $\mathbf{y}_n$ is a D dimensional vector
- D could be very large, e.g., 10^3 to 10^6

Gaussian Process Regression Network (Wilson et al. 2012) :



(Wilson et al. 2012)

- Nonstationary, highly flexible
- Analogy of the neural network output layer
- Both latent inputs and weights are GPs

Contributions

Issue of Current Inference Approaches for GPRN:

- Markov Chain Monte Carlo (Wilson et al. 2012)

Pros: Asymptotically converge to the true posterior.

Cons: Inefficient and hard to diagnose the convergence with high dimensional output space

- Fully Factorized Variational Inference (Wilson et al. 2012)

$$q(\{\mathbf{w}_{ij}\}, \{\hat{\mathbf{f}}_k\}) = \prod_{k=1}^{K} q(\hat{\mathbf{f}}_k) \prod_{i=1}^{D} \prod_{j=1}^{K} q(\mathbf{w}_{ij})$$

ignores posterior correlation

- Nonparametric Variational Inference (Nguyen et al. 2013)

$$q(\mathbf{u}) = \frac{1}{Q} \sum_{j=1}^{Q} \mathcal{N}(\mathbf{u}|\boldsymbol{\mu}_j, v_j \mathbf{I})$$

over-simplified posterior correlation

Our Method: Structure Variational Inference

- Matrix Gaussian posterior: fully capture the posterior dependency of the latent functions

$$q(\mathbf{F}) = \mathcal{MN}(\mathbf{F}, \mathbf{M}, \boldsymbol{\Sigma}, \boldsymbol{\Omega}) = \mathcal{N}(\mathrm{vec}(\mathbf{F})|\mathrm{vec}(\mathbf{M}), \boldsymbol{\Sigma} \otimes \boldsymbol{\Omega})$$

- Tensor Normal posterior: fully capture the posterior dependency of all the weights

$$q(\mathcal{W}) = \mathcal{TN}(\mathcal{W}|\mathcal{U}, \boldsymbol{\Gamma}_1, \ldots, \boldsymbol{\Gamma}_{M+2})$$
$$= \mathcal{N}(\mathrm{vec}(\mathcal{W})|\mathrm{vec}(\mathcal{U}), \boldsymbol{\Gamma}_1 \otimes \ldots \otimes \boldsymbol{\Gamma}_{M+2})$$

- Then the variational posterior is given by

$$q(\mathcal{W}, \mathbf{F}) = q(\mathcal{W}) q(\mathbf{F})$$

Our Method (continued):

- Learning: Stochastic Variational Inference + Re-parameterization tricks

$$\mathcal{L} = -\mathrm{KL}\big(q(\mathcal{W})\|p(\mathcal{W})\big) - \mathrm{KL}\big(q(\mathbf{F})\|p(\mathbf{F})\big)$$
$$+ \mathbb{E}_q[\log p(\mathbf{Y}|\mathbf{X}, \mathcal{W}, \mathbf{F})]$$

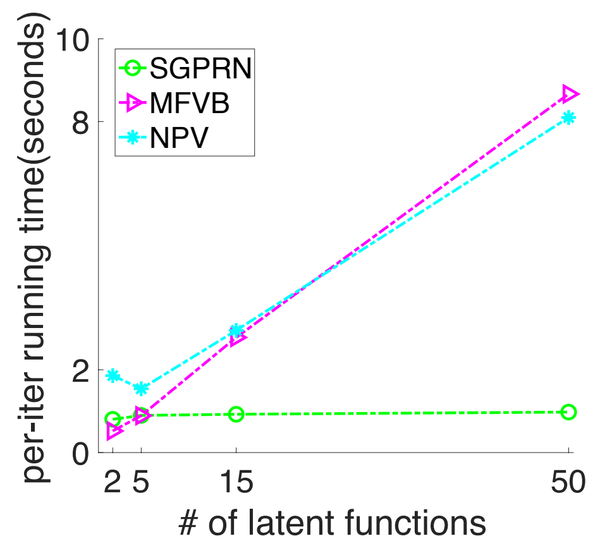
- Linear Complexity: $\mathcal{O}(NDK)$ when $D \gg \{N, K\}$

	Time Complexity
Fully Factorized VI	$\mathcal{O}(NK^2D)$
Non-parametric VI	$\mathcal{O}(QN^2KD)$
Our approach	$\mathcal{O}(N^3 + K^3 + Md^2 + NDK)$

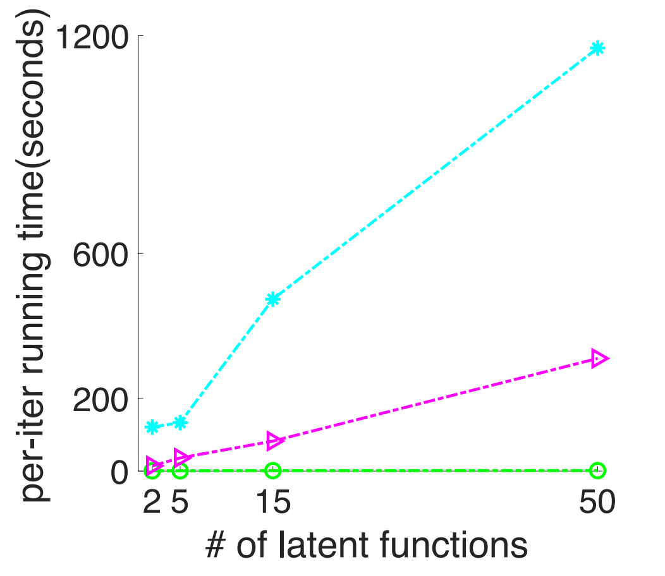
Experiments

Toy problems and time analysis:

Jura	
SGPRN	**0.5127**±0.002
MFVB	0.5237±0.001
NPV	0.6088±9e-06
Equity	
SGPRN	**2.5759e-05**±2e-07
MFVB	4.4530e-05±6e-07
NPV	4.7267e-05±9e-18
PM2.5	
SGPRN	**1.07089**±0.02
MFVB	1.3916±0.06
NPV	14.7101±0.14

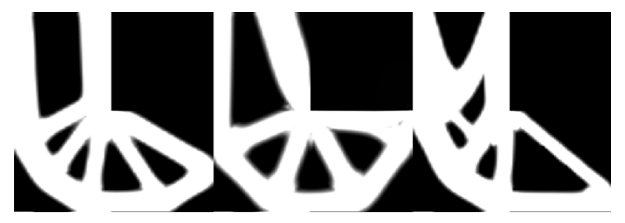


(per-iter time of Jura)

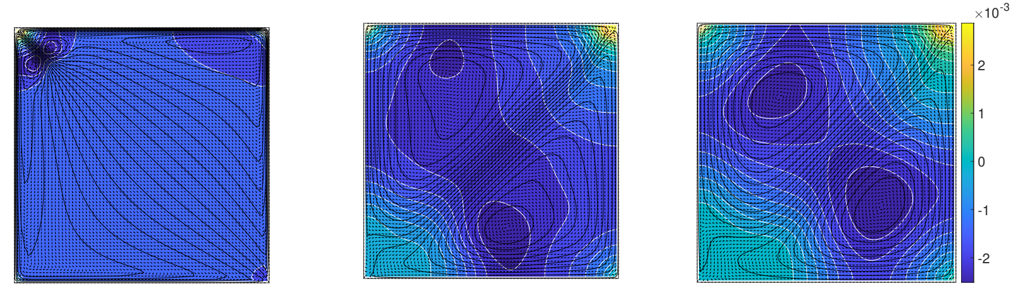


(per-iter time of PM2.5)

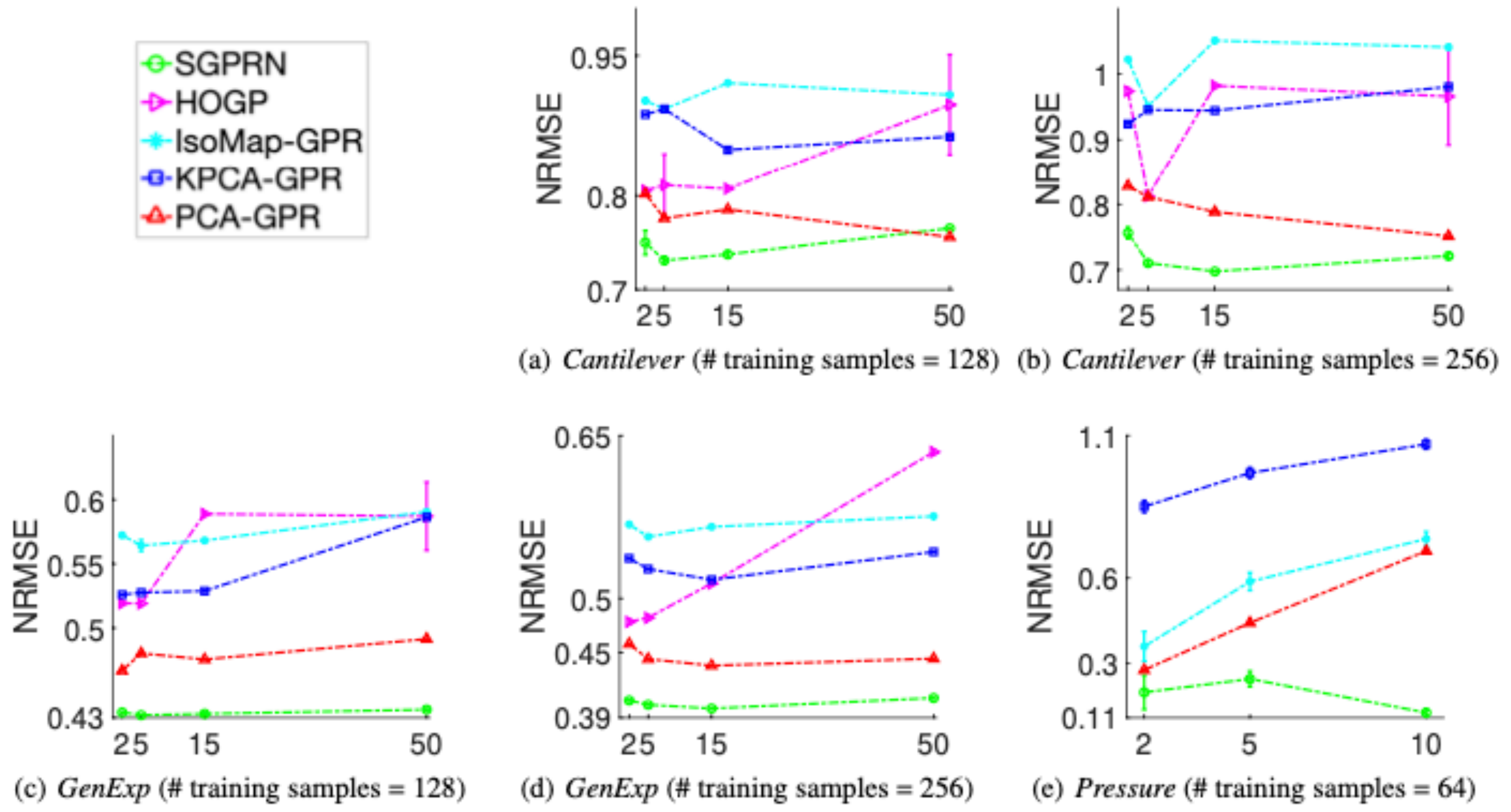
Intermediate and Large-scale problems:



(Example of Cantilever, D=3200)



(Example of Pressure t=1,5,10. D=1M)



(a) *Cantilever* (# training samples = 128) (b) *Cantilever* (# training samples = 256)

(c) *GenExp* (# training samples = 128) (d) *GenExp* (# training samples = 256) (e) *Pressure* (# training samples = 64)

(Comparisons of nRmse on intermediate and large-scale problems)