Embarrassingly Parallel Inference for Gaussian Processes

Michael Zhang, Sinead Williamson

The University of Texas at Austin
Department of Statistics and Data Sciences
College of Natural Sciences

Introduction

- A typical problem in statistics and machine learning is learning a latent function.
- Ex: Regression, classification, optimization.
- Learning a general, non-linear function is not a trivial task.

Gaussian Processes

- Gaussian processes (GP) are distributions over real-valued functions, $f : \mathbb{R}^D \to \mathbb{R}$ with mean function $m(\cdot)$ and covariance function $\Sigma(\cdot, \cdot)$
- A GP-distributed function evaluated at any finite set of points is multivariate normal:

$$f|X \sim \mathsf{N}\left(m(X), \Sigma(X, X')\right)$$

- Due to normal property, GPs are often used as priors for learning non-linear functions.
- Ex: Regression

$$f \sim \mathsf{GP}(0, \Sigma), \quad f|X \sim \mathsf{N}\left(0, \Sigma(X, X')\right), \quad Y|X, f, \sigma^2 \sim \mathsf{N}(f(X), \sigma^2 I)$$

Gaussian Process Inference

- Fitting GP models typically requires inferring hyperparameters Θ.
- Inference involves inverting the $N \times N$-dimensional covariance matrix, which costs $O(N^3)$.
- This cost of inversion prevents GPs from being used in large scale situations.

Sparse Gaussian Process Inference

- In general there are two approaches to scalable GP inference.
- Sparse GP methods introduce $M << N$ pseudo-inputs which are chosen to represent the function posterior.
- Benefits: Requires only inverting an $M \times M$ matrix ($O(NM^2)$ for regression).
- Drawbacks: Requires more pseudo-inputs with fast-varying functions.

Sparse Gaussian Process Inference

- Snelson and Ghahramani (2005) proposed FITC method to learn sparse inputs and hyperparameters jointly through optimization.
- Titsias (2009) derived a variational inference (VFE) method to learn sparse inputs. Hensman et al. (2013) developed SVI method to learn from mini-batches of data.
- From Bauer et al. (2016):

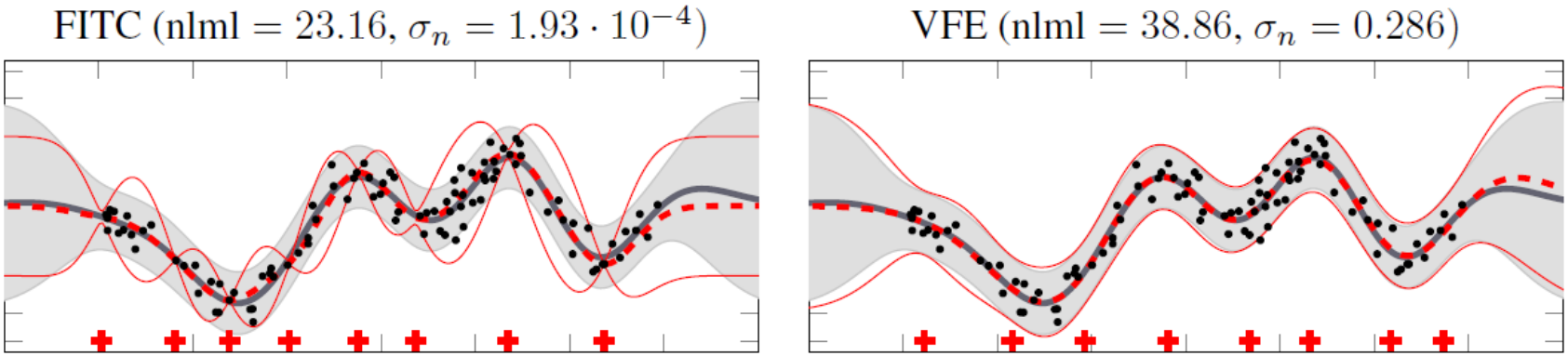


Figure 1: Behaviour of FITC and VFE on a subset of 100 data points of the Snelson dataset for 8 inducing inputs (red crosses indicate inducing inputs; red lines indicate mean and 2σ) compared to the prediction of the full GP in grey. Optimised values for the full GP: nlml $= 34.15$, $\sigma_n = 0.274$

Local Gaussian Process Methods

- Local GP inference divides the data into K partitions and fits a separate GP model for each partition. Equivalent to replacing full covariance matrix with block diagonal matrix.
- Benefits: We can easily model non-stationary functions and parallelize inference. Fitting K partitions on average scales $O(N^3/K^2)$.
- Drawbacks: Local methods ignore long-range correlations in the function, fixed partitions may lead to discontinuities between regions.

Local Gaussian Process Methods

- Mixture-of-experts models, like Bayesian Treed Gaussian Process (Gramacy and Lee, 2008), partitions the input space into independent GPs and averages over the partitioning.
- Product-of-experts models, like Robust Bayesian Committee Machine (Deisenroth and Ng, 2015), fits the data across independent GPs in parallel and perform predictions with weighted products of experts

Problem Statement

- We seek to address the problems in scalable GP inference of modeling short and long term correlations, as well as possible non-stationary elements.
- Sampling or averaging over partitions in local GP methods can solve these problem but is difficult and slow.
- We propose a method that is takes advantage of block-diagonal inversion convenience of local GP methods, but averages over partitions in parallel through importance sampling.

Importance Sampling

- Suppose we are interested in the integral:

$$\bar{f} = \int f(x)p(x)\,\mathrm{d}x$$

  but we cannot compute the integral easily.
- If we have a proposal distribution $q(x)$ we can approximate $\bar{f}$ with an unbiased estimator:

$$\bar{f} = \int \frac{f(x)p(x)}{q(x)} q(x)\,\mathrm{d}x \approx \frac{1}{J}\sum_{j=1}^{J} f(x^{(j)})\frac{p(x^{(j)})}{q(x^{(j)})}$$

  by drawing J samples from $q(x)$
- p/q is also known as the importance sampling weight, w

Importance Gaussian Process Sampler

- We explicitly assume input is distributed with a Normal-Inv. Wishart mixture model:

$$x_i \sim \text{Normal}(\mu_{z_i}, \Gamma_{z_i}), \quad (\mu_k, \Gamma_k) \sim \text{Normal-Inv. Wishart}(\cdot)$$

$$z_i \sim \text{Categorical}(\pi), \quad \pi \sim \text{Dirichlet}(\alpha)$$

- Each proposal for the IS is a draw from the marginalized distribution of the partition assignment:

$$Z|X \sim \int P(Z|X,\pi)P(\pi)\mathrm{d}\pi \int\int P(X|\mu,\Gamma)P(\mu,\Gamma)\,\mathrm{d}\mu\mathrm{d}\Gamma$$

Importance Gaussian Process Sampler

- After sampling J partition assignments, we fit K separate GP models to the partitioned data and infer GP hyperparameters Θ_k via MAP estimation.
- The IS weights each proposed model according to:

$$w_j \propto \frac{p(Z|X,Y)}{p(Z|X)} \propto \frac{p(X,Y|Z)p(Z)}{p(X|Z)p(Z)} = p(Y|X,Z)$$

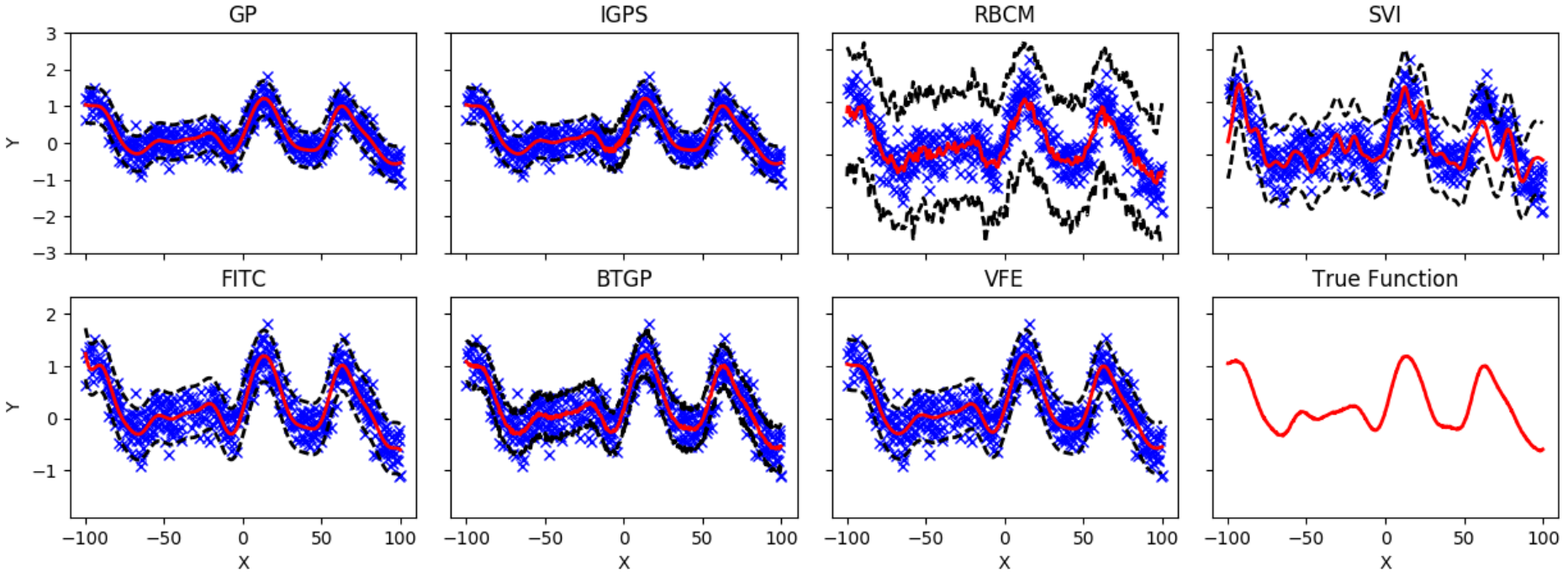
- Because we calculate self-normalized weights, IS has bias of $O(1/J)$.

Importance Gaussian Process Sampler

- Complexity of IGPS: $O(JN^3/K^2)$, for J importance samples.
- In contrast to the treed GP (Gramacy and Lee, 2008) we can average over the partitions quickly with comparable order of complexity
- Each proposal is independent so we can trivially distribute inference without repeated communication (embarrassingly parallel)
- After partitioning, we can fit each GP mixture independently–per thread complexity of $O(N^3/K^3)$
- By allowing each partition to have its own hyperparameters, we can also model non-stationary data easily.

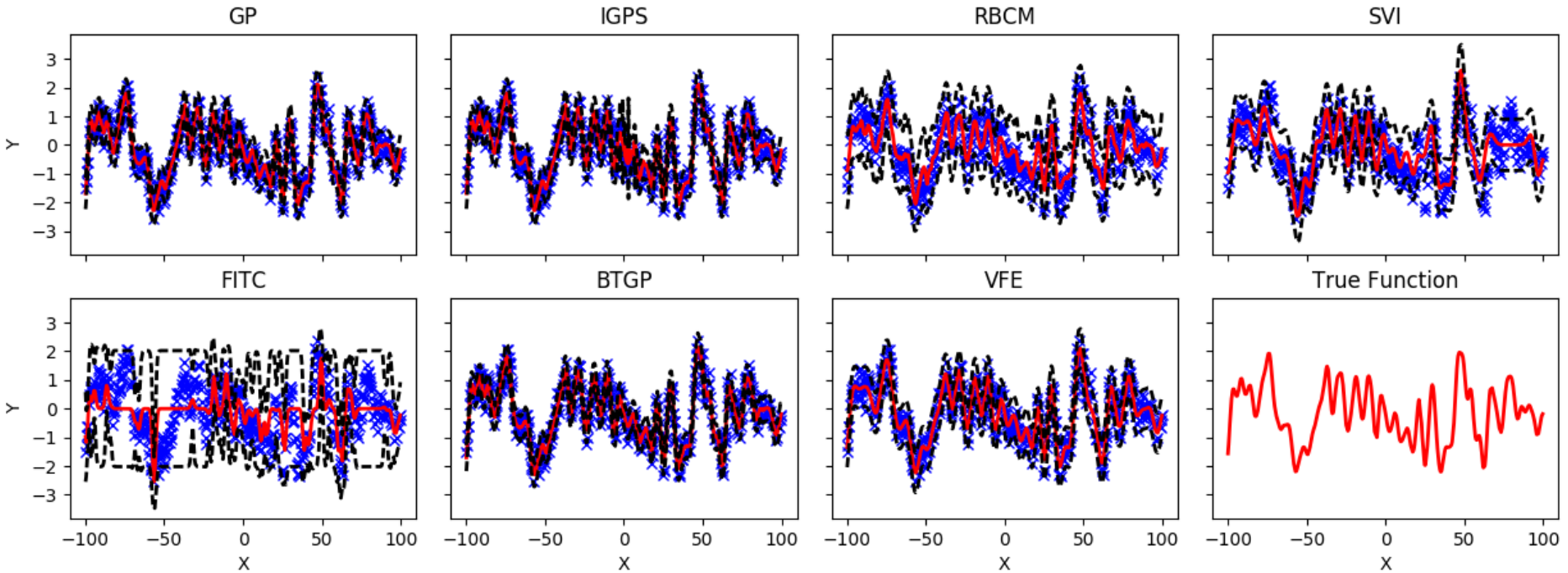
Results

Synthetic data, stationary, long length-scale, $N = 1000$, $K = 20$, $M = 50$



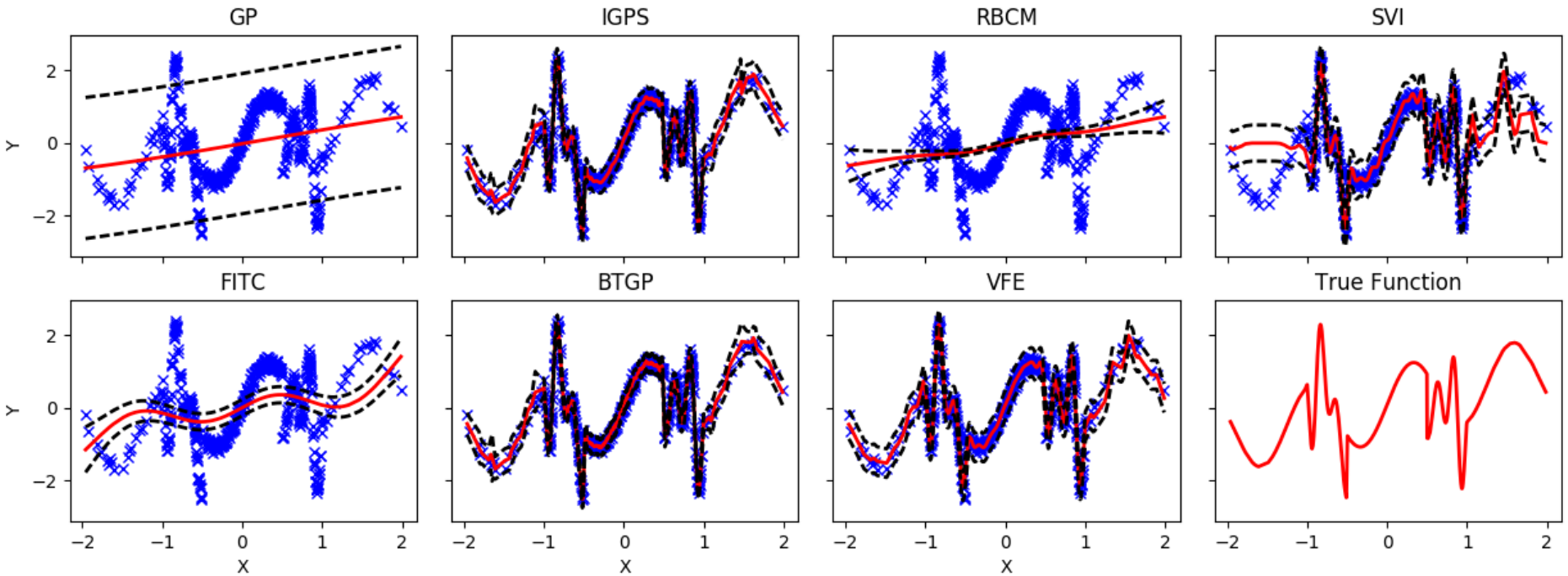
Results

Synthetic data, stationary, short length-scale, $N = 1000$, $K = 20$, $M = 50$



Results

Synthetic data, non-stationary, $N = 1000$, $K = 20$, $M = 50$



Results

- "High-dimensional" data set is synthetic 50-dimensional data set with inputs generated from GMM and output generated from GP.
- "Flight Delay" data set is real-world large scale data example.

Table: Log likelihoods obtained on the stationary, non-stationary and large-N regression tasks

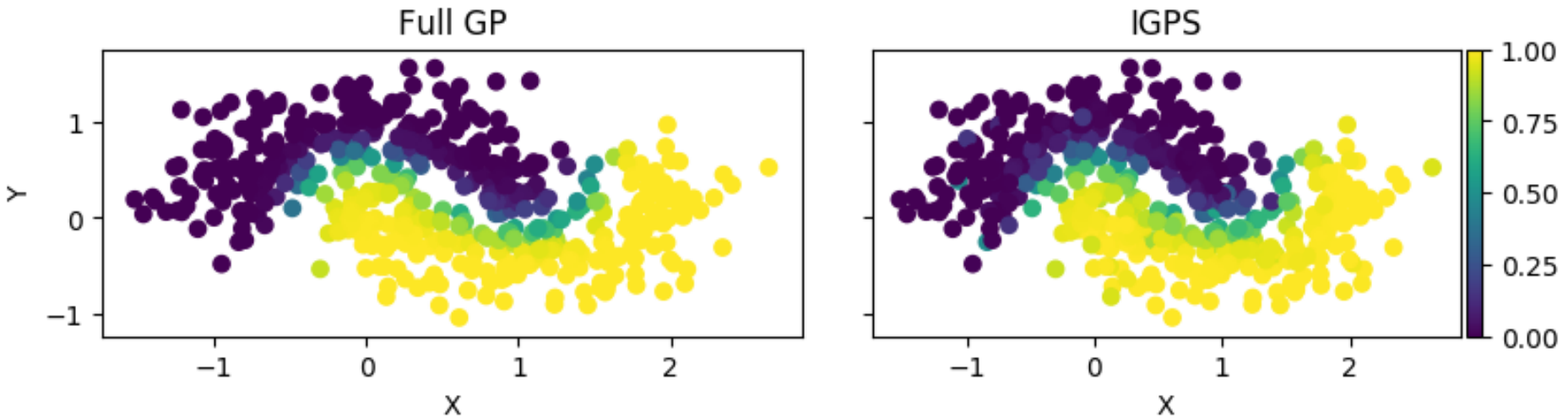
Data	GP	FITC	**IGPS**	BTGP	RBCM	SVI	DTC
Stationary, long lengthscale	-33.79	-35.28	-49.29	-80.15	-255.23	-444.43	-33.80
Stationary, short lengthscale	-31.32	-539.55	-43.15	-64.16	-280.08	-355.51	-125.92
Non-stationary	-711.57	-1.39e4	291.11	119.65	-1.17e5	-91.95	167.26
High-dimensional	-60.45	-108.71	-93.08	-147.53	-117.31	-141.00	-126.67
Flight Delay	x	x	-2.49e4	x	-7.13e7	-1.53e8	x

Results

Table: Classification log likelihood results (left) and AUC results (right)

Data	GP	IGPS	FITC
Pima	-128.79	-135.09	-128.61
Parkinsons	-17.00	-22.76	-28.42
WDBC	-15.50	-12.62	-18.01

Data	GP	IGPS	FITC
Pima	0.83	0.81	0.83
Parkinsons	0.86	0.93	0.88
WDBC	0.83	0.91	0.81



Conclusion

- GPs are nice models due to flexibility, but not scalable due to inversion of covariance matrix.
- Previous methods for scalable inference have issues modeling certain types of functions (short length scale, long length scale, non-stationarity)
- Our proposed method can capture general types of functions, inference can be carried out in embarrassingly parallel.
- Future work: GPU implementation, GP models beyond regression.

Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding Probabilistic Sparse Gaussian Process Approximations. *ArXiv e-prints*.

Cao, Y. and Fleet, D. J. (2014). Generalized product of experts for automatic and principled fusion of Gaussian process predictions. In *Modern Nonparametrics 3: Automating the Learning Pipeline workshop at NIPS*.

Deisenroth, M. and Ng, J. W. (2015). Distributed Gaussian processes. In *International Conference on Machine Learning*, pages 1481–1490.

Gramacy, R. B. and Lee, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*.

Rasmussen, C. E. and Ghahramani, Z. (2002). Infinite mixtures of Gaussian process experts. In *Neural Information Processing Systems*, pages 881–888.

Snelson, E. and Ghahramani, Z. (2005). Sparse Gaussian processes using pseudo-inputs. In *Neural Information Processing Systems*, pages 1257–1264.

Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, volume 5, pages 567–574.

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