Survey on Sparse Gaussian Processes for Big data

Sunny Kant
Roll No.- 15817740
CS698X (Course Project)

1 Problem Motivation

Gaussian Process takes $O(n^3)$ time for training and atleast $O(n)$ time during prediction, where n in number of training data points. So, for dataset of size of millions or billions, there is need to scale up the method. A solution to this is to infer posterior only from a subset of data without losing much accuracy and achieve computational efficiency.

2 Introduction

2.1 Gaussian Process

- Gaussian Process defines distributions over functions. GP models are non-parametric probabilistic models for Bayesian supervised learning.

- Assuming training data $\{x_n, y_n\}_{n=1}^N \cdot x_n \in R^D, y_n \in R$.
  and, $y_n = f(x_n) + \epsilon_n$, where $\epsilon_n \sim \mathcal{N}(\epsilon_n|0, \beta)$.
  $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \beta\mathbf{I}_N)$ Assuming prior $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$

  $$p(y_*|y) = \mathcal{N}\left(y_*|\mu_*, \sigma_*^2\right) \mu_* = \mathbf{k}_*^\top (\mathbf{K} + \beta \mathrm{I_N})^{-1} y \sigma_*^2 = \kappa(x_*, x_*) + \beta - \mathbf{k}_*^\top (\mathbf{K} + \beta \mathrm{I_N})^{-1} \mathbf{k}_*$$

  where x_* is test data point.

2.2 Sparse Gaussian Process

- Let $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^m, \mathbf{u} = [f(z_1), f(z_2), ..., f(z_n)]$ be the set of inducing variables.

  $$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}\left(\mathbf{y}|\mathbf{f}, \beta^{-1}\mathbf{I}\right) p(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K}_{mm}) p(\mathbf{f}|\mathbf{u}) = \mathcal{N}\left(\mathbf{f}|\mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{u}, \tilde{\mathbf{K}}\right)$$

  where, $\tilde{\mathbf{K}} = \mathbf{K}_{nn} - \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}$

3 Literature review

3.1 Greedy Selection of points

- Seeger. et. al. had used greedy selection of new point to the active set (inducing points set) based on the approximated information gain score or differential entropy score of each data point x_i.

- The process of selection was as fast as selecting active set at random.

3.2 Pseudo Inputs

- $p(y|\mathbf{x}, \overline{\mathbf{X}}, \overline{\mathbf{f}}) = \mathcal{N}\left(y|\mathbf{k}_\mathbf{x}^\top \mathbf{K}_M^{-1}\overline{\mathbf{f}}, K_{\mathbf{xx}} - \mathbf{k}_\mathbf{x}^\top \mathbf{K}_M^{-1}\mathbf{k}_\mathbf{x} + \sigma^2\right)$
  $p(\mathbf{y}|\mathbf{X}, \overline{\mathbf{X}}, \overline{\mathbf{f}}) = \prod_{n=1}^N p\left(y_n|\mathbf{x}_n, \overline{\mathbf{X}}, \overline{\mathbf{f}}\right) = \mathcal{N}\left(\mathbf{y}|\mathbf{K}_{NM}\mathbf{K}_M^{-1}\overline{\mathbf{f}}, \mathbf{\Lambda} + \sigma^2\mathbf{I}\right)$

where $\mathbf{\Lambda} = diag(\lambda), \lambda_n = K_{nn} - \mathbf{k}_n^\top \mathbf{K}_M^{-1} \mathbf{k}_n$, and $\left[\mathbf{K}_{NM}\right]_{nm} = K\left(\mathbf{x}_n, \overline{\mathbf{x}}_m\right)$

Assuming prior on pseudo inputs as $p(\overline{\mathbf{f}}|\overline{\mathbf{X}}) = \mathcal{N}\left(\overline{\mathbf{f}}|\mathbf{0}, \mathbf{K}_M\right)$

Posterior distribution over pseudo targets $\overrightarrow{\mathbf{f}}$
$p(\overline{\mathbf{f}}|\mathcal{D}, \overline{\mathbf{X}}) = \mathcal{N}\left(\overline{\mathbf{f}}|\mathbf{K}_M \mathbf{Q}_M^{-1} \mathbf{K}_{MN} \left(\mathbf{\Lambda} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{y}, \mathbf{K}_M \mathbf{Q}_M^{-1} \mathbf{K}_M\right)$
,where $\mathbf{Q}_M = \mathbf{K}_M + \mathbf{K}_{MN} \left(\mathbf{\Lambda} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{K}_{NM}$

- Predictive distribution after integrating out $\tilde{\mathbf{f}}$

  $p\left(y_*|\mathbf{x}_*, \mathcal{D}, \overline{\mathbf{X}}\right) = \int d\overline{\mathbf{f}} p\left(y_*|\mathbf{x}_*, \overline{\mathbf{X}}, \overline{\mathbf{f}}\right) p(\overline{\mathbf{f}}|\mathcal{D}, \overline{\mathbf{X}}) = \mathcal{N}\left(y_*|\mu_*, \sigma_*^2\right)$ where,
  $\mu_* = \mathbf{k}_*^\top \mathbf{Q}_M^{-1} \mathbf{K}_{MN} \left(\mathbf{\Lambda} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{y}$
  $\sigma_*^2 = K_{**} - \mathbf{k}_*^\top \left(\mathbf{K}_M^{-1} - \mathbf{Q}_M^{-1}\right) \mathbf{k}_* + \sigma^2$

- $$p(\mathbf{y}|\mathbf{X}, \overline{\mathbf{X}}, \Theta) = \int d\overline{f} p(\mathbf{y}|\mathbf{X}, \overline{\mathbf{X}}, \overline{\mathbf{f}}) p(\overline{\mathbf{f}}|\overline{\mathbf{X}}) = \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}_{NM}\mathbf{K}_M^{-1}\mathbf{K}_{MN} + \mathbf{\Lambda} + \sigma^2\mathbf{I}\right)$$
- Maximizing marginal likelihood with respect to $\{\overline{\mathbf{X}}, \Theta\}$ will give optimal values of the pseudo inputs.[4]
- Predictive distribution calculations would be of $\mathcal{O}\left(M^2 N\right)$ *cost* .

3.3 FITC and PITC Approximation

Consider:
$q_{\mathrm{FITC}}(\mathbf{f}|\mathbf{u}) = \prod_{i=1}^n p\left(f_i|\mathbf{u}\right) = \mathcal{N}\left(K_{\mathbf{f},\mathbf{u}} K_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, diag\left[K_{\mathbf{f},\mathbf{f}}\right]\right)$,
$and, q_{\mathrm{FTC}}\left(f_*|\mathbf{u}\right) = p\left(f_*|\mathbf{u}\right)$

$$q_{FTTC}\left(\mathbf{f}, f_*\right) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} Q_{\mathbf{f},\mathbf{f}} - diag\left[Q_{\mathbf{f},\mathbf{f}} - K_{\mathbf{f},\mathbf{f}}\right] & Q_{\mathbf{f},*} \\ Q_{*,\mathbf{f}} & K_{*,*} \end{bmatrix}\right)$$

$q_{\mathrm{FTC}}\left(f_*|\mathbf{y}\right) = \mathcal{N}\left(Q_{*,\mathrm{f}}\left(Q_{\mathrm{f,f}} + \Lambda\right)^{-1} \mathbf{y}, K_{*,*} - Q_{*,\mathrm{f}}\left(Q_{\mathrm{f,f}} + \Lambda\right)^{-1} Q_{\mathbf{f},*}\right)$
$= \mathcal{N}\left(K_{*,\mathrm{u}} \Sigma K_{\mathrm{u,f}} \Lambda^{-1} \mathbf{y}, K_{*,*} - Q_{*,*} + K_{*,\mathrm{u}} \Sigma K_{\mathrm{u},*}\right)$ where, $\Sigma = \left(K_{\mathrm{u,u}} + K_{\mathrm{u,f}} \Lambda^{-1} K_{\mathrm{f,u}}\right)^{-1}$ and $\Lambda = diag\left[K_{\mathrm{f,f}} - Q_{\mathrm{f,f}} + \sigma_{\mathrm{noise}}^2 I\right]$

- Similarly, instead of fully independent, block level independence can also be considered. [3]
  $q_{\mathrm{PTTC}}(\mathbf{f}|\mathbf{u}) = \mathcal{N}\left(K_{\mathrm{f},\mathbf{u}} K_{u,\mathbf{u}}^{-1} \mathbf{u}, blockdiag\left[K_{\mathrm{f,f}} - Q_{\mathrm{f,f}}\right]\right)$
- Covariance matrix in both are of order of m. So, Computational cost is $O(nm^2)$ initially and $O(m)$ and $O(m^2)$ per test case during prediction time.

3.4 Variational Learning of inducing variables (VFE)

- Variational lower bound of true log marginal likelihood is-

$$F_V\left(X_m, \phi\right) = \int p\left(\mathbf{f}|\mathbf{f}_m\right) \phi\left(\mathbf{f}_m\right) \log \frac{p(\mathbf{y}|\mathbf{f}) p\left(\mathbf{f}_m\right)}{\phi\left(\mathbf{f}_m\right)} d\mathbf{f} d\mathbf{f}_m$$

- Maximizing following above likelihood for optimal ϕ analytically, we can get,

$$F_V(X_m) = \log\left[N\left(\mathbf{y}|\mathbf{0}, \sigma^2 I + K_{nm} K_{mm}^{-1} K_{mn}\right)\right] - \frac{1}{2\sigma^2} Tr\left[K_{nn} - K_{nm} K_{mm}^{-1} K_{mn}\right]$$

- Maximizing the bound wrt to $\left(X_m, \sigma^2, \theta\right)$, we can get optimal values.
- The first term encourages fitting the data y.The second trace term says to minimize the total variance of $p\left(\mathbf{f}|\mathbf{f}_m\right)$.
- The method can be seen as EM, where, E step we add one point into the inducing set and at the M step we update the hyperparameters.[5]

3.5 Variational EM

- Since, $p(\mathbf{f}, \mathbf{u}|\mathbf{y}) = p(\mathbf{f}|\mathbf{u}, \mathbf{y})p(\mathbf{u}|\mathbf{y})$
  Approximate posterior is: $q(\mathbf{f}, \mathbf{u}|\mathbf{y}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u})$
  Variational distribution can be considered as mixture of Gaussians $q(\mathbf{u}|\lambda) = \sum_{k=1}^{K} \pi_k q_k(\mathbf{u}|\mathbf{m}_k, \mathbf{S}_k) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{Q} \mathcal{N}(\mathbf{u}_j; \mathbf{m}_{kj}, \mathbf{S}_{kj})$

$$\log p(\mathbf{y}) \geq \mathcal{L}_{\text{elbo}} = \int q(\mathbf{u}|\lambda)p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) df \, \mathbf{du} - \mathbf{KL}(q(\mathbf{u}|\lambda)\|p(\mathbf{u}))$$

- Parameters of co-variance function and variational parameters $\lambda = \{\pi_k, \mathbf{m}_{kj}, \mathbf{S}_{kj}\}$ are variational-EM alternating optimization framework [1].

3.6 MCMC for Sparse GPs

- Assuming $\mathbf{f}^*$ to be function value at test points. θ to be co variance function parameters . and $q(\mathbf{u}, \theta)$ be variational posterior.
  $p(\mathbf{f}^\star|\mathbf{y}) = p(\mathbf{f}^\star|\mathbf{f}, \theta)\, p(\mathbf{f}, \theta|\mathbf{y}) \mathrm{d}\theta \mathrm{d}\mathbf{f}$
  $q(\mathbf{f}^\star) = p(\mathbf{f}^\star|\mathbf{u}, \theta)\, q(\mathbf{u}, \theta) \mathrm{d}\theta \mathrm{d}\mathbf{u}$

$$\mathcal{K}\mathrm{KL}\left[q(\mathbf{f}^\star, \mathbf{f}, \mathbf{u}, \theta) \| p(\mathbf{f}^\star, \mathbf{f}, \mathbf{u}, \theta|\mathbf{y})\right]$$

$$= -\mathrm{E}_{q(\mathbf{f}^*, \mathbf{f}, \mathbf{u}, \theta)}\left[\log \frac{p(\mathbf{f}^\star|\mathbf{u}, \mathbf{f}, \theta) p(\mathbf{u}|\mathbf{f}, \theta) p(\mathbf{f}, \theta|\mathbf{y})}{p(\mathbf{f}^\star|\mathbf{u}, \mathbf{f}, \theta) p(\mathbf{f}|\mathbf{u}, \theta) q(\mathbf{u}, \theta)}\right]$$

$$= -\mathrm{E}_{q(\mathbf{f}, \mathbf{u}, \theta)}\left[\log \frac{p(\mathbf{u}|\mathbf{f}, \theta) p(\mathbf{f}|\theta) p(\theta) p(\mathbf{y}|\mathbf{f}) / p(\mathbf{y})}{p(\mathbf{f}|\mathbf{u}, \theta) q(\mathbf{u}, \theta)}\right]$$

$$= -\mathrm{E}_{q(\mathbf{f}, \mathbf{u}, \theta)}\left[\log \frac{p(\mathbf{u}|\theta) p(\theta) p(\mathbf{y}|\mathbf{f})}{q(\mathbf{u}, \theta)}\right] + \log p(\mathbf{y})$$

- Rearranging the terms in above equation, we can get-

$$\mathcal{K} = \mathrm{KL}\left[q(\mathbf{u}, \theta) \| \frac{p(\mathbf{u}|\theta) p(\theta) E_{p(\mathbf{f}|\mathbf{u}, \theta)}[\log p(\mathbf{y}|\mathbf{f})]}{C}\right] - \log C + \log p(\mathbf{y})$$

  where C is constant, normalizing the distribution.

- Minimizing the above KL divergence, optimal variational posterior can be derived as-

  $\log \hat{q}(\mathbf{u}, \theta) = E_{p(\mathbf{f}|\mathbf{u}, \theta)}[\log p(\mathbf{y}|\mathbf{f})] + \log p(\mathbf{u}|\theta) + \log p(\theta) - \log C$ This optimal distribution doesnot take any particular distribution in gereral. We use MCMC for sampling, which requires $\mathcal{O}\left(NM^2\right)$ computations.

- The positions of inducing points are initialized using k-means clustering of data. A Gaussian to the posterior can be fit and optimisation like AdaDelta or LBFGS can be used. Furthuer, optimized results can be used in HMC for furthur optimisation.

3.7 SVI for GPs

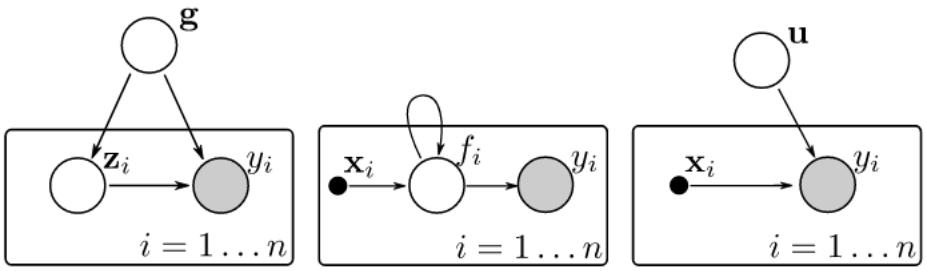
Bound

$$\log p(\mathbf{y}|\mathbf{u}) = \log \langle p(\mathbf{y}|\mathbf{f}) \rangle_{p(\mathbf{f}|\mathbf{u})} \geq \langle \log p(\mathbf{y}|\mathbf{f}) \rangle_{p(\mathbf{f}|\mathbf{u})} \mathcal{L}_1$$

$$\log p(\mathbf{y}|\mathbf{X}) = \log \int p(\mathbf{y}|\mathbf{u}) p(\mathbf{u}) \mathrm{d}\mathbf{u} \geq \log \int \exp\{\mathcal{L}_1\} p(\mathbf{u}) \mathrm{d}\mathbf{u} \mathcal{L}_2$$

Global variables are required to apply SVI for GP regression. Assume $\mathbf{u}$ to be global variables and $q(\mathbf{u})$ to be variational distribution of $\mathbf{u}$,, we derive a lower bound on $\mathcal{L}_2$.

$$\log p(\mathbf{y}|\mathbf{X}) \geq \langle \mathcal{L}_1 + \log p(\mathbf{u}) - \log q(\mathbf{u}) \rangle_{q(\mathbf{u})} \mathcal{L}_3$$



(a) Requirements for SVI (b) Gaussian Process regression (c) Variational GP regression

- Assume $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{m}, \mathbf{S})$

$$\mathcal{L}_3 = \sum_{i=1}^{n} \left\{ \log \mathcal{N}\left(y_i|\mathbf{k}_{mn}^{-1}\mathbf{m}, \beta^{-1}\right) - \frac{1}{2}\beta\widetilde{k}_{i,i} - \frac{1}{2}tr\left(\mathbf{S}\mathbf{\Lambda}_i\right) \right\} - \mathrm{KL}[q(\mathbf{u})\|p(\mathbf{u})]$$

  This can be written as a sum of n terms, each corresponding to one input-output pair.

$$\frac{\partial \mathcal{L}_3}{\partial \mathbf{m}} = \beta \mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}\mathbf{y} - \mathbf{\Lambda}\mathbf{m}, \frac{\partial \mathcal{L}_3}{\partial \mathbf{S}} = \frac{1}{2}\mathbf{S}^{-1} - \frac{1}{2}\mathbf{\Lambda}$$

  where, $\Lambda_i = \beta \mathbf{K}_{mm}^{-1}\mathbf{k}_{mn}\mathbf{k}_{mn}^{\top}\mathbf{K}_{mm}^{-1}$

- SVI takes steps in the direction of approximate natural gradient.$\widetilde{\mathrm{g}}(\theta) = G(\theta)^{-1}\frac{\partial \mathcal{L}}{\partial \theta}$ where $G(\theta)^{-1}$ is inverse Fisher information. Suppose, $\theta_1 = \mathbf{S}^{-1}\mathbf{m}, \theta_2 = -\frac{1}{2}\mathbf{S}^{-1}$ and $\eta_1 = \mathbf{m}, \eta_2 = \mathbf{m}\mathbf{m}^{\top} + \mathbf{S}$

  $\widetilde{\mathbf{g}}(\theta) = G(\theta)^{-1}\frac{\partial \mathcal{L}_3}{\partial \theta} = \frac{\partial \mathcal{L}_3}{\partial \eta}$

Using $\theta_{(t+1)} = \theta_{(t)} + \ell\frac{\mathrm{d}\mathcal{L}_3}{\mathrm{d}\eta}$.

$$\theta_{2(t+1)} = -\frac{1}{2}\mathbf{S}_{(t+1)}^{-1} = -\frac{1}{2}\mathbf{S}_{(t+1)}^{-1} + \ell\left(-\frac{1}{2}\mathbf{\Lambda} + \frac{1}{2}\mathbf{S}_{(t)}^{-1}\right)$$

$$\theta_{1(t+1)} = \mathbf{S}_{(t+1)}^{-1}\mathbf{m}_{(t+1)} = \mathbf{S}_{(t)}^{-1}\mathbf{m}_{(t)} + \ell\left(\beta\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}\mathbf{y} - \mathbf{S}_{(t)}^{-1}\mathbf{m}_{(t)}\right)$$

To perform stochastic variational inference with latentvariables, we require a factorisation as in bound 3.

$$\log p(\mathbf{y}) = \log \int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int q(\mathbf{X})\left\{\mathcal{L}_3 + \log p(\mathbf{X}) - \log q(\mathbf{X})\right\}\mathrm{d}\mathbf{X}$$

Considering $q\left(\mathbf{x}_i\right)$ as i.i.d. Expectations of $\mathcal{L}_3$ are tractable for certian covariance function.

To perform SVI in this model, we now alternate be- tween selecting a minibatch of data, and optimisting the relevant variables of $q(\mathbf{X})$ with $q(\mathbf{u})$ fixed, and up- dating $q(\mathbf{u})$ using the approximate natural gradient.

We can use data in stochastic way or in mini-batches for updates.

4 Methods used

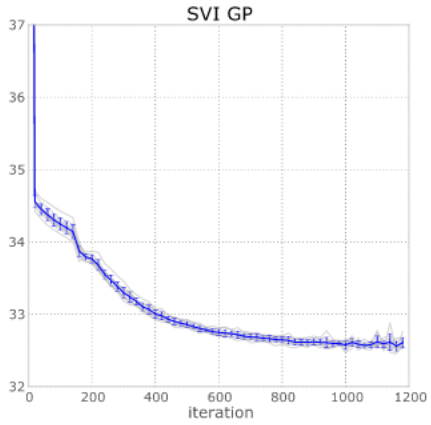
Reducing the dimension of data points can also help us to use only useful components of data points and thus reduce a lot of computations and saves time. I tried SVI code to reproduce results obtained in [2]. I also tried to use SVI for GP using GPflow librabry on toy dataset. I used SVI for GP on the airline delay dataset[2] also by first reducing the dimension by PCA.

5 Libraries used

- GPflow: Gaussian process library that uses TensorFlow for its core computations and Python for its front end. It has implementation for both full and variational sparse covariance prior with Gaussian and non Gaussian likelihood. (SVI and MCMC)
- Gpytorch: Implementation of GPs with GPU acceleration and Pytorch framework.

6 Results

Root mean squared errors in predicting flight delays using information about the flight.



I haven't got good results on PCA reduced dataset(SVI for GP). RMSE obtained after 400 iterations is around 48.2. As Automatic relevance determination parameters for the features used for predicting flight delays were used in [2], so loss in case of PCA reduced SVI method for GP would be much more than the loss shown in above plot.

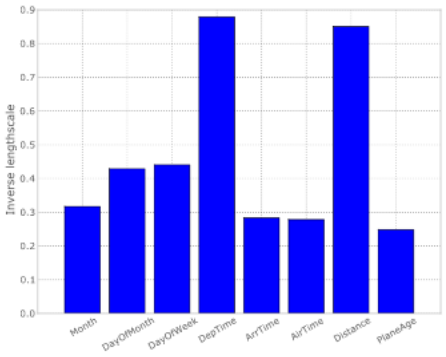


Figure 1: Automatic relevance determination param-eters for the features used for predicting flight delays (figure from [2])

7 Things I learned

I learned the various methods to approximate Gaussian Process to Sparse GP to handle big data. A lot of work has been done in this field. Going through each paper and thinking of another method and searching for the work in that field leads me to another papers. I have learned to use GP for classification and regression in case of any dataset, although pre processing time would be a constraint. I will use the things all the recent advancements to futhur improve accurcay and beat the accuracies and time of previous paper.

8 Possible future work

1. Most of the above methods uses k-means clustering of data for initialization of initial inducing points. And they fix the size of inducing points(k/no. of clusters) in prior.
2. Non parametric Bayesian modeling for number of clusters might improve accuracy.
3. Furthur, dimensionality reduction of data points can be made in such a way that reduced data points gives more intutuive meaning to GP. The idea would be similar to Amortized learning for dimensioanlity reduction for intial data points to learn parameters with online version of SVI updates. And Aromatized model for dimensionality reduction can be used then for left data points and future data points.

References

[1] Amir Dezfouli and Edwin V Bonilla. Scalable inference for gaussian process models with black-box likelihoods. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1414–1422. Curran Associates, Inc., 2015.

[2] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intellegence*, pages 282–290. auai.org, 2013.

[3] Joaquin Quiñonero Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res.*, 6:1939–1959, December 2005.

[4] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT Press, 2006.

[5] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In David van Dyk and Max Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR.