

---

# Improving the Gaussian Process Sparse Spectrum Approximation by Representing Uncertainty in Frequency Inputs

---

Yarin Gal  
Richard Turner  
University of Cambridge

YG279@CAM.AC.UK  
RET26@CAM.AC.UK

## Abstract

Standard sparse pseudo-input approximations to the Gaussian process (GP) cannot handle complex functions well. Sparse spectrum alternatives attempt to answer this but are known to over-fit. We suggest the use of variational inference for the sparse spectrum approximation to avoid both issues. We model the covariance function with a finite Fourier series approximation and treat it as a random variable. The random covariance function has a posterior, on which a variational distribution is placed. The variational distribution transforms the random covariance function to fit the data. We study the properties of our approximate inference, compare it to alternative ones, and extend it to the distributed and stochastic domains. Our approximation captures complex functions better than standard approaches and avoids over-fitting.

## 1 Introduction

The Gaussian process (GP, [Rasmussen & Williams, 2006](#)) is a powerful tool for modelling distributions over non-linear functions. It offers robustness to over-fitting, a principled way to tune hyper-parameters, and uncertainty bounds over the outputs. These properties are critical for tasks including non-linear function regression, reinforcement learning, density estimation, and more ([Brochu et al., 2010](#); [Rasmussen et al., 2003](#); [Engel et al., 2005](#); [Titsias & Lawrence, 2010](#)). But the advantages of the Gaussian process come with a great computational cost. Evaluating the GP posterior involves a large matrix inversion – for  $N$  data points the model requires  $O(N^3)$  time complexity.

Many approximations to the GP have been proposed to reduce the model’s time complexity. [Quiñonero-Candela & Rasmussen \(2005\)](#) survey approaches relying on “*sparse pseudo-input*” approximations. In these, a small number of points in the input space with corresponding outputs (“inducing inputs and outputs”) are used to define a new Gaus-

sian process. The new GP is desired to be as close as possible to the GP defined on the entire dataset, and the matrix inversion is now done with respect to the inducing points alone. These approaches are suitable for locally complex functions. The approximate model would place most of the inducing points in regions where the function is complex, and only a small number of points would be placed in regions where the function is not. Highly complex functions cannot be modelled well with this approach.

[Lázaro-Gredilla et al. \(2010\)](#) suggested an alternative approximation to the GP model. In their paper they suggest the decomposition of the GP’s stationary covariance function into its Fourier series. The infinite series is then approximated with a finite one. They optimise over the frequencies of the series to minimise some divergence from the full Gaussian process. This approach was named a “*sparse spectrum*” approximation. This approach is closely related to the one suggested by [Rahimi & Recht \(2007\)](#) in the randomised methods community (random projections). In [Rahimi & Recht \(2007\)](#)’s approach, the frequencies are randomised (sampled from some distribution rather than optimised) and the Fourier coefficients are computed analytically. Both approaches capture globally complex behaviour, but the direct optimisation of the different quantities often leads to some form of over-fitting (as reported in [Wilson et al., 2014](#) for the SSGP and shown below for random projections). Similar over-fitting problems observed with the *sparse pseudo-input* approximation were answered with variational inference ([Titsias, 2009](#)).

We suggest the use of variational inference for the *sparse spectrum* approximation. This allows us to avoid over-fitting while efficiently capturing globally complex behaviour. We replace the stationary covariance function with a finite approximation obtained from Monte Carlo integration. This finite approximation is a random variable, and conditioned on a dataset this random variable has an intractable posterior. We approximate this posterior with variational inference, resulting in a non-stationary finite rank covariance function. The approximating variational distribution transforms the covariance function to fit the data well. The prior from the GP model keeps the approximating distribution from over-fitting to the data.

Like in (Lázaro-Gredilla et al., 2010), we can marginalise over the Fourier coefficients. This results in approximate inference with  $\mathcal{O}(NK^2 + K^3)$  time complexity with  $N$  data points and  $K$  inducing frequencies (components in the Fourier expansion). This is the same as that of the sparse pseudo-input and sparse spectrum approximations. We can further optimise a variational distribution over the frequencies reducing the time complexity to  $\mathcal{O}(NK^2)$ . This factorises the lower bound and allows us to perform distributed inference, resulting in  $\mathcal{O}(K)$  time complexity given a sufficient number of nodes in a distributed framework. We can approximate the latter lower bound and use random subsets of the data (mini batches) employing stochastic variational inference (Hoffman et al., 2013). This results in  $\mathcal{O}(SK^2)$  time complexity with  $S \ll N$  the size of the mini-batch<sup>1</sup>.

In the experiments section we demonstrate the properties of our GP approximation and compare it to alternative approximations. We describe qualitative properties of the approximation and discuss how the approximation can be used to learn the covariance function by fitting to the data. We compare the approximation to the full Gaussian process, sparse spectrum GP, sparse pseudo-input GP, and random projections. We show that alternative approximations either over-fit or under-fit even on simple datasets. We empirically demonstrate the advantages of the variational inference in avoiding over-fitting by comparing the approximation to the sparse spectrum one on audio data from the TIMIT dataset. We compare the stochastic optimisation to the non-stochastic one, and compare the performance to the sparse pseudo-input SVI. Finally, we inspect the model’s time accuracy trade-off and show that it avoids over-fitting as the number of parameters increases.

## 2 Sparse Spectrum Approximation in Gaussian Process Regression

We use Bochner’s theorem (Bochner, 1959) to reformulate the covariance function in terms of its frequencies. Since our covariance function  $\mathbf{K}(\mathbf{x}, \mathbf{y})$  is stationary, it can be represented as  $\mathbf{K}(\mathbf{x} - \mathbf{y})$  for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^Q$ . Following Bochner’s theorem,  $\mathbf{K}(\mathbf{x} - \mathbf{y})$  can be represented as the Fourier transform of some finite measure  $\sigma^2 p(\mathbf{w})$  with  $p(\mathbf{w})$  a probability density,

$$\begin{aligned} \mathbf{K}(\mathbf{x} - \mathbf{y}) &= \int_{\mathbb{R}^Q} \sigma^2 p(\mathbf{w}) e^{-2\pi i \mathbf{w}^T (\mathbf{x} - \mathbf{y})} d\mathbf{w} \\ &= \int_{\mathbb{R}^Q} \sigma^2 p(\mathbf{w}) \cos(2\pi \mathbf{w}^T (\mathbf{x} - \mathbf{y})) d\mathbf{w} \end{aligned} \quad (1)$$

since the covariance function is real-valued.

This can be approximated as a finite sum with  $K$  terms

<sup>1</sup>Python code for all inference algorithms is available at <http://github.com/yaringal/VSSGP>

using Monte Carlo integration,

$$\mathbf{K}(\mathbf{x} - \mathbf{y}) \approx \frac{\sigma^2}{K} \sum_{k=1}^K \cos(2\pi \mathbf{w}_k^T ((\mathbf{x} - \mathbf{z}_k) - (\mathbf{y} - \mathbf{z}_k)))$$

with  $\mathbf{w}_k \sim p(\mathbf{w})$  and  $\mathbf{z}_k$  some  $Q$  dimensional vectors for  $k = 1, \dots, K$ . The points  $\mathbf{z}_k$  act as inducing inputs, and will have corresponding inducing frequencies in our approximation. For the sparse spectrum GP, these points take value 0. These will be explained in detail in a later section.

Using identity 1 proved in the appendix we rewrite the terms above for every  $k$  as

$$\begin{aligned} &\cos(2\pi \mathbf{w}_k^T ((\mathbf{x} - \mathbf{z}_k) - (\mathbf{y} - \mathbf{z}_k))) \\ &= \int_0^{2\pi} \frac{1}{2\pi} \sqrt{2} \cos(2\pi \mathbf{w}_k^T (\mathbf{x} - \mathbf{z}_k) + b) \\ &\quad \cdot \sqrt{2} \cos(2\pi \mathbf{w}_k^T (\mathbf{y} - \mathbf{z}_k) + b) db. \end{aligned}$$

This integral can again be approximated as a finite sum using Monte Carlo integration. To keep the notation simple, we approximate the integral with a single sample<sup>2</sup> for every  $k$ ,

$$\begin{aligned} \mathbf{K}(\mathbf{x} - \mathbf{y}) &\approx \frac{\sigma^2}{K} \sum_{k=1}^K \sqrt{2} \cos(2\pi \mathbf{w}_k^T (\mathbf{x} - \mathbf{z}_k) + b_k) \\ &\quad \cdot \sqrt{2} \cos(2\pi \mathbf{w}_k^T (\mathbf{y} - \mathbf{z}_k) + b_k) \\ &=: \widehat{\mathbf{K}}(\mathbf{x} - \mathbf{y}) \end{aligned}$$

with  $b_k \sim \text{Unif}[0, 2\pi]$ , defining the approximate covariance function  $\widehat{\mathbf{K}}$ . We refer to  $(\mathbf{w}_k)_{k=1}^K$  as inducing frequencies and to  $(b_k)_{k=1}^K$  as phases, and denote  $\boldsymbol{\omega} = (\mathbf{w}_k, b_k)_{k=1}^K$ . Note that this integral could be approximated with any arbitrary number of samples instead.

We denote  $\mathbf{X} \in \mathbb{R}^{N \times Q}$  the inputs and  $\mathbf{Y} \in \mathbb{R}^{N \times D}$  the outputs of a real-valued dataset with  $N$  data points. In Gaussian process regression we find the probability  $P(\mathbf{Y}|\mathbf{X})$  with the assumption that the function generating  $\mathbf{Y}$  is drawn from a Gaussian process. The full GP model is defined as (assuming stationary covariance function  $\mathbf{K}(\cdot, \cdot)$ ):

$$\mathbf{F} | \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}(\mathbf{X}, \mathbf{X}))$$

$$\mathbf{Y} | \mathbf{F} \sim \mathcal{N}(\mathbf{F}, \tau^{-1} \mathbf{I}_N)$$

with some precision hyper-parameter  $\tau$ .

Using  $\widehat{\mathbf{K}}$  instead as the covariance function of the Gaussian process yields the following generative model:

$$\mathbf{w}_k \sim p(\mathbf{w}), \quad b_k \sim \text{Unif}[0, 2\pi],$$

$$\boldsymbol{\omega} = (\mathbf{w}_k, b_k)_{k=1}^K$$

<sup>2</sup>The above transformation and approximate integration are used in the *randomised methods* literature (“Random projections”, Rahimi & Recht, 2007). It was shown to give better approximation than Monte Carlo integration of eq. 1. Intuitively it is equivalent to a random phase shift for each basis function.

$$\begin{aligned}\widehat{\mathbf{K}}(\mathbf{x}, \mathbf{y}) &= \frac{\sigma^2}{K} \sum_{k=1}^K \sqrt{2} \cos(2\pi \mathbf{w}_k^T (\mathbf{x} - \mathbf{z}_k) + b_k) \\ &\quad \cdot \sqrt{2} \cos(2\pi \mathbf{w}_k^T (\mathbf{y} - \mathbf{z}_k) + b_k) \\ \mathbf{F} | \mathbf{X}, \boldsymbol{\omega} &\sim \mathcal{N}(\mathbf{0}, \widehat{\mathbf{K}}(\mathbf{X}, \mathbf{X})) \\ \mathbf{Y} | \mathbf{F} &\sim \mathcal{N}(\mathbf{F}, \tau^{-1} \mathbf{I}_N).\end{aligned}$$

### 3 Random Covariance Functions

$\mathbf{K}$  is a deterministic covariance function of its inputs;  $\widehat{\mathbf{K}}$  is a random finite rank covariance function. As such, we can find the conditional distribution of the covariance function given a dataset (more precisely, the conditional distribution of  $\boldsymbol{\omega}$ ). This is a powerful view of this approximation – it allows us to transform the covariance function to fit the data well, while the prior keeps it from over-fitting to the data. We will use  $\widehat{\mathbf{K}}$  as our Gaussian process covariance function from now on, replacing  $\mathbf{K}$ . This results in the following predictive distribution:

$$p(\mathbf{Y} | \mathbf{X}) = \int p(\mathbf{Y} | \mathbf{F}) p(\mathbf{F} | \boldsymbol{\omega}, \mathbf{X}) p(\boldsymbol{\omega}) d\boldsymbol{\omega} d\mathbf{F}.$$

We can integrate this analytically for  $\mathbf{F}$  and obtain

$$p(\mathbf{Y} | \mathbf{X}) = \int \mathcal{N}(\mathbf{Y}; \mathbf{0}, \widehat{\mathbf{K}}(\mathbf{X}, \mathbf{X}) + \tau^{-1} \mathbf{I}_N) p(\boldsymbol{\omega}) d\boldsymbol{\omega}$$

but this involves the inversion of  $\widehat{\mathbf{K}}(\mathbf{X}, \mathbf{X}) + \tau^{-1} \mathbf{I}_N$ , which does not allow us to integrate over  $\boldsymbol{\omega}$  (even variationally!). Instead, we introduce an auxiliary random variable.

Denoting the  $1 \times K$  row vector

$$\phi(\mathbf{x}, \boldsymbol{\omega}) = \left[ \sqrt{\frac{2\sigma^2}{K}} \cos(2\pi \mathbf{w}_k^T (\mathbf{x} - \mathbf{z}_k) + b_k) \right]_{k=1}^K$$

and the  $N \times K$  feature matrix

$$\Phi = [\phi(\mathbf{x}_n, \boldsymbol{\omega})]_{n=1}^N,$$

we have  $\widehat{\mathbf{K}}(\mathbf{X}, \mathbf{X}) = \Phi \Phi^T$ . We rewrite  $p(\mathbf{Y} | \mathbf{X})$  as

$$p(\mathbf{Y} | \mathbf{X}) = \int \mathcal{N}(\mathbf{Y}; \mathbf{0}, \Phi \Phi^T + \tau^{-1} \mathbf{I}_N) p(\boldsymbol{\omega}) d\boldsymbol{\omega}.$$

Following an identity proved in (Bishop, 2006, page 93) we introduce a  $K \times 1$  auxiliary random variable  $\mathbf{a}_d \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$  to the distribution inside the integral above,

$$\begin{aligned}\mathcal{N}(\mathbf{y}_d; \mathbf{0}, \Phi \Phi^T + \tau^{-1} \mathbf{I}_N) \\ = \int \mathcal{N}(\mathbf{y}_d; \Phi \mathbf{a}_d, \tau^{-1} \mathbf{I}_N) \mathcal{N}(\mathbf{a}_d; \mathbf{0}, \mathbf{I}_K) d\mathbf{a}_d,\end{aligned}$$

where  $\mathbf{y}_d$  is the  $d$ 'th column of the  $N \times D$  matrix  $\mathbf{Y}$ .

Writing  $\mathbf{A} = [\mathbf{a}_d]_{d=1}^D$ , the above is equivalent to<sup>3</sup>

$$p(\mathbf{Y} | \mathbf{X}) = \int p(\mathbf{Y} | \mathbf{A}, \mathbf{X}, \boldsymbol{\omega}) p(\mathbf{A}) p(\boldsymbol{\omega}) d\mathbf{A} d\boldsymbol{\omega}. \quad (2)$$

<sup>3</sup>This is equivalent to the weighted basis function interpretation of the Gaussian process (Rasmussen & Williams, 2006).

We refer to  $\mathbf{A} \in \mathbb{R}^{K \times D}$  as the Fourier coefficients.

Regarding  $\boldsymbol{\omega}$  as parameters and optimising these values (integrating over  $\mathbf{A}$ ) results in the *sparse spectrum* approximation (Lázaro-Gredilla et al., 2010). Regarding  $\mathbf{A}$  as parameters and optimising these values (leaving  $\boldsymbol{\omega}$  constant) results in a method known as ‘‘random projections’’ (Rahimi & Recht, 2007). Related work to random projections variationally integrates over the hyper-parameters while leaving  $\boldsymbol{\omega}$  constant (Tan et al., 2013).

We can extend the above to sums of covariance functions as well. Following proposition 1 in the appendix, given a sum of covariance functions with  $L$  components (with each corresponding to  $\Phi_i$  an  $N \times K$  matrix) we have  $\Phi = [\Phi_i]_{i=1}^L$  an  $N \times LK$  matrix. We will continue the development of our method using weighted sums of automatic relevance determination (ARD) squared exponential (SE) covariance functions. Note however that our method is general and can be extended to other covariance functions as well. A sum of SE covariance functions with  $L$  components is given by

$$\mathbf{K}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^L \sigma_i^2 \exp\left(-\frac{1}{2} \sum_{q=1}^Q \frac{(x_q - y_q)^2}{l_{iq}^2}\right)$$

with weights  $\sigma_i^2$  and length-scales  $l_{iq}$ . We write  $\mathbf{L}_i = \text{diag}([2\pi l_{qi}]_{q=1}^Q)$ .

For  $p(\mathbf{w})$  composed of a single SE component, we follow proposition 2 in the appendix and perform a change of variables, resulting in  $p(\mathbf{w})$  a standard normal distribution with the parameters of  $p(\mathbf{w})$  now expressed in  $\Phi$  instead. For  $p(\mathbf{w})$  composed of several components, for each component  $i$  we get  $\Phi_i$  is an  $N \times K$  matrix with elements

$$\sqrt{\frac{2\sigma_i^2}{K}} \cos(2\pi (\mathbf{L}_i^{-1} \mathbf{w}_k)^T (\mathbf{x} - \mathbf{z}_k) + b_k),$$

where for simplicity, we index  $\mathbf{w}_k$  and  $b_k$  with  $k = 1, \dots, LK$  as a function of the component  $i$ .

### 4 Variational Inference

The predictive distribution for an input point  $\mathbf{x}^*$  is given by

$$p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{A}, \boldsymbol{\omega}) p(\mathbf{A}, \boldsymbol{\omega} | \mathbf{X}, \mathbf{Y}) d\mathbf{A} d\boldsymbol{\omega}, \quad (3)$$

with  $\mathbf{y}^* \in \mathbb{R}^{1 \times D}$ . The distribution  $p(\mathbf{A}, \boldsymbol{\omega} | \mathbf{X}, \mathbf{Y})$  cannot be evaluated analytically. Instead we define an approximating *variational* distribution  $q(\mathbf{A}, \boldsymbol{\omega})$ , whose structure is easy to evaluate.

We would like our approximating distribution to be as close as possible to the posterior distribution obtained from the full GP. We thus minimise the Kullback–Leibler divergence

$$\text{KL}(q(\mathbf{A}, \boldsymbol{\omega}) | p(\mathbf{A}, \boldsymbol{\omega} | \mathbf{X}, \mathbf{Y})),$$

resulting in the approximate predictive distribution

$$q(\mathbf{y}^*|\mathbf{x}^*) = \int p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{A}, \boldsymbol{\omega})q(\mathbf{A}, \boldsymbol{\omega})d\mathbf{A}d\boldsymbol{\omega}. \quad (4)$$

Minimising the Kullback–Leibler divergence is equivalent to maximising the log evidence lower bound

$$\mathcal{L} := \int q(\mathbf{A}, \boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{A}, \mathbf{X}, \boldsymbol{\omega})d\mathbf{A}d\boldsymbol{\omega} - \text{KL}(q(\mathbf{A}, \boldsymbol{\omega})||p(\mathbf{A})p(\boldsymbol{\omega})) \quad (5)$$

with respect to the variational parameters defining  $q(\mathbf{A}, \boldsymbol{\omega})$ .

We define a factorised variational distribution  $q(\mathbf{A}, \boldsymbol{\omega}) = q(\mathbf{A})q(\boldsymbol{\omega})$ . We define  $q(\boldsymbol{\omega})$  with  $\boldsymbol{\omega} = (\mathbf{w}_k, b_k)_{k=1}^{LK}$  to be a joint Gaussian distribution and a uniform distribution,

$$\begin{aligned} \mathbf{w}_k &\sim \mathcal{N}(\mu_k, \Sigma_k), & k &= 1, \dots, LK \\ b_k &\sim \text{Unif}(\alpha_k, \beta_k), & k &= 1, \dots, LK \end{aligned}$$

with  $\Sigma_k$  diagonal,  $0 \leq \alpha_k \leq \beta_k \leq 2\pi$ , and define  $q(\mathbf{A}) = \prod_{d=1}^D q(\mathbf{a}_d)$  (with  $\mathbf{a}_d \in \mathbb{R}^{LK \times 1}$ ) by

$$\mathbf{a}_d \sim \mathcal{N}(\mathbf{m}_d, \mathbf{s}_d), \quad d = 1, \dots, D$$

with  $\mathbf{s}_d$  diagonal. We evaluate the log evidence lower bound and optimise over  $\{\mu_k, \Sigma_k, \alpha_k, \beta_k\}_{k=1}^{LK}$ ,  $\{\mathbf{m}_d, \mathbf{s}_d\}_{d=1}^D$ , and  $\{\sigma_i, \mathbf{L}_i\}_{i=1}^L$  to maximise Eq. 5.

#### 4.1 Evaluating the Log Evidence Lower Bound

Given  $\mathbf{A}$  and  $\boldsymbol{\omega}$ , we evaluate the probability of the  $d$ 'th column of  $\mathbf{Y}$ ,  $\mathbf{y}_d$ , as

$$\begin{aligned} \log p(\mathbf{y}_d|\mathbf{A}, \mathbf{X}, \boldsymbol{\omega}) &= \\ &= -\frac{N}{2} \log(2\pi\tau^{-1}) - \frac{\tau}{2} (\mathbf{y}_d - \Phi\mathbf{a}_d)^T (\mathbf{y}_d - \Phi\mathbf{a}_d). \end{aligned}$$

Note that  $\mathbf{y}_d$  is an  $N \times 1$  vector,  $\Phi$  is an  $N \times LK$  matrix, and  $\mathbf{a}_d$  is an  $LK \times 1$  vector.

We need to evaluate the expectations of  $\mathbf{y}_d^T \Phi \mathbf{a}_d$  and  $\mathbf{a}_d^T \Phi^T \Phi \mathbf{a}_d$  (both scalar values) under  $q(\mathbf{A})q(\boldsymbol{\omega})$ :

$$E_{q(\mathbf{A})q(\boldsymbol{\omega})}(\mathbf{y}_d^T \Phi \mathbf{a}_d) = \mathbf{y}_d^T E_{q(\boldsymbol{\omega})}(\Phi) E_{q(\mathbf{A})}(\mathbf{a}_d),$$

and

$$E_{q(\mathbf{A})q(\boldsymbol{\omega})}(\mathbf{a}_d^T \Phi^T \Phi \mathbf{a}_d) = \text{tr} \left( E_{q(\boldsymbol{\omega})}(\Phi^T \Phi) E_{q(\mathbf{A})}(\mathbf{a}_d \mathbf{a}_d^T) \right).$$

The values  $E_{q(\mathbf{A})}(\mathbf{a}_d)$  and  $E_{q(\mathbf{A})}(\mathbf{a}_d \mathbf{a}_d^T)$  are evaluated as

$$E_{q(\mathbf{A})}(\mathbf{a}_d) = \mathbf{m}_d, \quad E_{q(\mathbf{A})}(\mathbf{a}_d \mathbf{a}_d^T) = \mathbf{s}_d + \mathbf{m}_d \mathbf{m}_d^T.$$

Next we evaluate  $E_{q(\boldsymbol{\omega})}(\Phi)$ . Remember that  $\Phi$  depends on  $\boldsymbol{\omega}$  and that  $q(\boldsymbol{\omega}) = q((\mathbf{w}_k, b_k)_{k=1}^{LK})$ . We write as shorthand  $\bar{\mathbf{x}}_{nk} := 2\pi \mathbf{L}_i^{-1}(\mathbf{x}_n - \mathbf{z}_k)$  with component  $i$  appropriate to  $k$ . Following identity 2 proved in the appendix, we have that the expectation of a single cosine in  $\Phi$  with respect to  $q(\mathbf{w})$  is

$$E_{q(\mathbf{w})}(\cos(\mathbf{w}^T \bar{\mathbf{x}}_n + b)) = e^{-\frac{1}{2} \bar{\mathbf{x}}_n^T \Sigma \bar{\mathbf{x}}_n} \cos(\mu^T \bar{\mathbf{x}}_n + b)$$

where  $\mu$  is the mean of  $q(\mathbf{w})$  and  $\Sigma$  is its covariance (note

that we omitted the index  $k$ ). We get

$$\begin{aligned} \left( E_{q(\boldsymbol{\omega})}(\Phi) \right)_{n,k} &= \sqrt{\frac{2\sigma_i^2}{K}} e^{-\frac{1}{2} \bar{\mathbf{x}}_{nk}^T \Sigma_k \bar{\mathbf{x}}_{nk}} \\ &\quad \cdot E_{q(b_k)}(\cos(\mu_k^T \bar{\mathbf{x}}_{nk} + b_k)) \end{aligned}$$

with the integration with respect to  $q(b_k)$  trivial.

Next we evaluate  $E_{q(\boldsymbol{\omega})}(\Phi^T \Phi)$ , an  $LK \times LK$  matrix:

$$\begin{aligned} \left( E_{q(\boldsymbol{\omega})}(\Phi^T \Phi) \right)_{i,j} &= \sum_{n=1}^N \frac{2\sigma_i^2}{LK} E(\cos(\mathbf{w}_i^T \bar{\mathbf{x}}_{ni} + b_i) \\ &\quad \cdot \cos(\mathbf{w}_j^T \bar{\mathbf{x}}_{nj} + b_j)) \end{aligned}$$

where the expectation is over  $q(\mathbf{w}_i, b_i, \mathbf{w}_j, b_j)$  for  $i, j \leq LK$ .

For  $i \neq j$ , from independence we can break the expectation of each term into a product of  $E_{q(\mathbf{w}_i, b_i)}(\cos(\mathbf{w}_i^T \bar{\mathbf{x}}_{ni} + b_i))$  terms, and for  $i = j$  we follow identity 3 to obtain

$$\begin{aligned} E_{q(\mathbf{w}_i, b_i)}(\cos(\mathbf{w}_i^T \bar{\mathbf{x}}_{ni} + b_i)^2) &= \\ &= \frac{1}{2} + \frac{1}{2} e^{-2\bar{\mathbf{x}}_{ni}^T \Sigma_i \bar{\mathbf{x}}_{ni}} E_{q(b_i)}(\cos(2\mu_i^T \bar{\mathbf{x}}_{ni} + 2b_i)). \end{aligned}$$

In conclusion, we obtained our optimisation objective:

$$\begin{aligned} \mathcal{L} &= \sum_{d=1}^D \left( -\frac{N}{2} \log(2\pi\tau^{-1}) - \frac{\tau}{2} \mathbf{y}_d^T \mathbf{y}_d \right. \\ &\quad \left. + \tau \mathbf{y}_d^T E_{q(\boldsymbol{\omega})}(\Phi) \mathbf{m}_d \right. \\ &\quad \left. - \frac{\tau}{2} \text{tr}(E_{q(\boldsymbol{\omega})}(\Phi^T \Phi)(\mathbf{s}_d + \mathbf{m}_d \mathbf{m}_d^T)) \right) \\ &\quad - \text{KL}(q(\mathbf{A})||p(\mathbf{A})) - \text{KL}(q(\boldsymbol{\omega})||p(\boldsymbol{\omega})). \quad (6) \end{aligned}$$

The KL divergence terms can be evaluated analytically for the Gaussian and uniform distributions.

#### 4.2 Optimal variational distribution over $\mathbf{A}$

In the above we optimise over the variational parameters for  $\mathbf{A}$ , namely  $\mathbf{m}_d$  and  $\mathbf{s}_d$  for  $d \leq D$ . This allows us to attain a reduction in time complexity compared to previous approaches and use stochastic inference, as will be explained below. This comes with a cost, as the dependence between  $\boldsymbol{\omega}$  and  $\mathbf{A}$  can render the optimisation hard.

We can find the optimal variational distribution  $q(\mathbf{A})$  analytically, which allows us to optimise  $q(\boldsymbol{\omega})$  and the hyperparameters alone. In proposition 3 in the appendix we show that the optimal variational distribution is given by

$$q(\mathbf{a}_d) = \mathcal{N}(\Sigma E_{q(\boldsymbol{\omega})}(\Phi^T) \mathbf{y}_d, \tau^{-1} \Sigma)$$

with  $\Sigma = (E_{q(\boldsymbol{\omega})}(\Phi^T \Phi) + \tau^{-1} I)^{-1}$ .

The lower bound to optimise then reduces to

$$\mathcal{L} = \sum_{d=1}^D \left( -\frac{N}{2} \log(2\pi\tau^{-1}) - \frac{\tau}{2} \mathbf{y}_d^T \mathbf{y}_d + \frac{1}{2} \log(|\tau^{-1} \Sigma|) \right)$$

$$\begin{aligned}
 & + \frac{1}{2} \tau \mathbf{y}_d^T E_{q(\omega)}(\Phi) \Sigma E_{q(\omega)}(\Phi^T) \mathbf{y}_d \\
 & - \text{KL}(q(\omega) \| p(\omega)). \tag{7}
 \end{aligned}$$

## 5 Distributed Inference and Stochastic Inference

Evaluating  $\mathcal{L}$  in equation 6 requires  $\mathcal{O}(NK^2)$  time complexity (for fixed  $Q, D$ , diagonal  $\mathbf{s}_d$ , and covariance function with one component  $L = 1$ ). This stems from the term  $E_{q(\omega)}(\Phi^T \Phi) - a$   $K \times K$  matrix, where each element is composed of a sum over  $N$ .

Following the ideas of (Gal et al., 2014) we show that the approximation can be implemented in a distributed framework. Write

$$\begin{aligned}
 \mathcal{L}_{nd} = & -\frac{1}{2} \log(2\pi\tau^{-1}) - \frac{\tau}{2} y_{nd} y_{nd} + \tau y_{nd} E_{q(\omega)}(\phi_n) \mathbf{m}_d \\
 & - \frac{\tau}{2} \text{tr}(E_{q(\omega)}(\phi_n^T \phi_n) (\mathbf{s}_d + \mathbf{m}_d \mathbf{m}_d^T))
 \end{aligned}$$

with  $\phi_n = \phi(\mathbf{x}_n, \omega)$ . We can break the optimisation objective in equation 6 into a sum over  $N$ ,

$$\mathcal{L} = \sum_{d=1}^D \sum_{n=1}^N \mathcal{L}_{nd} - \text{KL}(q(\mathbf{A}) \| p(\mathbf{A})) - \text{KL}(q(\omega) \| p(\omega)). \tag{8}$$

These terms can be computed concurrently on  $T$  different nodes in a distributed framework, requiring  $\mathcal{O}(NK^2/T)$  time complexity in each iteration. Scaling the number of nodes with dataset size gives a practical time complexity of  $\mathcal{O}(K^2)$ . We can further break the computation of  $\mathcal{L}_{nd}$  into a sum over  $K$  as well, thus reducing the time complexity to  $\mathcal{O}(K)$  with  $K$  inducing points. This is in comparison to distributed inference with sparse pseudo-input GPs which takes  $\mathcal{O}(K^3)$  time complexity in practice with  $K$  inducing points, resulting from the covariance matrix inversion. This is a major advantage, as empirical results suggest that in many real-world applications the number of inducing points should scale with the data.

We can exploit the above representation and perform stochastic variational inference (SVI) by approximating the objective with a subset of the data, resulting in noisy gradients (Hoffman et al., 2013). Here we use as our objective

$$\begin{aligned}
 \mathcal{L} \approx & \frac{N}{|S|} \sum_{d=1}^D \sum_{n \in S} \mathcal{L}_{nd} \\
 & - \text{KL}(q(\mathbf{A}) \| p(\mathbf{A})) - \text{KL}(q(\omega) \| p(\omega)) \tag{9}
 \end{aligned}$$

with a mini-batch  $S$  of randomly selected points. This is an unbiased estimator to the lower bound. The time complexity of each iteration is  $\mathcal{O}(SK^2)$  with  $S \ll N$  the size of the random subset, compared to  $\mathcal{O}(SK^2 + K^3)$  of GP SVI using sparse pseudo-input approximation (Hensman et al., 2013).

## 6 Predictive Distribution

The approximate predictive distribution for a point  $\mathbf{x}^*$  is given by equation 4. Denoting  $\mathbf{M} = [\mathbf{m}_d]_{d=1}^D$ , we have

$$E_{q(\mathbf{y}^* | \mathbf{x}^*)}(\mathbf{y}^*) = E_{q(\omega)}(\phi_*) \mathbf{M} \tag{10}$$

following proposition 4 in the appendix.

The variance of the predictive distribution is given by

$$\text{Var}_{q(\mathbf{y}^* | \mathbf{x}^*)}(\mathbf{y}^*) = \tau^{-1} \mathbf{I}_D + \Psi \tag{11}$$

$$+ \mathbf{M}^T (E_{q(\omega)}(\phi_*^T \phi_*) - E_{q(\omega)}(\phi_*)^T E_{q(\omega)}(\phi_*)) \mathbf{M}$$

with  $\Psi_{i,j} = \text{tr}(E_{q(\omega)}(\phi_*^T \phi_*) \cdot \mathbf{s}_i) \cdot \mathbb{1}[i = j]$ , following proposition 5 in the appendix ( $\mathbb{1}$  is the indicator function).

When the optimal variational distribution over  $\mathbf{A}$  is used, we have  $\mathbf{M} = \Sigma E_{q(\omega)}(\Phi^T) \mathbf{Y}$  and  $\mathbf{s}_i = \tau^{-1} \Sigma$  for all  $i$ .

## 7 Properties of the Approximate Model

We have presented a variational sparse spectrum approximation to the Gaussian process (VSSGP in short). We gave 3 approximate models with different lower bounds: an approximate model with an optimal variational distribution over  $\mathbf{A}$  (equation 7, referred to as VSSGP), an approximate model with a *factorised* lower bound over the data points (equations 6, 8, referred to as factorised VSSGP – fVSSGP), and an approximation to the lower bound of the factorised VSSGP for use in stochastic optimisation over subsets of the data (equation 9, referred to as stochastic factorised VSSGP – sfVSSGP).

The VSSGP model generalises on some of the GP approximations brought in the introduction. Fixing  $\Sigma_k$  at zero in our approximate model (as well as  $\alpha_k$  and  $\beta_k$  at 0 and  $2\pi$ ) and optimising only  $\mu_k$  results in the *sparse spectrum approximation*. Randomising  $\mu_k$ , we obtain the *random projections approximation* (Rahimi & Recht, 2007). Indeed, for  $\Sigma_k = \mathbf{0}$  and fixed phases we have that  $E_{q(\omega)}(\Phi^T \Phi) = E_{q(\omega)}(\Phi^T) E_{q(\omega)}(\Phi)$  and  $E_{q(\omega)}(\Phi) = \Phi$ , and equation 7 recovers equation 8 in (Lázaro-Gredilla et al., 2010).

The points  $\mathbf{z}_k$  act as *inducing inputs* with  $\mathbf{w}_k$  and  $b_k$  acting as *inducing frequencies and phases* at these inputs. This is similar to the *sparse pseudo-input approximation*, but instead of having inducing values in the output space, we have the inducing values in the frequency domain. These are necessary for the approximation. Without these points (or equivalently, setting these to  $\mathbf{0}$ ), the features would decay quickly for data points far from the origin (the fixed point  $\mathbf{0}$ ).

The distribution over the frequencies is optimised to fit the data well. The prior is used to regulate the fit and avoid over-fitting to the data. This approximation can be used to learn covariance functions by fitting them to the data. This is similar to the ideas brought in (Duvenaud et al., 2013) where the structure of a covariance function is sought by

looking at possible compositions of these. This can give additional insight into the data. In (Duvenaud et al., 2013) the structure of the covariance composition is used to explain the data. In the approximation presented here the spectrum of the covariance function can be used to explain the data.

It is interesting to note that although the approximated covariance function  $\mathbf{K}(\mathbf{x}, \mathbf{y})$  has to be stationary (i.e. it is represented as  $\mathbf{K}(\mathbf{x}, \mathbf{y}) = \mathbf{K}(\mathbf{x} - \mathbf{y})$ ), the approximate posterior is not. This is in contrast to the SSGP that results in a stationary approximation. Furthermore, unlike the SSGP, our approximation is not periodic. This is one of the theoretical limitations of the sparse spectrum approximation. The limitation arises from the fact that the covariance is represented as a weighted sum of cosines in the SSGP. In the our approximation this is avoided by decaying the cosines to zero. This and other properties of the approximation are discussed further in discussion 1 in the appendix.

## 8 Experiments

We study the properties of VSSGP and compare it to alternative approximations, showing its advantages. We compare the VSSGP to the full Gaussian process (denoted Full GP), sparse spectrum GP approximation (denoted SSGP), sparse pseudo-input GP approximation (denoted SPGP), and the random projections approximation (denoted RP). We compare the VSSGP to the fVSSGP and sfVSSGP that offer improved time complexity. We further compare sfVSSGP to the existing sparse pseudo-input GP approach used with SVI (denoted sSPGP, Hensman et al., 2013). We inspect the model’s time accuracy trade-off and show that it avoids over-fitting as the number of parameters increases.

### 8.1 VSSGP Properties

We evaluate the predictive mean and uncertainty of the VSSGP on the atmospheric CO<sub>2</sub> concentrations dataset derived from in situ air samples collected at Mauna Loa Observatory, Hawaii (Keeling et al., 2004). We fit the approximate model using a sum of squared exponential covariance functions with two components initialised with initial length-scales [0.1, 1000]. We randomised the phases following the Monte Carlo integration (instead of optimising a variational distribution on these<sup>4</sup>) and initialise the frequencies at random. We use 10 inducing inputs for each component ( $K = 10$ ), set the observation noise precision to  $\tau = 10$ , and covariance noise to  $\sigma^2 = 1$ . LBFGS (Zhu et al., 1997) was used to optimise the objective given in equation 7, and was run for 500 iterations.

Figure 1 shows the predictive mean with the predictive uncertainty increasing far from the data. This is a property shared with the SE GP. The covariance hyper-parameters optimise to length-scales [0.52, 43.83] and covariance noise

<sup>4</sup>This seems to work better in practice.

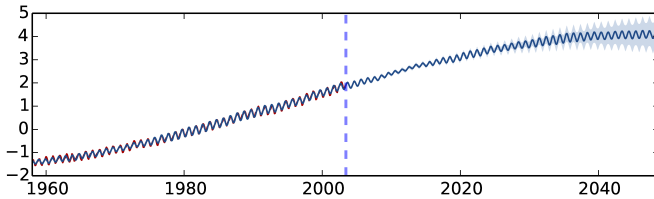


Figure 1. Predictive mean and uncertainty on the Mauna Loa CO<sub>2</sub> concentrations dataset. In red is the observed function; in blue is the predictive mean plus/minus two standard deviations. In this example the approximating distribution is used with a sum of squared exponential covariance functions with two components ( $L = 2, K = 10$ ).

[0.37, 5.18], correspondingly. The frequency with the smallest standard-deviation (highest confidence) for the first component is 3.27 (corresponding to a period of 1 year, capturing the short term behaviour). For the second component it is 0.58 (corresponding to a period of 475 years capturing the long term behaviour).

### 8.2 Comparison to Existing GP Approximations

We compare various GP approximations on the solar irradiance dataset (Lean, 2004). We scaled the dataset dividing by the data standard deviation, and removed 5 segments of length 20. We followed the experiment set-up of the previous section and used the same initial parameters for all approximate models. We use a single SE setting its length-scale  $l = 1$ , and used 50 inducing inputs. LBFGS was used for 1000 iterations. The RP model was run twice with two different settings: once following the same set-up of the other models, optimising over the model hyper-parameters (RP<sub>1</sub>), and once keeping all hyper-parameters fixed and setting the observation noise precision to  $\tau = 100$  with  $K = 500$  inducing inputs<sup>5</sup> (RP<sub>2</sub>).

Figure 2 shows qualitatively the predictive mean and uncertainty of the various approaches. SSGP and RP seem to over-fit the function using high frequencies with high confidence. SPGP seems to under-fit the function, but has accurate predictive mean and uncertainty at points where many inducing inputs lie (such as the flat region). VSSGP’s predictive mean resembles that of the full GP, but with increased uncertainty throughout the space. Further, its uncertainty on the missing segments is smaller than that of the full GP (some frequencies have low uncertainty, thus used near the data). The full GP learnt length-scale is 4. VSSGP learnt a length-scale of 3, and SPGP learnt a length-scale of 5. SSGP and RP<sub>1</sub> learnt length-scales of 0.97, 1.66, i.e. the hyper-parameter optimisation found a local minimum.

Table 1 gives a quantitative comparison of the different approximations for the task of imputation. RMSE (root mean square error) of the approximate predictive mean on the

<sup>5</sup>This follows the usual use of the model in the randomised methods community. We experimented with various values of  $\tau$  and decided to use 100.

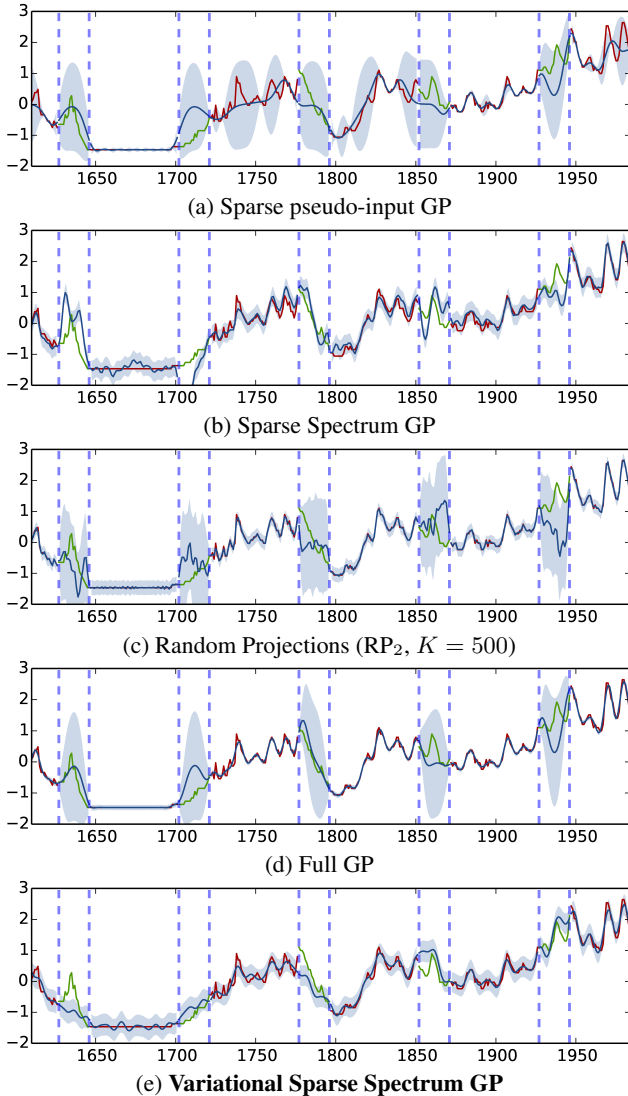


Figure 2. Predictive mean and uncertainty on the reconstructed solar irradiance dataset with missing segments, for the GP and various GP approximations. In red is the observed function and in green are the missing segments. In blue is the predictive mean plus/minus two standard deviations of the various approximations. All tests were done with the SE covariance function, and all sparse approximations use  $K = 50$  inducing inputs (apart from  $RP_2$  with  $K = 500$ ).

Solar	SPGP	SSGP	$RP_1$	$RP_2$	GP	<b>VSSGP</b>
Train	0.23	0.15	0.32	0.04	0.08	<b>0.13</b>
Test	0.61	0.63	0.65	0.76	0.50	<b>0.41</b>

Table 1. Imputation RMSE on both train and test sets, for the reconstructed solar irradiance dataset. All tests were done with the SE covariance function, and all sparse approximations use 50 inducing inputs (apart from  $RP_2$  that uses  $K = 500$ ).

missing segments was computed (test error), as well as the RMSE on the observed function (training error). Note that the full GP seems to get worse results than VSSGP. This might be because the learnt covariance function captures

the data better. It is interesting to note that the average standard deviation on the test set for the full GP is 0.63, whereas for the VSSGP it is 0.25, i.e. the model is more certain about its predictions.

### 8.3 From SSGP to Variational SSGP

We use variational inference in the VSSGP to avoid overfitting to the data, a behaviour that is often observed with SSGP. To test this we perform a direct comparison of the proposed approximate model to SSGP on the task of audio signal imputation. For this experiment we used a short speech signal with 1000 samples taken from the TIMIT dataset (Garofolo et al., 1993). We removed 5 segments of length 40 from the signal, and evaluated the imputation error (RMSE) of the predictive mean with  $K = 100$  inducing points. We used the same experiment set-up as before with a sum of 2 SE covariance functions with length-scales  $l = [2, 10]$  and observation noise precision  $\tau = 1000$  matching the signal magnitude. LBFGS was run for 1000 iterations. The experiment was repeated 5 times and the results averaged.

Audio 1K	SSGP	<b>VSSGP</b>
Train	0.0091 $\pm$ 0.0042	<b>0.0062 <math>\pm</math> 0.00048</b>
Test	0.088 $\pm$ 0.033	<b>0.034 <math>\pm</math> 0.0043</b>

Table 3. Imputation RMSE on both train and test sets, for a speech signal segment of length 1K ( $K = 100$ ).

Table 3 shows the RMSE of the training set and test set for the audio data. SSGP seems to achieve a small training error but cannot generalise well to unseen audio segments. VSSGP attains a slightly lower training error, and is able to impute unseen audio segments with better accuracy.

It is interesting to note that using the RMSE of the short-time Fourier transform of the original signal and the predicted mean (STFT, the common metric for audio imputation, with 25ms frame size and a hop size of 12ms), the SSGP model attains a training error of  $0.094 \pm 0.05$  and a test error of  $0.55 \pm 0.41$ . The VSSGP attains a training error of  $0.067 \pm 0.0067$  with a test error of  $0.17 \pm 0.022$ . For comparison, baseline performance of predicting 0 attains a training error of 0.44 and an error of 0.38 on the test set. GPs for audio data are examined further in (Turner, 2010).

### 8.4 VSSGP, factorised VSSGP, and stochastic factorised VSSGP

VSSGP, fVSSGP, and sfVSSGP all rely on different lower bounds to the same approximate model. Whereas VSSGP solves for the variational distribution over the Fourier coefficients analytically, fVSSGP optimises over these quantities. This reduces the time complexity, but with the price of potentially worsened performance. sfVSSGP further employs an approximation to the lower bound using random subsets of the data – following the idea that not all data

Audio 1K	VSSGP	fVSSGP	sfVSSGP
Train	<b>0.0062 ± 0.00048</b> (0.063±0.0068)	<b>0.0054 ± 0.00083</b> (0.055±0.0088)	<b>0.005 ± 0.003</b> (0.052±0.031)
Test	<b>0.034 ± 0.0043</b> (0.17±0.022)	<b>0.038 ± 0.0049</b> (0.22±0.028)	<b>0.04 ± 0.0066</b> (0.24±0.0089)

Table 2. Imputation RMSE (and in smaller font STFT RMSE) on train and test sets, for a speech signal segment of length 1K ( $K = 100$ ).

points have to be observed for a good fit to be found. This assumption has the potential to hinder performance even further. We next assess these trade-offs.

We repeated the experimental set-up of the previous section (and use the same RMSE for VSSGP). We optimise both fVSSGP and sfVSSGP for 5000 iterations instead of the 1000 of VSSGP. This is because the improved time complexity allows us to perform more function evaluations within the same time-frame. We optimise the fVSSGP lower bound with LBFSGS, and the sfVSSGP lower bound with RMSPROP (Tieleman & Hinton, 2012). RMSPROP performs stochastic optimisation with no need for learning-rate tuning – the learning rate changes adaptively based on the directions of the last two gradients.

Table 2 shows the RMSE for the train and test sets. Both fVSSGP and sfVSSGP effectively achieve the same test set accuracy (taking the standard deviation into account). We also see a slight decrease in train set RMSE.

### 8.5 Stochastic Variational Inference

We compared sfVSSGP to the SPGP approximation with stochastic variational inference (sSPGP, Hensman et al., 2013). We used the same audio experiment as above, but with a signal of length 16000. 25 random segments of length 80 were removed from the signal. sSPGP’s time complexity ( $\mathcal{O}(SK^2 + K^3)$  with mini-batch of size  $S$  and  $K$  inducing points) prohibits it from being used with a large number of inducing points. We therefore used 800 inducing points for sSPGP and 400 inducing inputs for each component in the covariance function of sfVSSGP ( $K = 400$ ).

The RMSE of sSPGP for the training set is 0.043 and for the test set is **0.034** (with a training time of 133 minutes using GPy (authors, 2012–2014)). The RMSE of sfVSSGP for the training set is 0.016 and for the test set is **0.034** (with a training time of 48 minutes). Using the same audio imputation metric as in the previous section, the STFT RMSE for sSPGP on the training set is 0.54 and on the test

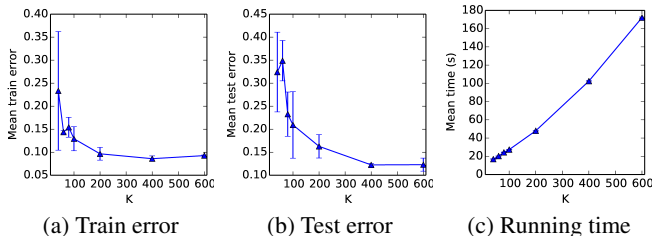


Figure 3. Mean and standard deviation for train error, test error, and running time, all as functions of the number of inducing points ( $K$ ) for a speech signal segment of length 4K.

set is **0.43**. The STFT RMSE for VSSGP on the training set is 0.18 with a test error **0.3**. For comparison, baseline performance of predicting 0 attains an error of 0.52 on the training set and an error of **0.62** on the test set.

### 8.6 Speed-Accuracy Trade-off

We inspect the speed-accuracy trade-off of the approximation (RMSE as a function of the number of inducing points) for the sfVSSGP approximation. We repeat the same audio experiment set-up with a speech signal with 4000 samples and evaluate the imputation error (RMSE) of the predictive mean with various numbers of inducing point. RMSPROP was run for 500 iterations. The experiment was repeated 5 times and the results averaged.

Figure 3 shows that the approximation offers increased accuracy with an increasing number of inducing points. No further improvement is achieved with more than 400 inducing points. The time scales quadratically with the number of inducing points. Note that the approximation does not over-fit to the data as the number of parameters increases.

## 9 Discussion

Our approximate inference relates to the Bayesian neural network (Bayesian NN, Mackay, 1992; MacKay, 1992). In the Bayesian NN a prior distribution is placed over the weights of an NN, and a posterior distribution (over the weights and outputs) is sought. The model offers a Bayesian interpretation to the classic NN, with the desired property of uncertainty estimates on the outputs. Inference in Bayesian NNs is generally hard, and approximations to the model are often used (Bishop, 2006, pp 277-290). Our GP approximate inference relates Bayesian NNs and GPs, and can be seen as a method for tractable variational inference in Bayesian NNs with a single hidden layer.

Future research includes the extension of our approximation to deep GPs (Damianou & Lawrence, 2012). We also aim to use the approximate model as a method for adding and removing units in an NN in a principled way. Lastly, we aim to replace the cosines in the Fourier expansion with alternative basis functions and study the resulting approximate model.

### Acknowledgements

The authors would like to thank James Robert Lloyd for some helpful discussions. YG is supported by the Google European Fellowship in Machine Learning. Funding was provided by the EPSRC (grant numbers EP/G050821/1 and EP/L000776/1) and Google (R.E.T.).



## References

- authors, The GPy. GPy: A Gaussian process framework in python. <http://github.com/SheffieldML/GPy>, 2012–2014.
- Bishop, Christopher M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.
- Bochner, Salomon. *Lectures on Fourier integrals*. Number 42. Princeton University Press, 1959.
- Brochu, Eric, Cora, Vlad M, and de Freitas, Nando. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.
- Damianou, Andreas C and Lawrence, Neil D. Deep Gaussian processes. *arXiv preprint arXiv:1211.0358*, 2012.
- Duvenaud, David, Lloyd, James Robert, Grosse, Roger, Tenenbaum, Joshua B., and Ghahramani, Zoubin. Structure discovery in nonparametric regression through compositional kernel search. In *Proceedings of the 30th International Conference on Machine Learning*, June 2013.
- Engel, Yaakov, Mannor, Shie, and Meir, Ron. Reinforcement learning with Gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pp. 201–208. ACM, 2005.
- Gal, Yarin, van der Wilk, Mark, and Rasmussen, Carl. Distributed variational inference in sparse Gaussian process regression and latent variable models. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3257–3265. Curran Associates, Inc., 2014.
- Garofolo, John S, Consortium, Linguistic Data, et al. *TIMIT: acoustic-phonetic continuous speech corpus*. Linguistic Data Consortium, 1993.
- Hensman, James, Fusi, Nicolo, and Lawrence, Neil D. Gaussian processes for big data. In Nicholson, Ann and Smyth, Padhraic (eds.), *UAI*. AUAI Press, 2013.
- Hoffman, Matthew D., Blei, David M., Wang, Chong, and Paisley, John. Stochastic Variational Inference. *Journal Of Machine Learning Research*, 14:1303–1347, MAY 2013. ISSN 1532-4435.
- Keeling, C.D., Whorf, T.P., and the Carbon Dioxide Research Group. Atmospheric CO<sub>2</sub> concentrations (ppmv) derived from in situ air samples collected at Mauna Loa Observatory, Hawaii. Scripps Institution of Oceanography (SIO), University of California, La Jolla, California USA 92093-0444, June 2004.
- Lázaro-Gredilla, Miguel, Quiñero-Candela, Joaquin, Rasmussen, Carl Edward, and Figueiras-Vidal, Aníbal R. Sparse spectrum Gaussian process regression. *The Journal of Machine Learning Research*, 11:1865–1881, 2010.
- Lean, J. Solar irradiance reconstruction. NOAA/NGDC Paleoclimatology Program, Boulder CO, USA, 2004. IGBP PAGES/World Data Center for Paleoclimatology. Data Contribution Series 2004-035.
- Mackay, David. The evidence framework applied to classification networks. *Neural computation*, 4(5):720–736, 1992.
- MacKay, David JC. A practical Bayesian framework for backpropagation networks. *Neural computation*, 4(3): 448–472, 1992.
- Quiñero-Candela, Joaquin and Rasmussen, Carl Edward. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:2005, 2005.
- Rahimi, Ali and Recht, Benjamin. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2007.
- Rasmussen, Carl Edward and Williams, Christopher K. I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006. ISBN 026218253X.
- Rasmussen, Carl Edward, Kuss, Malte, et al. Gaussian processes in reinforcement learning. In *NIPS*, volume 4, pp. 1, 2003.
- Tan, Linda S. L., Ong, Victor M. H., Nott, David J., and Jasra, Ajay. Variational inference for sparse spectrum Gaussian process regression. *arXiv:1306.1999*, 2013.
- Tieleman, T. and Hinton, G. Lecture 6.5 - rmsprop, COURSERA: Neural networks for machine learning, 2012.
- Titsias, Michalis and Lawrence, Neil. Bayesian Gaussian process latent variable model. *Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 6:844–851, 2010.
- Titsias, Michalis K. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 567–574, 2009.

Turner, Richard E. *Statistical Models for Natural Sounds*.  
PhD thesis, Gatsby Computational Neuroscience Unit,  
UCL, 2010.

Wilson, Andrew, Gilboa, Elad, Cunningham, John P, and  
Nehorai, Arye. Fast kernel learning for multidimen-  
sional pattern extrapolation. In *Advances in Neural In-  
formation Processing Systems*, pp. 3626–3634, 2014.

Zhu, Ciyou, Byrd, Richard H, Lu, Peihuang, and Nocedal,  
Jorge. Algorithm 778: L-BFGS-B: Fortran subroutines  
for large-scale bound-constrained optimization. *ACM  
Transactions on Mathematical Software (TOMS)*, 23(4):  
550–560, 1997.