

# Optimizing Gaussian Processes

## Honours Research Project

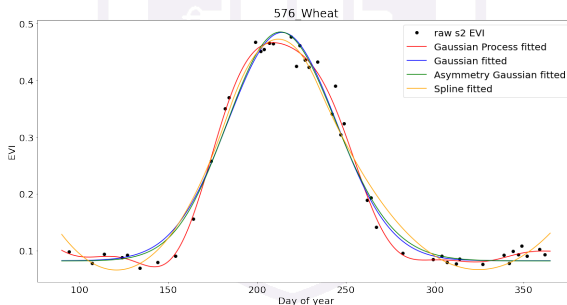
Michael Ciccotosto-Camp - 44302913



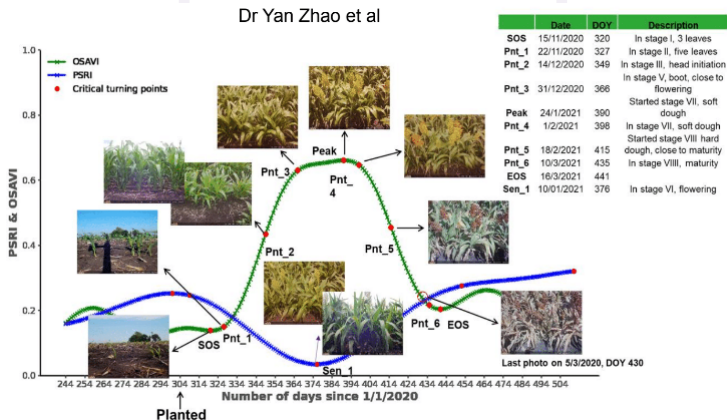
THE UNIVERSITY  
OF QUEENSLAND  
AUSTRALIA

# Problem Setting and Motivation

- The idea of studying time series prediction came from a research group from the Gatton campus, lead by Dr Potgieter, analysing crop growth from previous seasons to forecast when certain phenological stages will take place in the current harvest.



- Photo courtesy of Dr Andries Potgieter and Yan Zhao.



# Introduction to Gaussian Processes

- A Gaussian Process (GP) is a collection of random variables with index set  $I$ , such that every finite subset of random variables has a joint Gaussian distribution and are completely characterized by a mean function  $m : X \rightarrow \mathbb{R}$  and a kernel  $k : X \times X \rightarrow \mathbb{R}$  (in this context, think of the kernel as a function that provides some notion of similarity between points).

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))].$$

# Predictions

- Using the assumption that our data can be modelled as a Gaussian process, we can write out the new distribution of the observed noisy values along the points at which we wish to test the underlying function as

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_\star \end{bmatrix} \sim \mathbb{N} \left( \mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{X}\mathbf{X}} + \sigma_n^2 \mathbb{I}_{n \times n} & \mathbf{K}_{\mathbf{X}_\star \mathbf{X}} \\ \mathbf{K}_{\mathbf{X}_\star \mathbf{X}} & \mathbf{K}_{\mathbf{X}_\star \mathbf{X}_\star} \end{bmatrix} \right).$$

(using the notation  $(\mathbf{K}_{\mathbf{W}\mathbf{W}'} )_{i,j} \triangleq k(\mathbf{w}_i, \mathbf{w}'_j)$ )

# Predictions

- Using the assumption that our data can be modelled as a Gaussian process, we can write out the new distribution of the observed noisy values along the points at which we wish to test the underlying function as

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathbb{N} \left( \mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{X}\mathbf{X}} + \sigma_n^2 \mathbb{I}_{n \times n} & \mathbf{K}_{\mathbf{X}_* \mathbf{X}}^\top \\ \mathbf{K}_{\mathbf{X}_* \mathbf{X}} & \mathbf{K}_{\mathbf{X}_* \mathbf{X}_*} \end{bmatrix} \right).$$

(using the notation  $(\mathbf{K}_{\mathbf{W}\mathbf{W}'} )_{i,j} \triangleq k(\mathbf{w}_i, \mathbf{w}'_j)$ )

- The mean and covariance can then be computed as

$$\begin{aligned} \bar{\mathbf{y}}_* &= \mathbf{K}_{\mathbf{X}_* \mathbf{X}} \left[ \mathbf{K}_{\mathbf{X}\mathbf{X}} + \sigma_n^2 \mathbb{I}_{n \times n} \right]^{-1} \mathbf{y} \\ \text{cov}(\mathbf{y}_*) &= \mathbf{K}_{\mathbf{X}_* \mathbf{X}_*} - \mathbf{K}_{\mathbf{X}_* \mathbf{X}} \left[ \mathbf{K}_{\mathbf{X}\mathbf{X}} + \sigma_n^2 \mathbb{I}_{n \times n} \right]^{-1} \mathbf{K}_{\mathbf{X}_* \mathbf{X}}^\top. \end{aligned}$$

# Unoptimized GPR

---

## Algorithm 1: Unoptimized GPR

---

**input** : Observations  $\mathbf{X}, \mathbf{y}$  and a test input  $\mathbf{x}_*$ .

**output**: A prediction  $\bar{f}_*$  with its corresponding variance  $\mathbb{V}[f_*]$ .

- 1  $\mathbf{L} = \text{cholesky}(\mathbf{K}_{\mathbf{X}\mathbf{X}} + \sigma_n^2 \mathbb{I}_{n \times n})$
  - 2  $\boldsymbol{\alpha} = \text{lin-solve}(\mathbf{L}^\top, \text{lin-solve}(\mathbf{L}, \mathbf{y}))$
  - 3  $\bar{\mathbf{y}}_* = \mathbf{K}_{\mathbf{x}_* \mathbf{X}} \boldsymbol{\alpha}$
  - 4  $\mathbf{v} = \text{lin-solve}(\mathbf{L}, \mathbf{K}_{\mathbf{x}_* \mathbf{x}_*})$
  - 5  $\mathbb{V}[f_*] = \mathbf{K}_{\mathbf{x}_* \mathbf{x}_*} - \mathbf{v}^\top \mathbf{v}$
  - 6 **return**  $\bar{f}_*, \mathbb{V}[f_*]$
-

# Problems with Unoptimized GPR

---

## Algorithm 2: Unoptimized GPR

---

**input** : Observations  $\mathbf{X}, \mathbf{y}$  and a prediction inputs  $\mathbf{x}_*$ .

**output**: A prediction  $\bar{f}_*$  with its corresponding variance  $\mathbb{V}[f_*]$ .

---

- 1  $\mathbf{L} = \text{cholesky}(\mathbf{K}_{\mathbf{X}\mathbf{X}} + \sigma_n^2 \mathbb{I}_{n \times n})$
  - 2  $\boldsymbol{\alpha} = \text{lin-solve}(\mathbf{L}^\top, \text{lin-solve}(\mathbf{L}, \mathbf{y}))$
  - 3  $\bar{f}_* = \mathbf{K}_{\mathbf{x}_* \mathbf{X}} \boldsymbol{\alpha}$
  - 4  $\mathbf{v} = \text{lin-solve}(\mathbf{L}, \mathbf{K}_{\mathbf{x}_* \mathbf{x}_*})$
  - 5  $\mathbb{V}[f_*] = \mathbf{K}_{\mathbf{x}_* \mathbf{x}_*} - \mathbf{v}^\top \mathbf{v}$
  - 6 **return**  $\bar{f}_*, \mathbb{V}[f_*]$
- 

- Lines 1, 2 and 4 can be incredibly slow as computing  $\mathbf{K}_{\mathbf{X}\mathbf{X}}$  doing a Cholesky decomposition and performing linear solves scale poorly as the number of inputs,  $n$ , grows.



# Nystrom Approximation

- The Nystrom method we seek a matrix  $\mathbf{Q} \in \mathbb{R}^{n \times k}$  that satisfies  $\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^* \mathbf{A}\|_F \leq \varepsilon$ , where  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is positive semi definite matrix, to form the rank- $k$  approximation

# Nystrom Approximation

- The Nystrom method we seek a matrix  $\mathbf{Q} \in \mathbb{R}^{n \times k}$  that satisfies  $\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^* \mathbf{A}\|_F \leq \varepsilon$ , where  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is positive semi definite matrix, to form the rank- $k$  approximation

$$\mathbf{A} \simeq \mathbf{Q}\mathbf{Q}^* \mathbf{A}$$

# Nystrom Approximation

- The Nystrom method we seek a matrix  $\mathbf{Q} \in \mathbb{R}^{n \times k}$  that satisfies  $\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^* \mathbf{A}\|_F \leq \varepsilon$ , where  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is positive semi definite matrix, to form the rank- $k$  approximation

$$\begin{aligned}\mathbf{A} &\simeq \mathbf{Q}\mathbf{Q}^* \mathbf{A} \\ &\simeq \mathbf{Q}(\mathbf{Q}^* \mathbf{A} \mathbf{Q}) \mathbf{Q}^*\end{aligned}$$

# Nystrom Approximation

- The Nystrom method we seek a matrix  $\mathbf{Q} \in \mathbb{R}^{n \times k}$  that satisfies  $\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^* \mathbf{A}\|_F \leq \varepsilon$ , where  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is positive semi definite matrix, to form the rank- $k$  approximation

$$\begin{aligned}\mathbf{A} &\simeq \mathbf{Q}\mathbf{Q}^* \mathbf{A} \\ &\simeq \mathbf{Q} (\mathbf{Q}^* \mathbf{A} \mathbf{Q}) \mathbf{Q}^* \\ &= \mathbf{Q} (\mathbf{Q}^* \mathbf{A} \mathbf{Q}) (\mathbf{Q}^* \mathbf{A} \mathbf{Q})^\dagger (\mathbf{Q}^* \mathbf{A} \mathbf{Q}) \mathbf{Q}^*\end{aligned}$$

# Nystrom Approximation

- The Nystrom method we seek a matrix  $\mathbf{Q} \in \mathbb{R}^{n \times k}$  that satisfies  $\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^* \mathbf{A}\|_F \leq \varepsilon$ , where  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is positive semi definite matrix, to form the rank- $k$  approximation

$$\begin{aligned}\mathbf{A} &\simeq \mathbf{Q}\mathbf{Q}^* \mathbf{A} \\ &\simeq \mathbf{Q} (\mathbf{Q}^* \mathbf{A} \mathbf{Q}) \mathbf{Q}^* \\ &= \mathbf{Q} (\mathbf{Q}^* \mathbf{A} \mathbf{Q}) (\mathbf{Q}^* \mathbf{A} \mathbf{Q})^\dagger (\mathbf{Q}^* \mathbf{A} \mathbf{Q}) \mathbf{Q}^* \\ &\simeq (\mathbf{A} \mathbf{Q}) (\mathbf{Q}^* \mathbf{A} \mathbf{Q})^\dagger (\mathbf{Q}^* \mathbf{A}).\end{aligned}$$

# Random Fourier Feature Approximation

- The RFF technique hinges on Bochners theorem which characterises positive definite functions (namely kernels) and states that any positive definite functions can be represented as

$$k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y}) = \int_{\mathbb{C}^d} \exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle) \mu_k(d\boldsymbol{\omega})$$

where  $\mu_k$  is a positive finite measure on the frequencies of  $\boldsymbol{\omega}$ .

# Random Fourier Feature Approximation

- The RFF technique hinges on Bochners theorem which characterises positive definite functions (namely kernels) and states that any positive definite functions can be represented as

$$k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y}) = \int_{\mathbb{C}^d} \exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle) \mu_k(d\boldsymbol{\omega})$$

where  $\mu_k$  is a positive finite measure on the frequencies of  $\boldsymbol{\omega}$ .

- This integral can then be approximated via the following Monte Carlo estimate

$$k(\mathbf{x} - \mathbf{y}) = \int_{\mathbb{C}^d} \exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle) p(\boldsymbol{\omega}) d\boldsymbol{\omega}$$

# Random Fourier Feature Approximation

- The RFF technique hinges on Bochners theorem which characterises positive definite functions (namely kernels) and states that any positive definite functions can be represented as

$$k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y}) = \int_{\mathbb{C}^d} \exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle) \mu_k(d\boldsymbol{\omega})$$

where  $\mu_k$  is a positive finite measure on the frequencies of  $\boldsymbol{\omega}$ .

- This integral can then be approximated via the following Monte Carlo estimate

$$\begin{aligned} k(\mathbf{x} - \mathbf{y}) &= \int_{\mathbb{C}^d} \exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle) p(\boldsymbol{\omega}) d\boldsymbol{\omega} \\ &= \mathbb{E}_{\boldsymbol{\omega} \sim p(\cdot)} (\exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle)) \end{aligned}$$



# Random Fourier Feature Approximation

- The RFF technique hinges on Bochners theorem which characterises positive definite functions (namely kernels) and states that any positive definite functions can be represented as

$$k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y}) = \int_{\mathbb{C}^d} \exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle) \mu_k(d\boldsymbol{\omega})$$

where  $\mu_k$  is a positive finite measure on the frequencies of  $\boldsymbol{\omega}$ .

- This integral can then be approximated via the following Monte Carlo estimate

$$\begin{aligned} k(\mathbf{x} - \mathbf{y}) &= \int_{\mathbb{C}^d} \exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle) p(\boldsymbol{\omega}) d\boldsymbol{\omega} \\ &= \mathbb{E}_{\boldsymbol{\omega} \sim p(\cdot)} (\exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle)) \\ &\simeq \frac{1}{D} \sum_{j=1}^D \exp(i\langle \boldsymbol{\omega}_j, \mathbf{x} - \mathbf{y} \rangle) \end{aligned}$$

# Random Fourier Feature Approximation

- The RFF technique hinges on Bochners theorem which characterises positive definite functions (namely kernels) and states that any positive definite functions can be represented as

$$k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y}) = \int_{\mathbb{C}^d} \exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle) \mu_k(d\boldsymbol{\omega})$$

where  $\mu_k$  is a positive finite measure on the frequencies of  $\boldsymbol{\omega}$ .

- This integral can then be approximated via the following Monte Carlo estimate

$$\begin{aligned} k(\mathbf{x} - \mathbf{y}) &= \int_{\mathbb{C}^d} \exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle) p(\boldsymbol{\omega}) d\boldsymbol{\omega} \\ &= \mathbb{E}_{\boldsymbol{\omega} \sim p(\cdot)} (\exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle)) \\ &\simeq \frac{1}{D} \sum_{j=1}^D \exp(i\langle \boldsymbol{\omega}_j, \mathbf{x} - \mathbf{y} \rangle) \\ &= \sum_{j=1}^D \left( \frac{1}{\sqrt{D}} \exp(i\langle \boldsymbol{\omega}_j, \mathbf{x} \rangle) \right) \overline{\left( \frac{1}{\sqrt{D}} \exp(i\langle \boldsymbol{\omega}_j, \mathbf{y} \rangle) \right)} \end{aligned}$$

# Random Fourier Feature Approximation

- The RFF technique hinges on Bochners theorem which characterises positive definite functions (namely kernels) and states that any positive definite functions can be represented as

$$k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y}) = \int_{\mathbb{C}^d} \exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle) \mu_k(d\boldsymbol{\omega})$$

where  $\mu_k$  is a positive finite measure on the frequencies of  $\boldsymbol{\omega}$ .

- This integral can then be approximated via the following Monte Carlo estimate

$$\begin{aligned} k(\mathbf{x} - \mathbf{y}) &= \int_{\mathbb{C}^d} \exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle) p(\boldsymbol{\omega}) d\boldsymbol{\omega} \\ &= \mathbb{E}_{\boldsymbol{\omega} \sim p(\cdot)} (\exp(i\langle \boldsymbol{\omega}, \mathbf{x} - \mathbf{y} \rangle)) \\ &\simeq \frac{1}{D} \sum_{j=1}^D \exp(i\langle \boldsymbol{\omega}_j, \mathbf{x} - \mathbf{y} \rangle) \\ &= \sum_{j=1}^D \left( \frac{1}{\sqrt{D}} \exp(i\langle \boldsymbol{\omega}_j, \mathbf{x} \rangle) \right) \overline{\left( \frac{1}{\sqrt{D}} \exp(i\langle \boldsymbol{\omega}_j, \mathbf{y} \rangle) \right)} \\ &= \langle \boldsymbol{\varphi}(\mathbf{x}), \boldsymbol{\varphi}(\mathbf{y}) \rangle_{\mathbb{C}^D} \end{aligned}$$

# Krylov Subspace Methods

- $Ax^* = b.$



# Krylov Subspace Methods

- $\mathbf{Ax}^* = \mathbf{b}$ .
- $\mathbf{x}^* \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{v})$  where  $\mathcal{K}_k(\mathbf{A}, \mathbf{v}) = \text{l.s}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \dots, \mathbf{A}^{k-1}\mathbf{r}_0\}$ .

# Krylov Subspace Methods

- $\mathbf{A}\mathbf{x}^* = \mathbf{b}$ .
- $\mathbf{x}^* \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{v})$  where  $\mathcal{K}_k(\mathbf{A}, \mathbf{v}) = \text{l.s}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \dots, \mathbf{A}^{k-1}\mathbf{r}_0\}$ .
- CG:  $\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{A}}$  is minimized.

# Krylov Subspace Methods

- $\mathbf{Ax}^* = \mathbf{b}$ .
- $\mathbf{x}^* \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{v})$  where  $\mathcal{K}_k(\mathbf{A}, \mathbf{v}) = \text{l.s}\{\mathbf{r}_0, \mathbf{Ar}_0, \mathbf{A}^2\mathbf{r}_0, \dots, \mathbf{A}^{k-1}\mathbf{r}_0\}$ .
- CG:  $\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{A}}$  is minimized.
- MINRES:  $\|\mathbf{Ax} - \mathbf{b}\|_2$  is minimized.

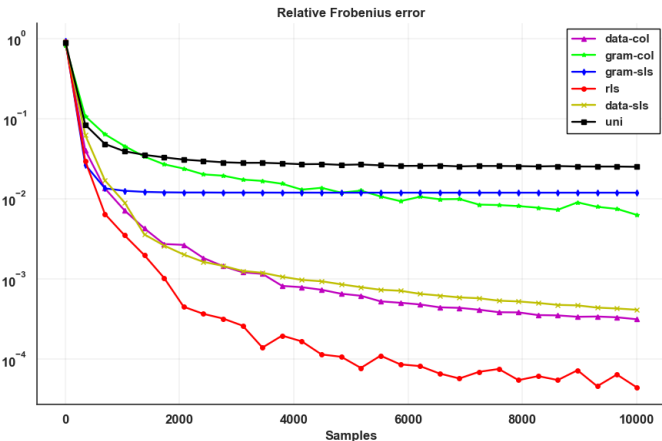


Figure: 3D-Spatial Network dataset using Nystrom



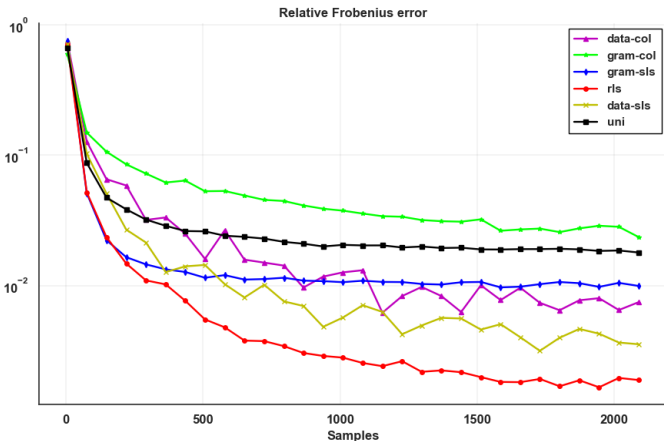


Figure: Abalone dataset using Nystrom

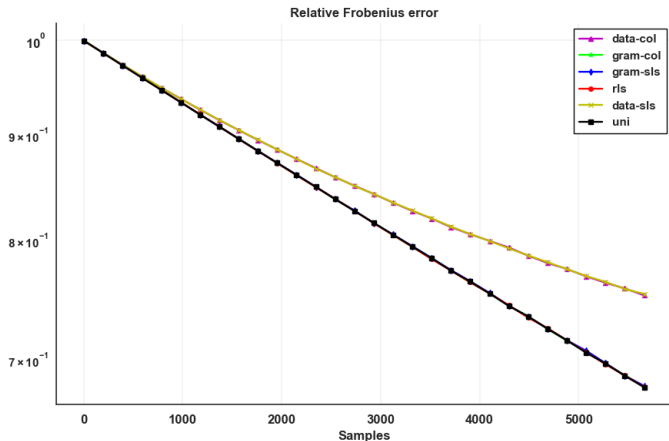


Figure: Temperature dataset using Nystrom

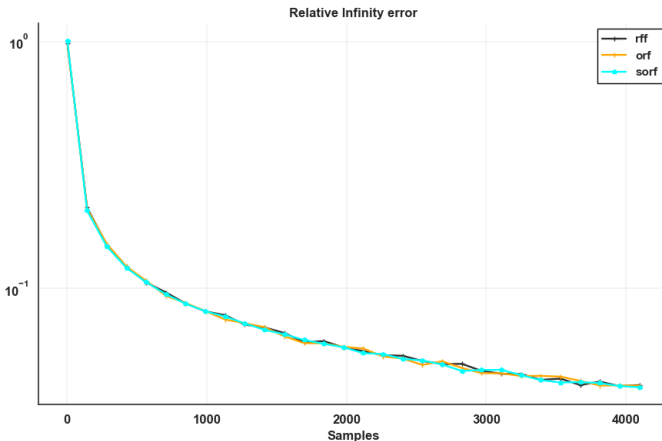


Figure: 3D-Spatial Network dataset using RFF

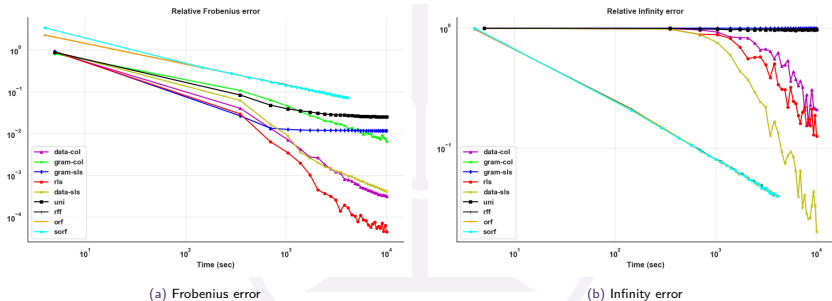
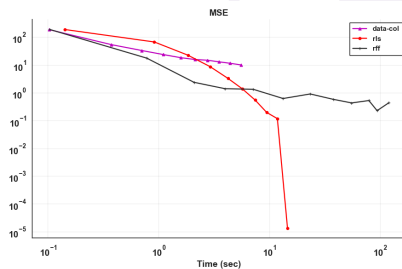


Figure: Comparison between Nystrom and RFF approximations for the 3D-Spatial network data.

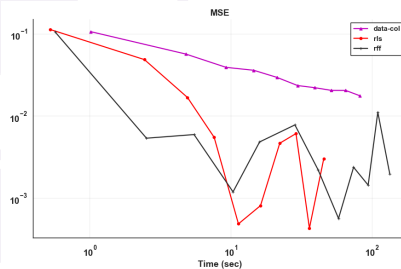
- How do Nystrom and RFF methods compare in terms of prediction?



- How do Nystrom and RFF methods compare in terms of prediction?



(a) Stock Market dataset



(b) Temperature dataset

Figure: Comparison between Nystrom and RFF approximations in GP prediction.

# How do MINRES and CG methods compare in terms of prediction?

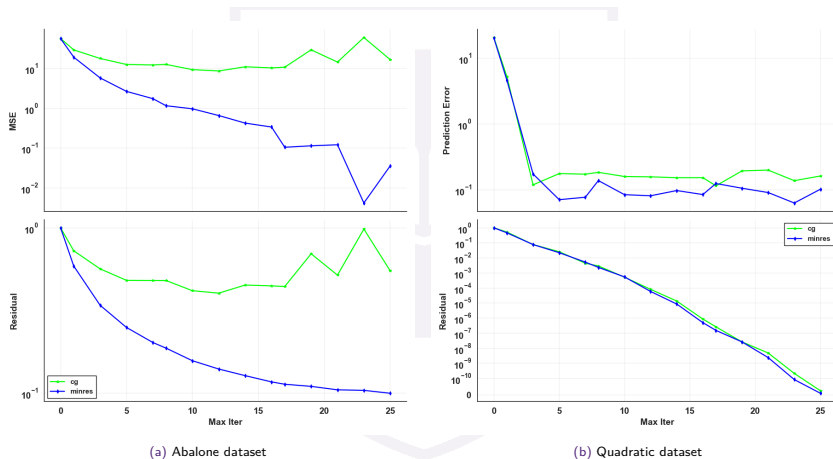
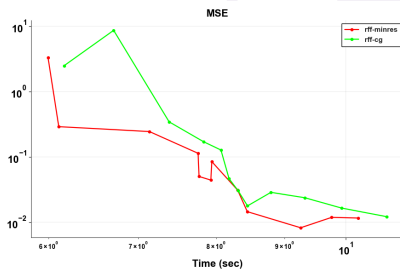
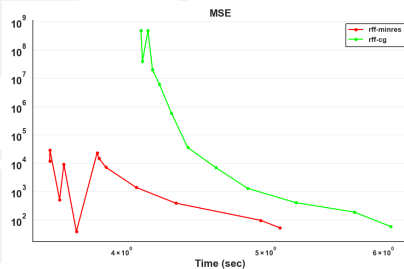


Figure: Comparison between MINRES and CG in GP prediction.

- Using approximation techniques together.



(a) Stock Market dataset



(b) Abalone dataset

**Figure:** Comparison between CG and MINRES when paried with RFF.



- $\|K_{X_*} x \rho - y_*\|_2^2$ , where  $\rho$  is our best estimate for  $[K_{XX} + \sigma_n^2 \mathbf{1}_{n \times n}] \rho = y$

- $\|K_{X_*} x \rho - y_*\|_2^2$ , where  $\rho$  is our best estimate for  $[K_{XX} + \sigma_n^2 \mathbf{1}_{n \times n}] \rho = y$
- $\|[K_{XX} + \sigma_n^2 \mathbf{1}_{n \times n}] \rho - y\|_2^2$

# Moving Forward

- Write these results up.
- Apply our findings to our initial remote sensing task.
- Look at multi-output Gaussian Processes for remote sensing.