Optimizing Gaussian Processes

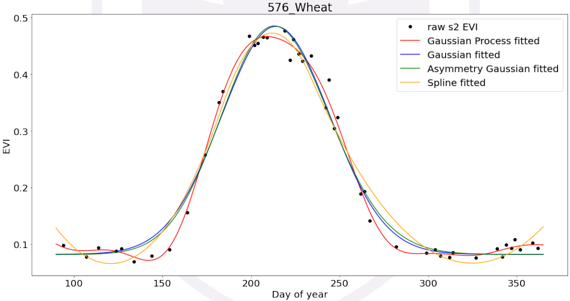
Honours Research Project

Michael Ciccotosto-Camp - 44302913

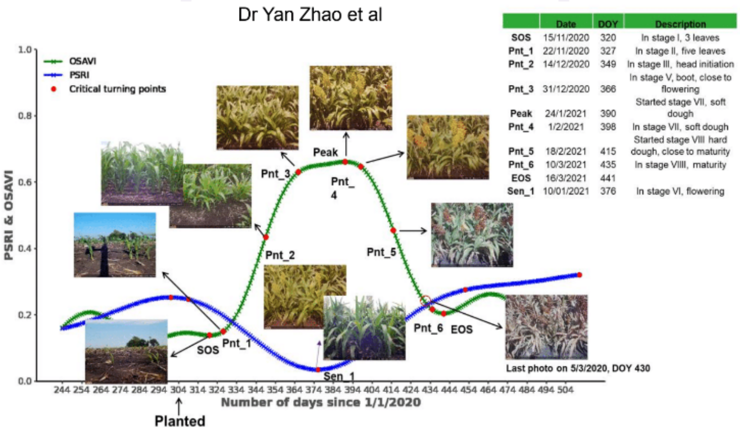
THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Problem Setting and Motivation

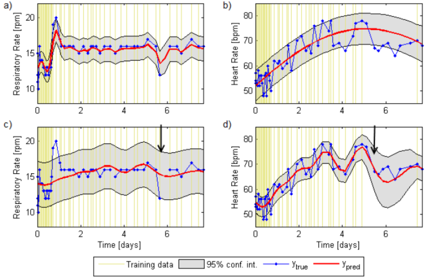
- A survey of various parametric models to compare with Gaussian processes.



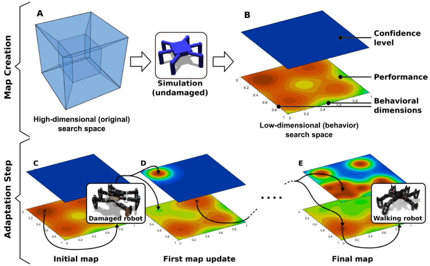
- Photo courtesy of A/Prof Andries Potgieter and Dr Yan Zhao.



	Date	DOY	Description
SOS	15/11/2020	320	In stage I, 3 leaves
Pnt_1	22/11/2020	327	In stage II, five leaves
Pnt_2	14/12/2020	349	In stage III, head initiation
Pnt_3	31/12/2020	366	In stage V, boot, close to flowering
Peak	24/1/2021	390	Started stage VII, soft dough
Pnt_4	1/2/2021	398	In stage VII, soft dough
Pnt_5	18/2/2021	415	Started stage VIII hard dough, close to maturity
Pnt_6	10/3/2021	435	In stage VIIII, maturity
EOS	16/3/2021	441	
Sen_1	10/01/2021	376	In stage VI, flowering



(a)



(b)

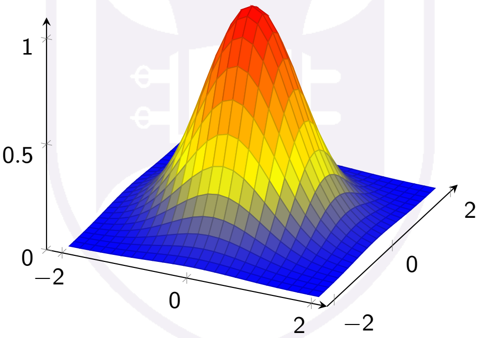
- (A) Gaussian processes used to filter noise from medical data, photo courtesy of R. Durichen etal. (B) Gaussian processes used to help robots adapt to physical impairments, photo courtesy of A. Cully etal.

Introduction to Gaussian Processes

- Gaussian Process (GP) are completely characterized by a mean function $m : X \to \mathbb{R}$ and a kernel $k : X \times X \to \mathbb{R}$.

$$
\begin{aligned}
m(\boldsymbol{x}) &= \mathbb{E}\left[f(\boldsymbol{x})\right] \\
k(\boldsymbol{x}, \boldsymbol{x}') &= \mathbb{E}\left[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))\right].
\end{aligned}
$$

- A very common kernel function used is the RBF or Gaussian kernel.



Predictions

- Our data and novel points should form a joint Gaussian distribution

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

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

Predictions

- Our data and novel points should form a joint Gaussian distribution

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

(using the notation $(\boldsymbol{K}_{\boldsymbol{WW}'})_{i,j} \triangleq k\left(\boldsymbol{w}_i, \boldsymbol{w}_j'\right))$

- The mean and covariance can then be computed as

$$\begin{aligned} \overline{\boldsymbol{y}_\star} &= \boldsymbol{K}_{\boldsymbol{X}_\star \boldsymbol{X}} \left[\boldsymbol{K}_{\boldsymbol{XX}} + \sigma_n^2 \mathbb{I}_{n\times n}\right]^{-1} \boldsymbol{y} \\ \operatorname{cov}(\boldsymbol{y}_\star) &= \boldsymbol{K}_{\boldsymbol{X}_\star \boldsymbol{X}_\star} - \boldsymbol{K}_{\boldsymbol{X}_\star \boldsymbol{X}} \left[\boldsymbol{K}_{\boldsymbol{XX}} + \sigma_n^2 \mathbb{I}_{n\times n}\right]^{-1} \boldsymbol{K}_{\boldsymbol{X}_\star \boldsymbol{X}}^\intercal. \end{aligned}$$

Unoptimized GPR

Algorithm 1: Unoptimized GPR

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

output: A prediction $\overline{y_\star}$ with its corresponding variance $\mathbb{V}[y_\star]$.

$\boldsymbol{L} = \text{cholesky}\left(\boldsymbol{K}_{\boldsymbol{XX}} + \sigma_n^2 \mathbb{I}_{n\times n}\right)$

$\alpha = \text{lin-solve}\left(\boldsymbol{L}^\intercal, \text{lin-solve}\left(\boldsymbol{L}, \boldsymbol{y}\right)\right)$

$\overline{y_\star} = \boldsymbol{K}_{\boldsymbol{x}_\star \boldsymbol{X}} \alpha$

$\boldsymbol{v} = \text{lin-solve}\left(\boldsymbol{L}, \boldsymbol{K}_{\boldsymbol{x}_\star \boldsymbol{X}}\right)$

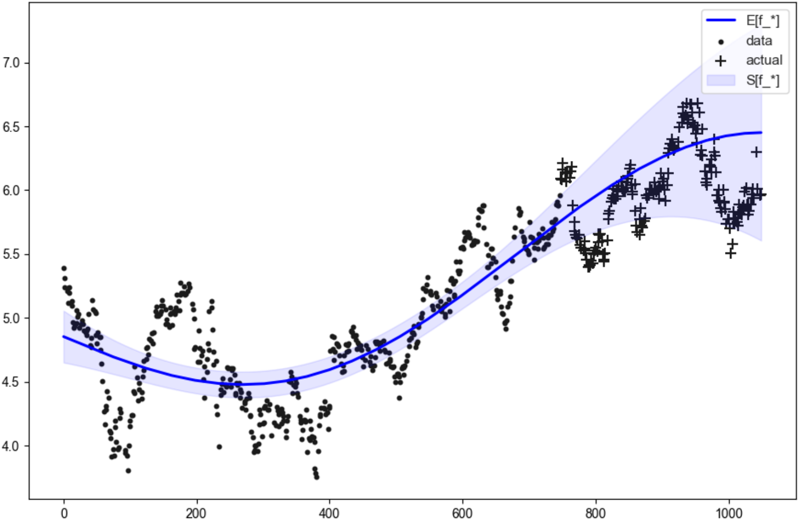
$\mathbb{V}[y_\star] = \boldsymbol{K}_{\boldsymbol{x}_\star \boldsymbol{x}_\star} - \boldsymbol{v}^\intercal \boldsymbol{v}$

return $\overline{y_\star}, \mathbb{V}[y_\star]$

Implementation

```python
def gp_reg_pred(X_train, Y_train, x_pred, sigma):
    n, d = X_train.shape
    # Create the Gram matrix corresponding to the training data set.
    K = exact_kernel(X_train, sigma=sigma)
    # Noise variance of labels.
    s = np.var(Y_train.squeeze())
    L = np.linalg.cholesky(K + s*np.eye(n))
    # Compute the mean at our test points.
    Lk = np.linalg.solve(L, exact_kernel(X_train, x_pred, sigma=sigma))
    Ef = np.dot(Lk.T, np.linalg.solve(L, Y_train))
    # Compute the variance at our test points.
    K_ = exact_kernel(x_pred, sigma=sigma)
    Vf = np.diag(K_) - np.sum(Lk**2, axis=0)
    return Ef, Vf
```

Stock Market Prediction



Problems with Unoptimized GPR

Algorithm 2: Unoptimized GPR

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

output: A prediction $\overline{y_\star}$ with its corresponding variance $\mathbb{V}[y_\star]$.

$\boldsymbol{L} = \text{cholesky}\left(\boldsymbol{K}_{\boldsymbol{XX}} + \sigma_n^2 \mathbb{I}_{n\times n}\right)$

$\alpha = \text{lin-solve}\left(\boldsymbol{L}^\intercal, \text{lin-solve}\left(\boldsymbol{L}, \boldsymbol{y}\right)\right)$

$\overline{y_\star} = \boldsymbol{K}_{\boldsymbol{x}_\star \boldsymbol{X}} \alpha$

$\boldsymbol{v} = \text{lin-solve}\left(\boldsymbol{L}, \boldsymbol{K}_{\boldsymbol{x}_\star \boldsymbol{X}}\right)$

$\mathbb{V}[y_\star] = \boldsymbol{K}_{\boldsymbol{x}_\star \boldsymbol{x}_\star} - \boldsymbol{v}^\intercal \boldsymbol{v}$

return $\overline{y_\star}, \mathbb{V}[y_\star]$

- The bottle necks that we would like to address are the computation of $\boldsymbol{K_{XX}}$ and the Cholesky decomposition.

- The bottle necks that we would like to address are the computation of K_{XX} and the Cholesky decomposition.
- Exact computation of K_{XX} replaced with Nystrom and RFF estimates.

- The bottle necks that we would like to address are the computation of $\boldsymbol{K_{XX}}$ and the Cholesky decomposition.
- Exact computation of $\boldsymbol{K_{XX}}$ replaced with Nystrom and RFF estimates.
- Looked at CG and MINRES to replace Cholesky decomposition.

Nystrom Approximation

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

Nystrom Approximation

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

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

Nystrom Approximation

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

$$\begin{aligned} \boldsymbol{A} &\simeq \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A} \\ &\simeq \boldsymbol{Q}\left(\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{Q}\right)\boldsymbol{Q}^* \end{aligned}$$

Nystrom Approximation

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

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

Nystrom Approximation

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

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

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

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

where μ_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

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

  where μ_k is a positive finite measure on the frequencies of $\boldsymbol{\omega}$.
- This integral can then be approximated via the following Monte Carlo estimate

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

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

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

  where μ_k is a positive finite measure on the frequencies of $\boldsymbol{\omega}$.
- This integral can then be approximated via the following Monte Carlo estimate

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Krylov Subspace Methods

- $\boldsymbol{A}\boldsymbol{x}^{\star} = \boldsymbol{b}$.

Krylov Subspace Methods

- $\boldsymbol{A}\boldsymbol{x}^\star = \boldsymbol{b}$.
- $\boldsymbol{x}^\star \in \boldsymbol{x}_0 + \mathcal{K}_n\left(\boldsymbol{A}, \boldsymbol{v}\right)$ where $\mathcal{K}_k\left(\boldsymbol{A}, \boldsymbol{v}\right) = \mathrm{l.\,s}\left\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0\right\}$.

Krylov Subspace Methods

- $\boldsymbol{A}\boldsymbol{x}^{\star} = \boldsymbol{b}$.
- $\boldsymbol{x}^{\star} \in \boldsymbol{x}_0 + \mathcal{K}_n\left(\boldsymbol{A}, \boldsymbol{v}\right)$ where $\mathcal{K}_k\left(\boldsymbol{A}, \boldsymbol{v}\right) = \mathrm{l.s}\left\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0\right\}$.
- CG: $\|\boldsymbol{x} - \boldsymbol{x}^{\star}\|_{\boldsymbol{A}}$ is minimized.

Krylov Subspace Methods

- $\boldsymbol{A}\boldsymbol{x}^{\star} = \boldsymbol{b}$.
- $\boldsymbol{x}^{\star} \in \boldsymbol{x}_0 + \mathcal{K}_n\left(\boldsymbol{A}, \boldsymbol{v}\right)$ where $\mathcal{K}_k\left(\boldsymbol{A}, \boldsymbol{v}\right) = \mathrm{l.s}\left\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0\right\}$.
- CG: $\|\boldsymbol{x} - \boldsymbol{x}^{\star}\|_{\boldsymbol{A}}$ is minimized.
- MINRES: $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$ is minimized.

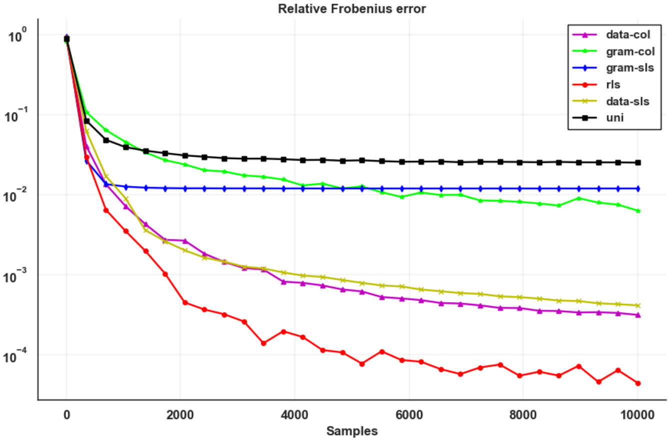


Figure: 3D-Spatial Network dataset using Nystrom

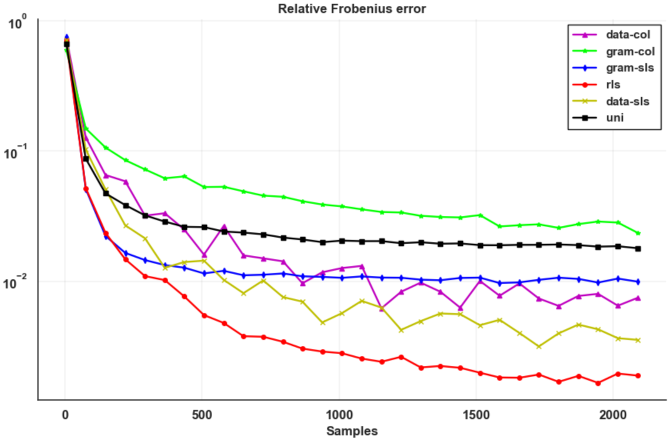


Figure: Abalone dataset using Nystrom

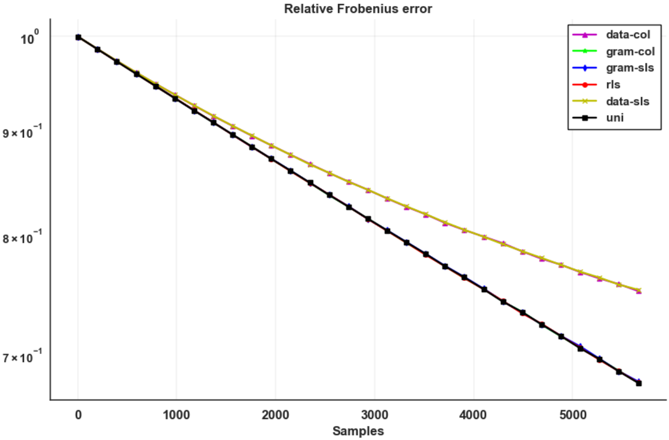


Figure: Temperature dataset using Nystrom

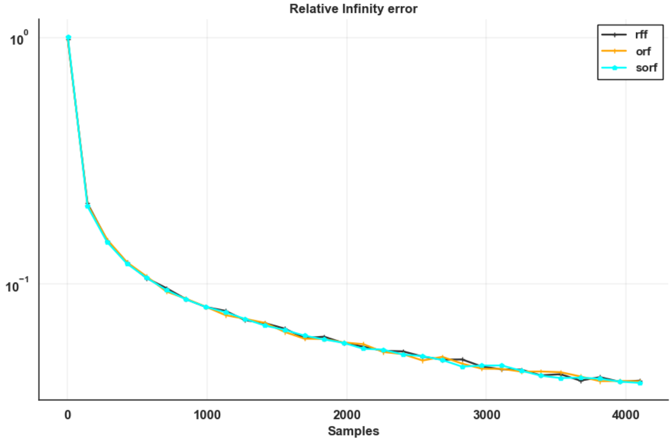
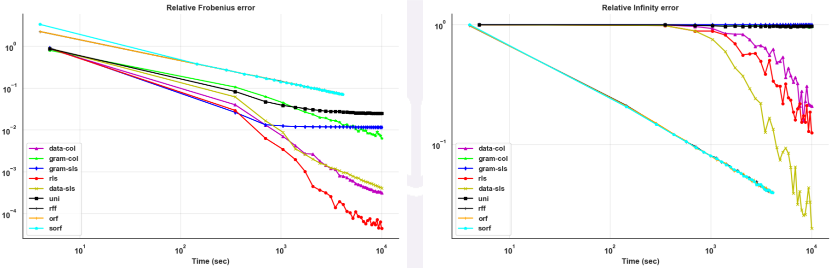


Figure: 3D-Spatial Network dataset using RFF

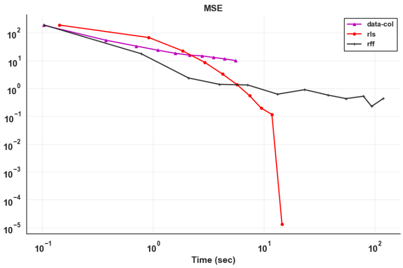


(a) Frobenius error (b) Infinity error

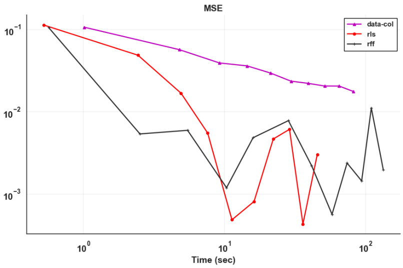
Figure: Comparison between Nystrom and RFF approximations for the 3D-Spatial network data.

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

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



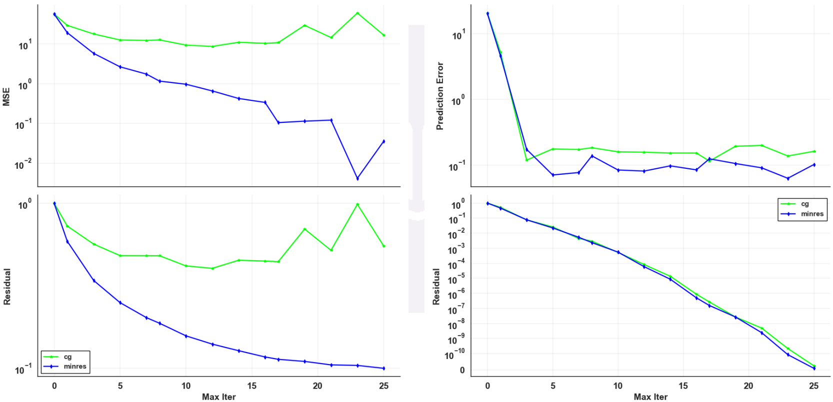
(a) Stock Market dataset



(b) Temperature dataset

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

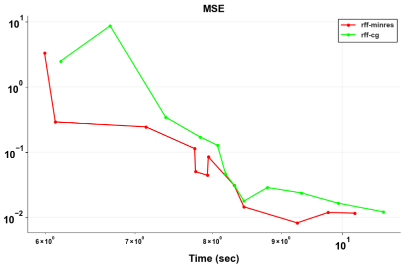
- How do MINRES and CG methods compare in terms of prediction?



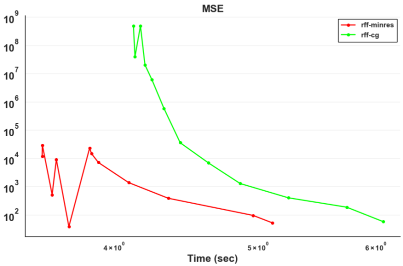
(a) Abalone dataset (b) Quadratic dataset

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.

- $\left\|\boldsymbol{K_{X_\star X}}\boldsymbol{\rho} - \boldsymbol{y}_\star\right\|_2^2$, where $\boldsymbol{\rho}$ is our best estimate for $\left[\boldsymbol{K_{XX}} + \sigma_n^2 \mathbf{1}_{n\times n}\right]\boldsymbol{\rho} = \boldsymbol{y}$

- $\left\|\boldsymbol{K}_{\boldsymbol{X}_\star \boldsymbol{X}} \boldsymbol{\rho} - \boldsymbol{y}_\star\right\|_2^2$, where $\boldsymbol{\rho}$ is our best estimate for $\left[\boldsymbol{K}_{\boldsymbol{XX}} + \sigma_n^2 \mathbf{1}_{n\times n}\right] \boldsymbol{\rho} = \boldsymbol{y}$

- $\left\| \left[\boldsymbol{K}_{\boldsymbol{XX}} + \sigma_n^2 \mathbf{1}_{n\times n}\right] \boldsymbol{\rho} - \boldsymbol{y}\right\|_2^2$

Moving Forward

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

Dr Yan Zhao et al

	Date	DOY	Description
SOS	15/11/2020	320	In stage I, 3 leaves
Pnt_1	22/11/2020	327	In stage II, five leaves
Pnt_2	14/12/2020	349	In stage III, head initiation
Pnt_3	31/12/2020	366	In stage V, boot, close to flowering
Peak	24/1/2021	390	Started stage VII, soft dough
Pnt_4	1/2/2021	398	In stage VII, soft dough
Pnt_5	18/2/2021	415	Started stage VIII hard dough, close to maturity
Pnt_6	10/3/2021	435	In stage VIIII, maturity
EOS	16/3/2021	441	
Sen_1	10/01/2021	376	In stage VI, flowering