Random Features, Density Estimate and Simultaneous Localization and Mapping

Yermek Kapushev

y.kapushev@skoltech.ru

Skolkovo Institute of Science and Technology
Moscow, Russia

Moscow, 2020

Skolkovo Institute of Science and Technology

Kernel Trick

- Data set

$$(\mathbf{X}, \mathbf{y}) = \{(x_1, y_1), \dots, (x_n, y_n)\} \in \mathcal{X} \times \mathbb{R}, \quad y_i = f(x_i) + \varepsilon_i, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

- Let $\phi(x) : \mathcal{X} \to \mathbb{R}^d$ be some feature map
- Suppose that $y = \beta^\top \phi(x)$, then prediction is given by

$$\hat{f}(x^*) = \phi(x^*)^\top \left(\phi(\mathbf{X})^\top \phi(\mathbf{X}) + \lambda \mathbf{I}\right)^{-1} \phi(\mathbf{X})^\top \mathbf{y}, \qquad \mathcal{O}(d^3)$$

- Let $k(x, x') = \langle \phi(x), \phi(x') \rangle$ be a *kernel function*. Then prediction can be rewritten as

$$\hat{f}(x^*) = \mathbf{k}_*^\top \left(\mathbf{K} + \lambda \mathbf{I}\right)^{-1} \mathbf{y}, \qquad \mathcal{O}(n^3)$$

where $\mathbf{k}_* = (k(x^*, x_1), \dots, k(x^*, x_n))$, $\mathbf{K}_{ij} = k(x_i, x_j)$.

Skoltech
Skolkovo Institute of Science and Technology

Random Fourier Features (RFF)

Theorem (Bochner)

A continuous kernel $k(x, x') = k(x - x')$ on $\mathbb{R}^d$ is positive definite if and only if $k(\delta)$ is a Fourier transform of a non-negative measure

$$k(x, x') = \int_{\Omega} p(w) e^{j w^{\top}(x - x')} dw$$

Skoltech
Skolkovo Institute of Science and Technology

RFF idea

Idea: use Monte-Carlo to approximate the integral[1]

$$k_{RFF}(x, x') = \frac{1}{D}\sum_{i=1}^{D} \cos(w_i^\top (x - x')) = \psi_{\mathbf{W}}(x)^T \psi_{\mathbf{W}}(x'),$$

where

- $w_i \sim p(w)$
- $\mathbf{W} = (w_1^\top, \dots, w_D^\top)^\top$
- $\psi_{\mathbf{w}}(x) = 1/\sqrt{D}(\cos(w_1^\top x), \sin(w_1^\top x), \dots, \cos(w_D^\top x), \sin(w_D^\top x))^\top$

$$\widehat{\mathbf{K}} = \Psi\Psi^T, \quad \Psi = \|\psi_{\mathbf{w}}(x_i)^\top\|_{i=1}^n$$

$\rightarrow$ Go back to linear model with $\psi_{\mathbf{W}}(x)$ features

[1] Rahimi, A., Recht, B. (2008). Random features for large-scale kernel machines.

Related works

- Quasi Monte-Carlo (QMC)[2]
- Gaussian Quadrature [3]
- Orthogonal Random Features (ORF)[4]
- Random Orthogonal Matrices (ROM)[5]
- Ridge Leverage Score based Features [6]

[2] Yang, et al. (2016). Quasi-Monte Carlo feature maps for shift-invariant kernels.
[3] Dao et al. (2017). Gaussian quadrature for kernel features.
[4] Felix, X. Yu, et al. (2016). Orthogonal random features.
[5] Choromanski, et al. (2016). Recycling randomness with structure for sublinear time kernel expansions.
[6] Avron, et al. (2017). Random Fourier features for kernel ridge regression: A pproximation bounds and statistical guarantees.

Integral representation of kernel function

$$k(x, x') = \int_{\Omega} \psi(w, x)\psi(w, x')p(w)dw = \int_{\Omega} f_{xx'}(w)p(w)dw,$$

where $p(w)$ is $\mathcal{N}(0, \sigma_p^2 \mathbf{I})$ – density associated with the kernel and $\psi(\cdot, x)$ is a feature map.

Quadrature rules

1. Change variables to spherical-radial coordinates $(w = r\mathbf{z}, \mathbf{z}^\top \mathbf{z} = 1, w^\top w = r)$

$$k(x, x') = \frac{(2\pi)^{-\frac{d}{2}}}{2} \int\limits_{U_d} \int\limits_{-\infty}^{\infty} e^{-\frac{r^2}{2}} |r|^{d-1} f_{xx'}(r\mathbf{z}) dr d\mathbf{z},$$

   where U_d is a d-dimensional unit sphere.
2. Use stochastic radial-spherical rules.

Skoltech
Skolkovo Institute of Science and Technology

Stochastic spherical-radial rules

- **Spherical-radial rules**[6] of degree (n, p)

$$SR_{\mathbf{Q},\rho}^{(n,p)}(f) = \sum_{j=1}^{p} \widetilde{w}_j \sum_{i=1}^{n} \frac{w_i}{2} \left(f(-\rho_i \mathbf{Q}\mathbf{z}_j) + f(\rho_i \mathbf{Q}\mathbf{z}_j) \right)$$

- Let $p = 2n + 1$. Then the weights are chosen such that the rule is exact for polynomials of degree $2n + 1$.

[6] Genz, A., & Monahan, J. (1998). Stochastic integration rules for infinite regions.

Examples

- Degree 1 rule

$$SR^{(1,1)}_{\mathbf{Q},\rho}(f) = \frac{f_{xx'}(\rho\mathbf{Q}\mathbf{z}) + f_{xx'}(-\rho\mathbf{Q}\mathbf{z})}{2},$$

where
  - $\rho \sim \chi(d)$, $\mathbf{Q}$ – random orthogonal matrix,
  - $\mathbf{z}$ – point on unit sphere.

$\rightarrow$ Classical Random Fourier Features

- Degree $(1,3)$ rule

$$SR_{\mathbf{Q},\rho}(f)^{(1,3)} = \sum_{i=1}^{d} \frac{f_{xx'}(-\rho\mathbf{Q}\mathbf{e}_i) + fxx'(\rho\mathbf{Q}\mathbf{e}_i)}{2d},$$

where $\mathbf{e}_i = (0,\ldots,0,1,0,\ldots,0)^\top$ with 1 in the i-th position.

$\rightarrow$ Orthogonal Random Features!

Degree $(3,3)$ rule

Degree $(3,3)$ rule

$$k_{QBF}(x,x') = \left(1 - \frac{1}{d+1}\sum_{j=1}^{d+1}\frac{d}{\rho_j^2}\right) f_{xx'}(\mathbf{0}) + \\ + \frac{d}{d+1}\sum_{j=1}^{d+1}\left[\frac{f_{xx'}(-\rho_j \mathbf{Q}\mathbf{v}_j) + f_{xx'}(\rho_j \mathbf{Q}\mathbf{v}_j)}{2\rho_j^2}\right],$$

where

- $\rho_j \sim \chi(d+2)$
- $\mathbf{v}_j$ is the j'th vertex of unit d-simplex $\mathbf{V}$
- $\mathbf{Q}$ is a random $d \times d$ orthogonal matrix.

Use structured $\mathbf{Q}$ matrix to speed up feature generation

Empirical results

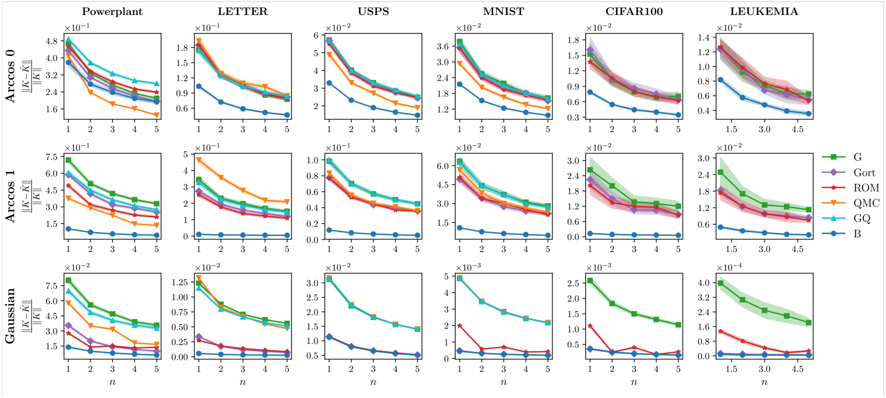


Figure: Error of kernel approximation on different datasets

Skoltech
Skolkovo Institute of Science and Technology

Empirical results

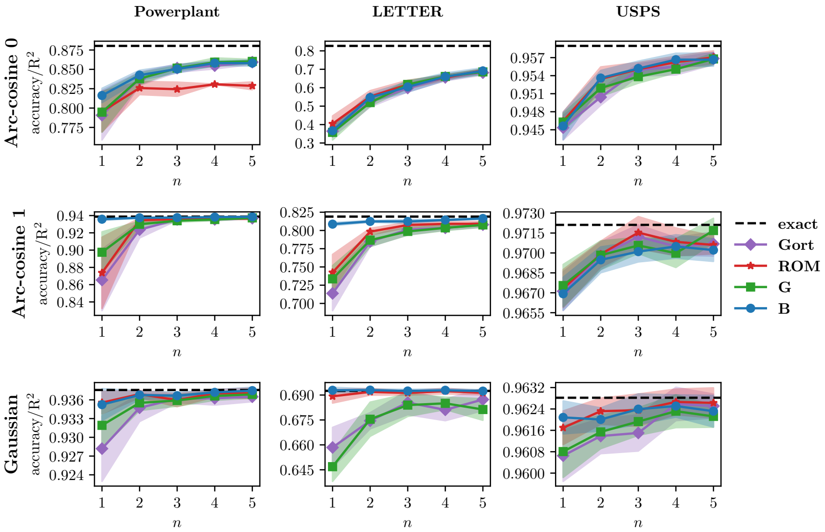


Figure: Error of regression/classification on different datasets

Skoltech
Skolkovo Institute of Science and Technology

Score Matching

- Given a data set $\{\mathbf{x}_i\}_{i=1}^n,\ \mathbf{x}_i \sim p_0(\mathbf{x})$, estimate unknown density $p_0(\mathbf{x})$.
- Find $p_\theta(\mathbf{x}) \in \mathcal{P}$ that minimizes Fisher divergence

$$J(p_0 \| p_\theta) = \frac{1}{2} \int p_0(\mathbf{x}) \|\nabla \log p_\theta(\mathbf{x}) - \nabla \log p_0(\mathbf{x})\|_2^2 d\mathbf{x}$$

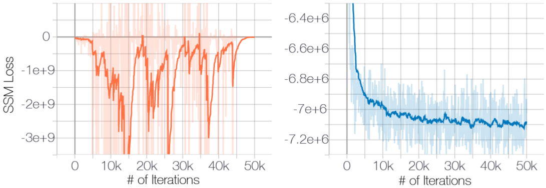
- Equivalent **score matching** objective

$$J_{SM}(p_0 \| p_\theta) = \mathbb{E}_{p_0} \left[\Delta \log p_\theta(\mathbf{x}) + \frac{1}{2} \|\nabla \log p_\theta(\mathbf{x})\|_2^2 \right]$$

Skoltech
Skolkovo Institute of Science and Technology

Issues

- Need to compute second derivatives
- $\text{supp } p_0 \neq \text{supp } p_\theta$



Song, Y., Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution.

Figure: **Left**: original and PCA reconstructions of images. **Middle**: score matching loss. **Right**: score matching loss with noisy data

Skoltech
Skolkovo Institute of Science and Technology

Kernel Exponential Family and Score Matching

- Consider distributions that satisfy

$$\log p_\theta(\mathbf{x}) = f(\mathbf{x}) + \log q_0(\mathbf{x}), \quad f \in \mathcal{H},$$

  where $\mathcal{H}$ is an RKHS with kernel k, q_0 is some generating density.
- Solution

$$f(\mathbf{x}) = -\frac{\xi}{\lambda} + \sum_{a=1}^{n}\sum_{i=1}^{d} \beta_{(\alpha-1)d+i}\partial_i k(\mathbf{x}_a, \cdot),$$

$$\xi = \frac{1}{n}\sum_{a=1}^{n}\sum_{i=1}^{d}\left(\partial_i k(\mathbf{x}_a, \cdot)\partial_i \log q_0(\mathbf{x}_a) + \partial_i^2 k(\mathbf{x}_a, \cdot)\right),$$

$$\left(\mathbf{G} + n\lambda\mathbf{I}\right)\beta = \frac{1}{\lambda}\mathbf{h},$$

$$\mathbf{G}_{(a-1)d+i,(b-1)d+j} = \partial_i\partial_{j+d}k(\mathbf{x}_a, \mathbf{x}_b),$$

$$\mathbf{h} = \langle \xi, \partial_i k(\mathbf{x_a}, \cdot)\rangle_{\mathcal{H}}$$

Skoltech
Skolkovo Institute of Science and Technology

Denoising score matching

Adding noise to input data is equivalent to convolution of the loss with noise distribution:

$$
\begin{aligned}
&\mathbb{E}_{p_\varepsilon}\mathbb{E}_{p_0}\left[\Delta \log p_\theta(\mathbf{x}+\boldsymbol{\varepsilon}) + \frac{1}{2}\|\nabla \log p_\theta(\mathbf{x}+\boldsymbol{\varepsilon})\|^2\right] \\
&= \mathbb{E}_{p_0}\left[\left(\left(\Delta \log p_\theta(\cdot) + \frac{1}{2}\|\nabla \log p_\theta(\cdot)\|^2\right) * p_\varepsilon\right)(\mathbf{x})\right]
\end{aligned}
$$

Solution derived only for Random Features

$$
\hat{f}(\mathbf{x}) = \frac{1}{\lambda}\boldsymbol{\phi}(\mathbf{x})^\top (\boldsymbol{H} + n\lambda \mathbf{I})^{-1}\boldsymbol{H}\boldsymbol{h} - \frac{1}{\lambda}\boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{h},
$$

where

$$
\boldsymbol{H} = \int p_\varepsilon(\mathbf{y})\partial\boldsymbol{\Phi}_y^\top \partial\boldsymbol{\Phi}_y d\mathbf{y}, \quad \boldsymbol{h} = \frac{1}{n}(\partial^2\boldsymbol{\Phi}_z * p(\mathbf{z}))^\top \mathbf{1},
$$

and

$$
[\partial\boldsymbol{\Phi}_y]_{(a-1)d+i} = \partial_i \boldsymbol{\phi}^\top(\boldsymbol{W}(\mathbf{x}_a + \mathbf{y}))
$$

Denoising Score Matching with Random Features

- Regularization:

$$\boldsymbol{h}_i \sim e^{-\sigma_\varepsilon^2 \|\mathbf{w}_i\|_2^2},$$

  i.e. small weights for high-frequency components

- Explicit dependence on noise parameters $\rightarrow$ easier to tune them
- We learn $p_0 * p_\varepsilon \rightarrow$ trade-off between
  - stability of convergence
  - closeness to p_0

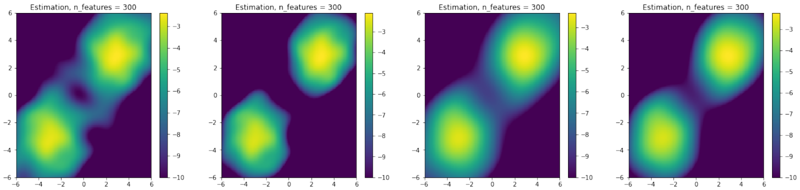
Skoltech
Skolkovo Institute of Science and Technology

Experimental setup

- Adjust kernel parameters on training set using Denoising Score Matching
- Adjust noise variance on hold-out set using Score Matching
- Compare 3 models: proposed approach (**DSM RFF**), score matching with RFF (**SM RFF**), score matching with Nyström approximation (**Nyström**)
- Datasets
  - synthetic 2D data sets: Cosine, Uniform, Banana, Funnel, Rings
  - UCI: RedWine, WhiteWine, MiniBoone
- Metrics:
  - Log-likelihood
  - Wasserstein distance
  - Fisher divergence (for synthetic datasets)

Skoltech
Skolkovo Institute of Science and Technology

Mixtures



(a) SM RFF, narrow kernel

(b) DSM RFF, narrow kernel

(c) SM RFF, wider kernel

(d) DSM RFF, wider kernel

Figure: Mixture of Gaussians

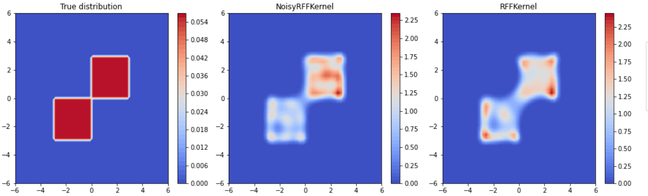


Figure: Mixture of Uniforms

Skoltech
Skolkovo Institute of Science and Technology

Experiments

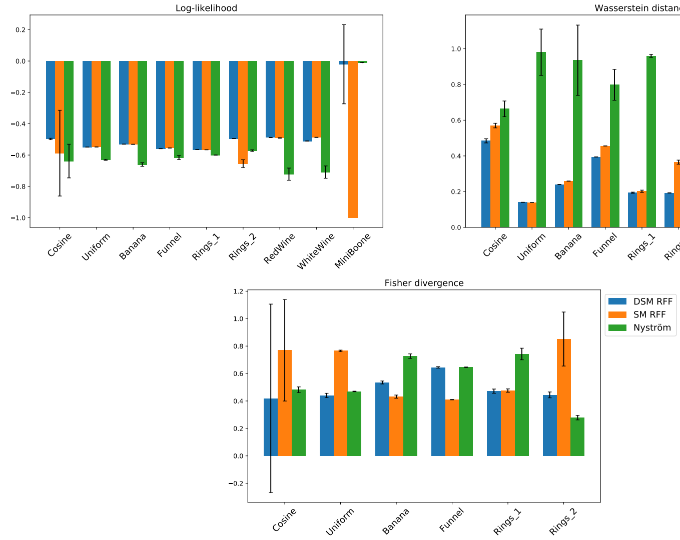


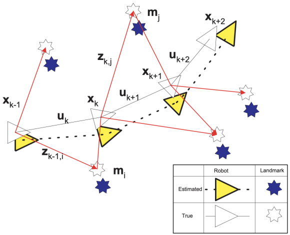
Figure: Comparison on synthetic and UCI data sets

Simultaneous Localization and Mapping (SLAM)

- Consider mobile robot moving in an environment.
- At each time step t_i we obtain measurements $\mathbf{z}_i$ corresponding to some **landmarks** $\boldsymbol{l} = \begin{bmatrix} \boldsymbol{l}_1 \\ \dots \\ \boldsymbol{l}_M \end{bmatrix}$

$$\mathbf{z}_i = \mathbf{h}(\mathbf{x}(t_i), \boldsymbol{l}) + \mathbf{n}_i, \quad \mathbf{n}_i \sim \mathcal{N}(0, \mathbf{R}_i).$$

- Control variables $\mathbf{u}_i$ (maybe missing or given at different timestamps)
- We want to estimate both the robot trajectory $\mathbf{x}(t_1), \dots, \mathbf{x}(t_T)$ and landmarks $\boldsymbol{l}$.
- Time-continuous SLAM: estimate trajectory as a function of time $\mathbf{x}(t)$.



Durrant-Whyte, H., Bailey, T. (2006)

Skoltech
Skolkovo Institute of Science and Technology

GP based SLAM

- Assumptions

$$\boldsymbol{x}(t) \sim \mathcal{GP}(\boldsymbol{\mu}_x, \boldsymbol{k}(t, t'))$$
$$\boldsymbol{l} \sim \mathcal{N}(\boldsymbol{\mu}_l, \mathbf{L})$$

- Let us denote $\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{l} \end{bmatrix}$. We want to maximize the posterior

$$p(\boldsymbol{\theta}|\boldsymbol{z}) \propto p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = -\frac{1}{2}\left(\sum_{i=1}^{T} \|\boldsymbol{z}_i - \boldsymbol{h}(\boldsymbol{\theta}(t_i))\|^2_{\mathbf{R}_i} + \|\boldsymbol{\theta} - \boldsymbol{\mu}\|^2_{\mathbf{P}}\right) = -J$$

RFF SLAM

- Random Features for locations

$$\boldsymbol{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ \alpha(t) \end{bmatrix} = \boldsymbol{\psi}(t)^\top \boldsymbol{b} + \varepsilon, \quad \begin{matrix} \boldsymbol{b} \in \mathbb{R}^{D\times 3}, i = \overline{1,3} \\ \boldsymbol{b} \sim \mathcal{N}(\boldsymbol{\mu}_b, \mathbf{B}) \end{matrix}$$

  where $\boldsymbol{\psi}(t)$ – random features, $\mathbf{B}$ is block-diagonal covariance matrix.

- Gauss-Newton method

$$(\delta\boldsymbol{b}^*, \delta\boldsymbol{l}) = \underset{\delta\boldsymbol{b}, \delta\boldsymbol{l}}{\operatorname{argmin}} \sum_{i=1}^{T} \|\boldsymbol{z}_i - \boldsymbol{h}(\boldsymbol{\psi}(t_i)\bar{\boldsymbol{b}}, \boldsymbol{l}) - \mathbf{H}_i \Psi_i \delta\boldsymbol{b}\|^2_{\mathbf{R}_i} + \|\bar{\boldsymbol{b}} + \delta\boldsymbol{b} - \boldsymbol{\mu}_b\|^2_{\mathbf{B}} + \|\bar{\boldsymbol{l}} + \delta\boldsymbol{l} - \boldsymbol{\mu}_l\|^2_{\mathbf{L}},$$

  where $\bar{\boldsymbol{b}}$ is current solution.

Skoltech
Skolkovo Institute of Science and Technology

Related works

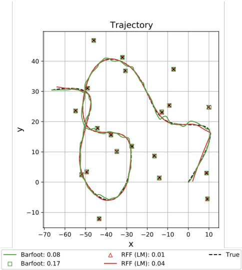
- A lot of works that use splines
- State-space model [7]

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{v}(t) + \mathbf{F}(t)\mathbf{w}(t),$$

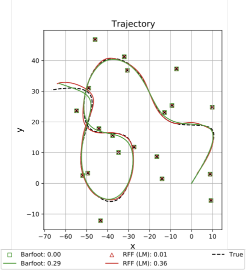
  - $\mathbf{A}(t), \mathbf{F}(t)$ are time-dependent system matrices,
  - $\mathbf{w}(t) \sim \mathcal{GP}(0, \mathbf{Q}_C \delta(t - t'))$
- Solution is GP with block-tridiagonal inverse $\mathbf{K}$ matrix.
- Assumes Markovian-trajectories.
- RBF kernel can better in some cases (like noisy observations)

[7] Barfoot et al. (2014). Batch Continuous-Time Trajectory Estimation as Exactly Sparse Gaussian Process Regression

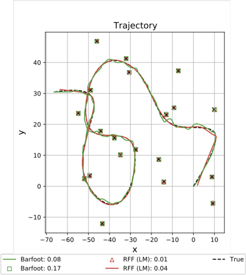
Synthetic trajectories



(a) Bearing only



(b) Range only



(c) Range-Bearing

Skoltech
Skolkovo Institute of Science and Technology

Synthetic trajectories

		Pos.	Rot.	Landmarks
RangeBearing	RFF	**0.033**	0.0018	**0.019**
	Barfoot	0.096	0.0013	0.191
Range	RFF	0.309	0.0197	0.004
	Barfoot	**0.208**	0.0114	0.001
Bearing	RFF	**0.036**	0.0016	**0.018**
	Barfoot	0.096	0.0013	0.191

Table: Relative Pose Errors

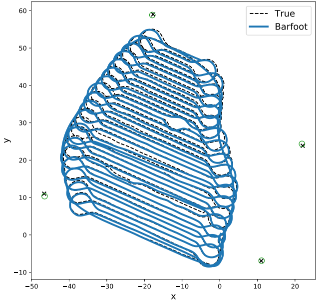
Relative position errors don't take into account drift:

$$\mathrm{RPE} = \sum_{i=1}^{T} \|\delta\hat{\mathbf{x}}_i \ominus \delta\mathbf{x}_i\|$$

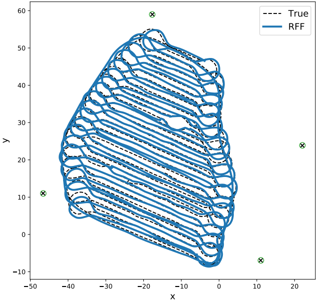
Skoltech
Skolkovo Institute of Science and Technology

Autonomous Lawn-Mower

Range only data set



(a) State-Space Model



(b) Random Features based

Skoltech
Skolkovo Institute of Science and Technology

Summary

Random Features

- proposed Quadrature-based Features
- accurate kernel approximation
- in downstream tasks benefit is smaller

Score Matching

- exact solution for Denoising Score Matching with RFF was proposed
- Natural regularization
- Faster than N yström-type approximation

SLAM

- Random Features give dense covariance $\rightarrow$ better accuracy in case of noisy data
- Random Features can oscillate

Skoltech
Skolkovo Institute of Science and Technology