Linear Multiple Low-Rank Kernel Based Stationary Gaussian Processes Regression for Time Series[1]

Richard Cornelius Suwandi & Juntao Wang

The Chinese University of Hong Kong, Shenzhen

November 22, 2021

[1] F. Yin, L. Pan, T. Chen, *et al.*, "Linear multiple low-rank kernel based stationary gaussian processes regression for time series," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5260–5275, 2020.

Table of Contents

Table of Contents

Motivation

- For the past decades, Gaussian process (GP) has been extensively studied and used in a number of diverse applications
- However, kernel design for GP and the associated hyper-parameter optimization are still difficult
- Traditionally, kernel design relies heavily on human intervention and thus often done subjectively

Related Works

- Multiple kernel learning: Learning a combination of primitive kernels and let data determine the best kernel configuration [Gönen, Mehmet and Alpaydın, 2011]
- Structure discovery: Search for the optimal combination of kernels over a space of kernel structures [Duvenaud *et al.*, 2013]
- Spectral kernel learning: Approximate the spectral density with a Gaussian mixture model [Wilson and Adams, 2013]

Main Contributions

- Propose a novel grid spectral mixture (GSM) kernel for time series modeling
- Redesign the SM kernel with convenient structures that can be exploited by some advanced optimization methods
- Derive two effective numerical methods for tuning the GP hyper-parameters

Table of Contents

GP Regression

- We consider the following GP regression model

$$y = f(\mathbf{x}) + e, \tag{1}$$

where

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_h))$$
$$e \sim \mathcal{N}(0, \sigma_e^2)$$

- The set of all unknown hyper-parameters is denoted by $\boldsymbol{\theta} \triangleq [\boldsymbol{\theta}_h^T, \sigma_e^2]^T$ and the dimension of $\boldsymbol{\theta}$ is assumed to be p

Spectral Mixture (SM) Kernel

- The SM kernel approximates the spectral density $S(f)$ of the underlying stationary kernel by a Gaussian mixture
- Taking the inverse Fourier transform of $S(f)$ yields a stationary kernel in the time-domain as

$$k_{SM}(\tau; \boldsymbol{\theta}_h) = \sum_{q=1}^{Q} \alpha_q \exp[-2\pi^2\tau^2\sigma_q^2] \cos(2\pi\tau\mu_q) \tag{2}$$

where $\boldsymbol{\theta}_h \triangleq [\alpha_1, \ldots, \alpha_Q, \mu_1, \ldots, \mu_Q, \sigma_1^2, \ldots, \sigma_Q^2]^T$ denotes the SM kernel hyper-parameters with Q mixture components

Limitations of SM Kernel

- It is generally difficult to tune the SM kernel hyper-parameters due to the non-convex nature of the optimization problem
- Since the optimization problem has no favorable structure, it may easily get stuck at a bad local optimum
- The number of kernel hyper-parameters to tune is $3Q$ which requires high computational time especially when Q is large

Table of Contents

Grid Spectral Mixture (GSM) Kernel

- To address the limitations of SM kernel, GSM kernel modifies the SM kernel by fixing the frequency and variance parameters using a pre-selected grid of points
- The grid points can be generated using either of the following strategies:
  1. Uniformly spaced grids
  2. Randomly spaced grids

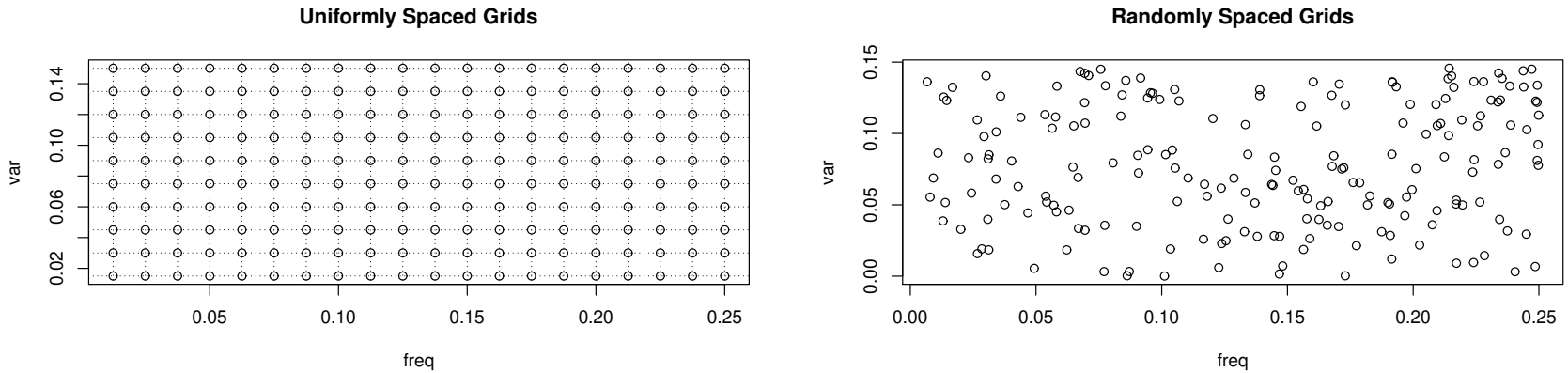

Figure: Illustration of the two strategies for generating grids. In this specific example, μ_{low} is set to be 0, $\mu_{high} = 0.25$, $\sigma_{low} = 0$ and $\sigma_{high} = 0.15$.

Table of Contents

2-D GSM Kernel

- By taking the inverse Fourier transform of the spectral density, the GSM kernel can be formulated as

$$k(\tau;\boldsymbol{\theta}_h) = \sum_{i=1}^{m} \alpha_i \underbrace{\exp\left[-2\pi^2\tau^2\sigma_i^2\right]\cos(2\pi\tau\mu_i)}_{k_i(\tau)} \tag{3}$$

  where $\boldsymbol{\theta}_h = \boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]^T \geq \mathbf{0}$ denotes the GSM kernel hyper-parameters
- Since the grids are generated in the 2-D (μ, σ) space, the resulting kernel is called 2-D GSM kernel

1-D GSM Kernel

- To further reduce the model complexity, we can fix the variance parameters σ_i to a small fixed value σ for $i = 1, \ldots, m$
- We obtain the following formulation for the GSM kernel

$$k(\tau; \boldsymbol{\theta}_h) = \sum_{i=1}^{m} \alpha_i \underbrace{\exp(-2\pi^2\tau^2\sigma^2)\cos(2\pi\tau\mu_i)}_{k_i(\tau)} \tag{4}$$

- The kernel given in Eq.(4) is called 1-D GSM kernel, because the grids are generated in the 1-D μ-space, given a fixed σ

Properties of GSM Kernel

Some properties of the GSM kernel are given as follows:

1. It is a valid kernel
2. For a given data set with n samples, when the variance parameter σ is chosen sufficiently small, then for any frequency parameter $\mu_i \in [0, 1/2)$, each sub-kernel matrix has low rank
3. It is smooth with closed-form derivatives of all orders
4. In contrast to most of the classic kernels, the sub-kernel sometimes demonstrates negative correlation between two data points
5. For big data set with size $n \gg \frac{4}{\pi\sigma}$, the sub-kernel matrix is sparse and close to a band matrix, which enables more efficient utilization of computer memory

Table of Contents

GSM Kernel Matrix Approximations

- When the number of sub-kernels m and the data size n is large, unaffordable memory is required to store the GSM sub-kernel matrices
- To reduce the computational complexity and memory usage, two types of kernel matrix approximations are adopted:
  1. Nyström approximation[2]
  2. Random Fourier feature approximation[3]

Note:

- In practice, a factor $\mathbf{L}_i$ satisfying $\mathbf{K}_i = \mathbf{L}_i \mathbf{L}_i^T$ is often stored instead of the sub-kernel matrices $\mathbf{K}_i$, especially when $\mathbf{K}_i$ has low rank

[2] C. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," English, in *Advances in Neural Information Processing Systems 13 (NIPS 2000)*, T. Leen, T. Dietterich, and V. Tresp, Eds., MIT Press, 2001, pp. 682–688.

[3] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," , ser. NIPS'07, Vancouver, British Columbia, Canada: Curran Associates Inc., 2007, pp. 1177–1184.

Table of Contents

GSM Hyper-parameter Optimization

- For the GSM kernel, the hyper-parameter optimization problem can be formulated as

$$\boldsymbol{\theta}_{ML} = \arg\min_{\boldsymbol{\theta} \geq \mathbf{0}} \boldsymbol{y}^T \boldsymbol{C}^{-1}(\boldsymbol{\theta}) \boldsymbol{y} + \log\det \boldsymbol{C}(\boldsymbol{\theta}) \tag{5}$$

  where $\boldsymbol{\theta} = [\boldsymbol{\alpha}^T, \sigma_e^2]^T$ and $\boldsymbol{C}(\boldsymbol{\theta}) \triangleq \sum_{i=1}^m \alpha_i \boldsymbol{K}_i + \sigma_e^2 \boldsymbol{I}_n$

- The above optimization problem belongs to the well-known difference-of-convex program (DCP)

Majorization-Minimization (MM) Method

- The basic idea of the MM method is to solve a difficult problem by solving a sequence of smaller problems:

$$\boldsymbol{\theta}^{k+1} = \arg\min_{\boldsymbol{\theta}\in\Theta} \bar{l}(\boldsymbol{\theta}, \boldsymbol{\theta}^k) \tag{6}$$

where $\bar{l}(\boldsymbol{\theta}, \boldsymbol{\theta}^k)$ is of the majorization function of $l(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^k$ satisfying:

1. $\bar{l}(\boldsymbol{\theta}, \boldsymbol{\theta}) = l(\boldsymbol{\theta})$ for $\boldsymbol{\theta} \in \Theta$
2. $l(\boldsymbol{\theta}) \leq \bar{l}(\boldsymbol{\theta}, \boldsymbol{\theta}')$ for $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \theta$

- For our problem, we let

$$l(\boldsymbol{\theta}) \triangleq \underbrace{\boldsymbol{y}^T \boldsymbol{C}^{-1}(\boldsymbol{\theta})\boldsymbol{y}}_{g(\boldsymbol{\theta})} - \underbrace{[-\log\det \boldsymbol{C}(\boldsymbol{\theta})]}_{h(\boldsymbol{\theta})} \tag{7}$$

  where $g(\boldsymbol{\theta})$ and $h(\boldsymbol{\theta})$ are both convex and differentiable functions
- We use the linear majorization by performing the first-order Taylor expansion of $h(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^k$ and obtain

$$\bar{l}(\boldsymbol{\theta}, \boldsymbol{\theta}^k) = g(\boldsymbol{\theta}) - h(\boldsymbol{\theta}^k) - \nabla_{\boldsymbol{\theta}}^T h(\boldsymbol{\theta}^k)(\boldsymbol{\theta} - \boldsymbol{\theta}^k) \tag{8}$$

- Hence, minimizing Eq.(6) at each iteration becomes a convex optimization problem

Nonlinearly Constrained ADMM

- The main idea of this method is to reformulate the original problem as

$$\arg\min_{\boldsymbol{S},\boldsymbol{\alpha}} \boldsymbol{y}^T \boldsymbol{S}\boldsymbol{y} - \log\det(\boldsymbol{S}) \tag{9}$$

subject to $\boldsymbol{S}\left(\sum_i^m \alpha_i \boldsymbol{K}_i + \sigma_e^2 \boldsymbol{I}_n\right) = \boldsymbol{I}_n$ and $\boldsymbol{\alpha} \geq \boldsymbol{0}$, where $\boldsymbol{S} \in \mathbb{R}^{n\times n}$

- Then, we can write the augmented Lagrangian function as

$$\begin{aligned} L_\rho(\boldsymbol{S},\boldsymbol{\alpha},\boldsymbol{\Lambda}) = {} & \boldsymbol{y}^T \boldsymbol{S}\boldsymbol{y} - \log\det(\boldsymbol{S}) \\ & + \left\langle \boldsymbol{\Lambda}, \boldsymbol{S}\left(\sum_i^m \alpha_i \boldsymbol{K}_i + \sigma_e^2 \boldsymbol{I}_n\right) - \boldsymbol{I}_n \right\rangle \\ & + \frac{\rho}{2}\left\| \boldsymbol{S}\left(\sum_i^m \alpha_i \boldsymbol{K}_i + \sigma_e^2 \boldsymbol{I}_n\right) - \boldsymbol{I}_n \right\|_F^2 \end{aligned} \tag{10}$$

where $\rho > 0$ is the regularization parameter

- The ADMM iteratively decomposes Eq. (10) into the following sub-problems:

$$\boldsymbol{S}^{k+1} = \arg\min_{\boldsymbol{S}} L_\rho\left(\boldsymbol{S}, \boldsymbol{\alpha}^k, \boldsymbol{\Lambda}^k\right) \tag{11}$$

$$\alpha_i^{k+1} = \arg\min_{\alpha_i} L_\rho\left(\boldsymbol{S}^{k+1}, \{\alpha_i, \boldsymbol{\alpha}_{-i}^{k,k+1}\}, \boldsymbol{\Lambda}^k\right), i = 1, ..., m \tag{12}$$

$$\boldsymbol{\Lambda}^{k+1} = \boldsymbol{\Lambda}^k + \rho'\left[\boldsymbol{S}^{k+1}\left(\sum_i^m \alpha_i^{k+1}\boldsymbol{K}_i + \sigma_e^2\boldsymbol{I}_n\right) - \boldsymbol{I}_n\right] \tag{13}$$

where $\boldsymbol{\alpha}_{-i}^{k,k+1} \triangleq [\alpha_1^{k+1}, \alpha_2^{k+1}, ..., \alpha_{i-1}^{k+1}, \alpha_{i+1}^{k}, ..., \alpha_m^{k}]^T$

- It can be verified that the sub-problems in Eq. (11) and Eq. (12) are both convex in terms of the optimization variables.
- Compared to the previous methods, this method has potential to find a better local minimum with smaller negative likelihood value and prediction MSE
- However, this method is only suitable for short time series because its sub-problems involve matrix inversion and matrix multiplications which scale as $\mathcal{O}(n^3)$

Table of Contents

Data Sets

Name	Description	Training $\mathcal{D}$	Test $\mathcal{D}_*$
ECG	Electrocardiography of an ordinary person measured over a period of time	680	20
CO2	CO2 concentration made between 1958 and the end of 2003	481	20
Electricity	Monthly average residential electricity usage in Iowa City 1971-1979	86	20
Employment	Wisconsin employment time series, trade, Jan. 1961 – Oct. 1975	158	20
Hotel	Monthly hotel occupied room average 1963-1976	148	20
Passenger	Passenger miles (Mil) flown domestic U.K., Jul. 1962-May 1972	98	20
Clay	Monthly production of clay bricks: million units. Jan 1956 – Aug 1995	450	20
Unemployment	Monthly U.S. female (16-19 years) unemployment figures (thousands) 1948-1981	380	20

Table: Details of the selected data sets.

Algorithm Setup

- GSM kernel based GP (**GSMGP**): Different setups in different experiments[4]
- SM kernel based GP (**SMGP**):
  1. The number of Gaussian mixture components $Q = 10$ or 500
  2. Default setup of the source code[5] (initialize weights $\boldsymbol{\alpha} = \frac{\sigma(data)}{Q}$, $\boldsymbol{\mu} \sim \mathcal{U}(0, f_s/2)$, $\boldsymbol{\sigma} \sim$ truncated Gaussian distribution)
  3. Optimize using nonlinear conjugate gradients (SGD in the source code)
- Sparse Spectrum GP (**SSGP**):
  1. The number of basis $m = 500$
  2. Default setup of the source code[6] (section 4.2 of Lázaro-Gredilla *et al.*[7])
  3. Optimize using a conjugate-gradient method

[4] `https://github.com/Paalis/MATLAB_GSM`

[5] `https://people.orie.cornell.edu/andrew/code/`

[6] `http://www.tsc.uc3m.es/~miguel/downloads.php`

[7] M. Lázaro-Gredilla, J. Quiñnero-Candela, C. E. Rasmussen, *et al.*, "Sparse spectrum gaussian process regression," *Journal of Machine Learning Research*, vol. 11, no. 63, pp. 1865–1881, 2010.

Table of Contents

Performance of The 2-D GSM Kernel with MM Method

Experiment setup:

- 2-D GSMGP:
  1. Randomly generate 20,000 grid points (μ_i, σ_i^2) with $\mu_i \in [0, 0.5]$ and $\sigma_i^2 \in [0, 0.15]$
  2. Initialize weights $\boldsymbol{\alpha}$ to $\mathbf{0}$
  3. Optimize using MM method
- Conduct 30 independent Monte-Carlo (MC) runs; calculated the program fail rate (PFR $= \frac{\#\ of\ \ runs\ \ stuck\ \ at\ \ a\ \ bad\ \ local\ \ minimum}{30}$)
- Compute MSE excluding fail runs

Performance of The 2-D GSM Kernel with MM Method

Name	SSGP MSE	SMGP MSE	SMGP PFR	GSMGP MSE	GSMGP PFR
ECG	1.6E-01	2.1E+00	0.63	NA	NA
CO2	2.0E+02	7.4E+04	0.83	NA	NA
Electricity	8.2E+03	1.8E+04	0.47	6.8E+03	0.2
Employment	7.7E+01	2.3E+04	0.27	3.9E+01	0.07
Hotel	1.9E+04	2.6E+05	0.33	2.4E+03	0
Passenger	6.9E+02	3.5E+03	0.37	1.7E+02	0
Clay	5.3E+02	4.8E+03	0.93	NA	NA
Unemploy	2.1E+04	1.2E+05	0.9	NA	NA

Table: Performance comparison between the proposed GSMGP (with 2-D grids) and its competitors, SSGP and SMGP, in terms of the MSE and the PFR.

Performance of The 2-D GSM Kernel with MM Method

Interpretation of results: 2-D GSMGP has gained well-improved prediction MSE and stability as compared to its competitors

- The performance of GSMGP becomes better and more stable, when the number of the grids grows beyond around 10,000
- ML solutions are sparse (the average number of non-zero α values generated by the ML method is equal to 26, 19, 17, 22, respectively for *Electricity, Employment, Hotel* and *Passenger*)
- However, due to large $dim(\alpha)$ and long data record, GSMGP cannot handle with data sets *ECG, CO2, Clay* and *Unemployment*

Table of Contents

Performance of The 1-D GSM Kernel with MM Method

Experiment setup:

- 1-D GSMGP:
  1. Uniformly generate m frequency parameter $\mu_i \in [0, 0.5)$ and fix $\sigma = 0.001$
  2. Initialize weights $\alpha_i \sim \mathcal{N}(\mu_\alpha = 0, \sigma_\alpha^2 = 10)$ and then $\alpha_i = max(\alpha_i, 0)$
  3. Optimize using MM method
- Conduct 100 independent MC runs; calculate the PFR
- Compute MSE excluding fail runs

Performance of The 1-D GSM Kernel with MM Method

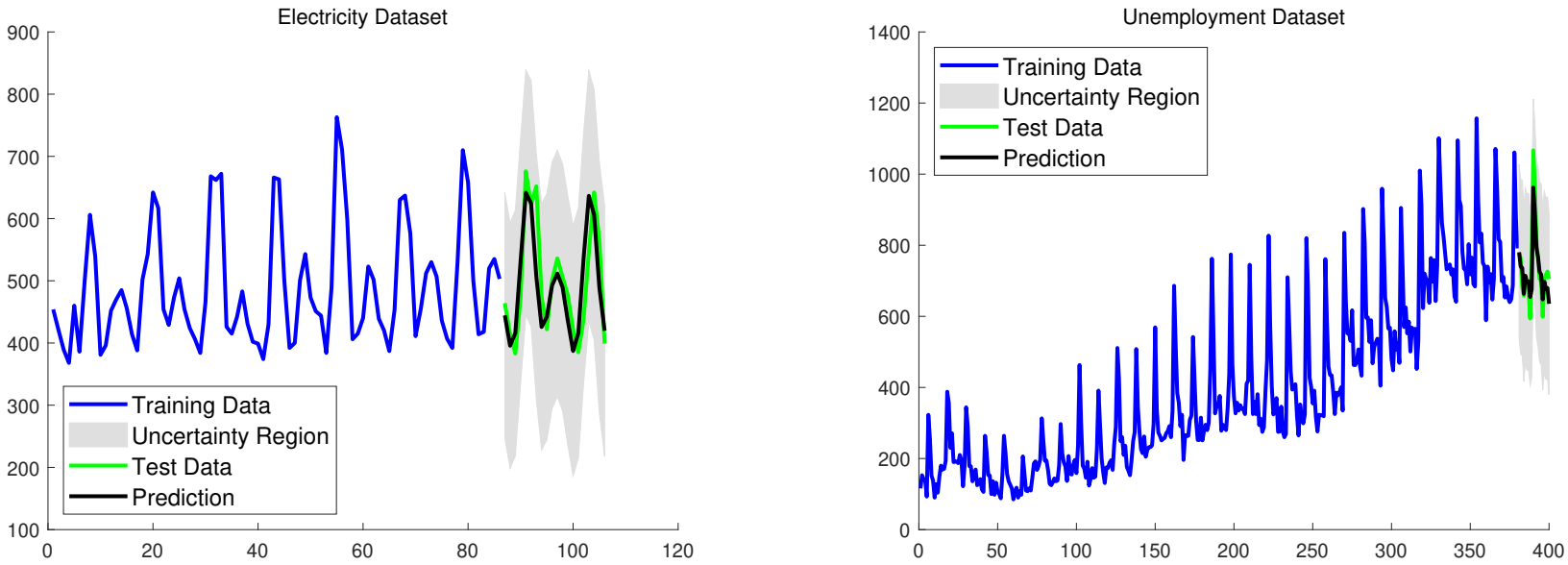

Figure: Training and test performance of the GSMGP using 1-D GSM kernel with $\sigma = 0.001$ and $m = 500$ uniformly generated grids.

Note: Similar results for other data sets can be found in *the page 10 of the supplement file*[8].

[8] https://ieeexplore.ieee.org/ielx7/78/8933520/9189863/supp1-3023008.pdf?arnumber=9189863

Performance of The 1-D GSM Kernel with MM Method

Name	1-D MSE	1-D Iterations	1-D PFR	2-D MSE	2-D Iterations
ECG	1.3E-02	24	0.01	NA	NA
CO2	1.5E+00	10	0.17	NA	NA
Electricity	4.7E+03	2	0.07	6.8E+03	2
Employment	1.1E+02	23	0.06	3.9E+01	1
Hotel	8.9E+02	14	0.02	2.4E+03	6
Passenger	1.9E+02	28	0.02	1.7E+02	13
Clay	1.9E+02	25	0.12	NA	NA
Unemploy.	3.6E+03	9	0.10	NA	NA

Table: Prediction MSE generated by two GSM kernels (one is using $m = 20000$ 2-D grids vs. the other using $m = 500$ 1-D grids).

Interpretation of results:

- The prediction MSE generated by the 1-D GSMGP degrades slightly as compared to that generated by the 2-D GSMGP in most cases
  1. 2-D GSMGP better covers the parameter space
  2. 2-D GSMGP may overfit the training data in some cases
- Due to reduced complexity, 1-D GSMGP can handle much longer time series than 2-D GSMGP

Performance of The 1-D GSM Kernel with MM Method

Name	GSMGP MSE	GSMGP CT (s)	GSMGP PFR	SMGP MSE	SMGP CT (s)	SMGP PFR
ECG	1.3E-02	140.4	0.01	1.9E-02	3.4E+03	0.3
CO2	1.5E+00	69.3	0.17	1.1E+00	2.0E+03	0.07
Electricity	4.7E+03	1.46	0.07	7.5E+03	1.0E+02	0
Employment	1.1E+02	31.2	0.06	0.7E+02	2.5E+02	0.03
Hotel	8.9E+02	17.5	0.02	2.8E+03	2.8E+02	0.97
Passenger	1.9E+02	14.7	0.02	1.6E+02	1.1E+02	0.23
Clay	1.9E+02	140.4	0.12	3.3E+02	3.4E+03	0
Unemploy.	3.6E+03	42.3	0.10	1.4E+04	1.4E+03	0.57

Table: Prediction MSE of the GSMGP with $m = 500$ 1-D grids vs. SMGP with $Q = 500$ Gaussian modes

Conclusion: The 1-D GSMGP has achieved overall better prediction results with much less computational time and higher stability as compared to the original SM kernel.

Table of Contents

Performance of The 1-D GSM Kernel with ADMM

Experiment setup:

- Conduct experiments only on small data sets (*Electricity* and *Unemployment*) due to the $\mathcal{O}(n^3)$ complexity of nonlinearly constrained ADMM
- ADMM:
  1. To update $\boldsymbol{S}$, we let $It_{\boldsymbol{S}} = 1000, \epsilon_{\boldsymbol{S}} = 10^{-15}, \delta = 1$
  2. For selecting the step size in light of the Armijo rule, we let $s = 10^{-4}, \beta = 1/5, h = 10^{-5}$
  3. The remainders are $\rho = 100, \rho' = \rho/2 = 50, \epsilon_{ADMM} = 10^{-3}$
  4. For the initial guess, we let $\boldsymbol{\Lambda}^{(0)} = \boldsymbol{I}$
- 1-D GSM-ADMM:
  1. Uniformly generate m frequency parameter $\mu_i \in [0, 0.5)$ and fix $\sigma = 0.001$
  2. For the *Electricity* data set, $\boldsymbol{\alpha}^{(0)}$ is obtained by fitting the nonparametric Welch periodogram; while for the *Unemployment* data set, $\boldsymbol{\alpha}^{(0)}$ is obtained by running just one iteration of the MM method

Performance of The 1-D GSM Kernel with ADMM

Performance Metric	Electricity	Unemployment
GSM-GD Objective	8.330E+02	3.838E+03
GSM-MM Objective	8.284E+02	3.779E+03
GSM-ADMM Objective	8.266E+02	3.776E+03
GSM-GD MSE	4.426E+03	1.481E+04
GSM-MM MSE	3.037E+03	2.248E+03
GSM-ADMM MSE	2.220E+03	2.222E+03
GSM-GD CT (s)	2272s	79189s
GSM-MM CT (s)	0.93s	8.40s
GSM-ADMM CT (s)	6351.17s	160367.25s

Table: Performance of three numerical optimization methods in terms of the objective function value, the prediction MSE, and the computational time

Note: The maximum number of iterations of the ADMM is restricted due to its slow convergence rate at the second half iterations.

Performance of The 1-D GSM Kernel with ADMM

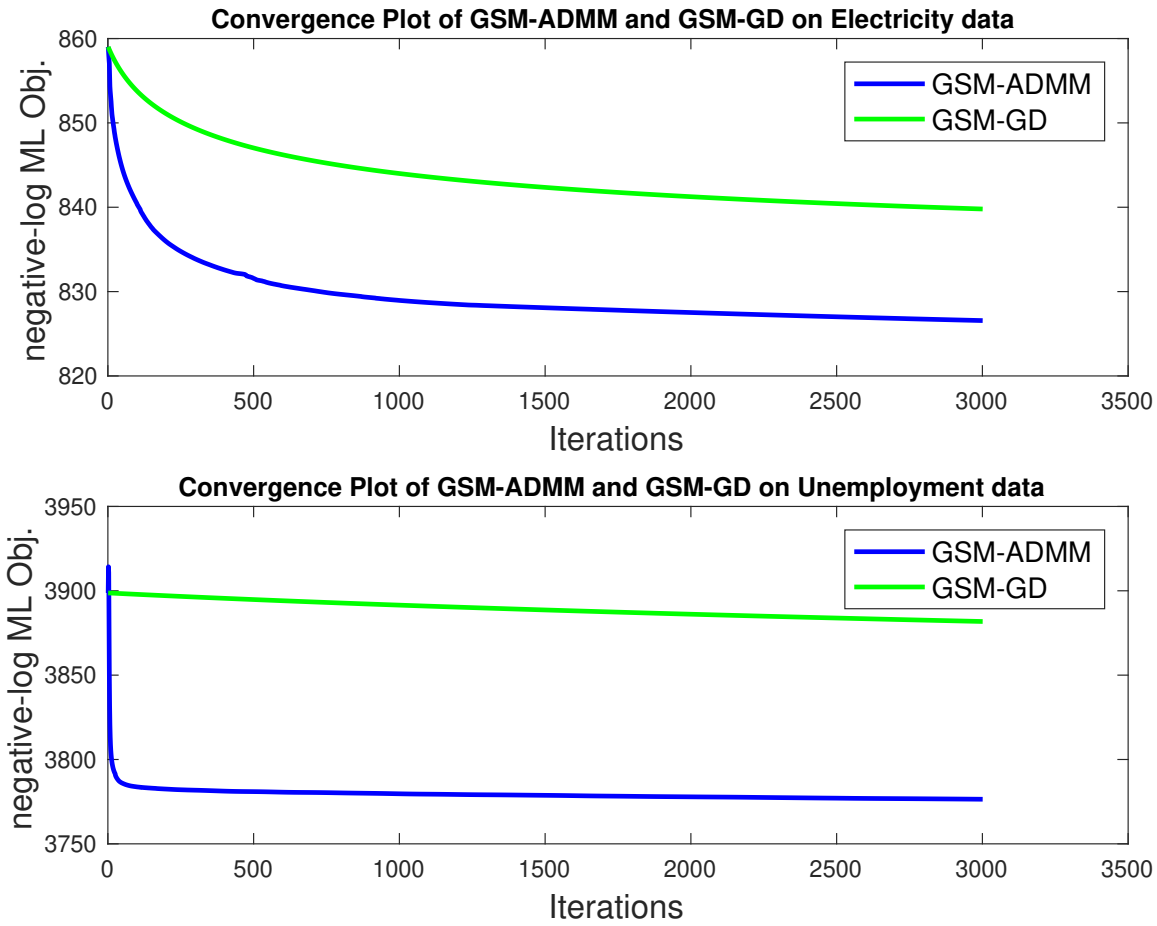

Figure: Negative log-likelihood versus iterations of the proposed ADMM as compared to the classic gradient projection.

Performance of The 1-D GSM Kernel with ADMM

Interpretation of results:

- Although the ADMM has not converged yet, it already reached the smallest objective function value and prediction MSE
- However, the ADMM is less favorable than the MM method in terms of the computational time
- The GSM-ADMM shows faster convergence rate as compared to GSM-GD

Table of Contents

Main Contents of This Paper

- One idea:
  - Represent parameters by **pre-selected grid of points**
- Two forms:
  - **2-D GSM kernel**
  - **1-D GSM kernel**
- Two approximations:
  - **Nyström approximation**
  - **Random Fourier Feature approximation**
- Two methods:
  - **Sequential majorization-minimization(MM) method**
  - **Nonlinearly constrained alternating direction method of multipliers(ADMM)**

Pros and Cons

Pros:

- GSM kernel:
  - Let the data choose the most appropriate kernels
  - Low-rank property of sub-kernels
  - Difference-of-convex program
  - Sparse solution
- MM method:
  - Better convergence speed
  - Economical computational time
  - Insensitivity to an initial guess
  - Competitive fitting and prediction performance
- ADMM:
  - Great potential to achieve a better local minimum

Pros and Cons

Cons:

- ADMM:
  - The proposed ADMM has high complexity and costs large computational time on the big data sets
- GSM kernel:
  - More suitable for low-dimensional time series
  - The number of mixtures m has to be very large to cover parameter space. So we usually choose a large number of mixtures for good approximation, but it can be computationally expensive [?]

Table of Contents

Ideas

- We propose a straightforward extension of the GSM kernel to handle multi-dimensional time series:

$$k(\tau) = \sum_{i=1}^{m} \alpha_i \underbrace{\prod_{j=1}^{d} \exp\left\{-2\pi^2 \tau_j{}^2 v_i^{(j)}\right\} \cos\left(2\pi \tau_j \mu_i^{(j)}\right)}_{k_i(\tau)} \tag{14}$$

- We fix the variance parameters $v_i^{(j)} = \sigma$, for all i and j
- For each dimension $j = 1, \ldots, d$, we sample $[\mu_1^{(j)}, \mu_2^{(j)}, \ldots, \mu_m^{(j)}]$ using the grid generation strategy

Simulation: Data Set

We consider the 2-D case and simulate the data set from a GP with SM kernel using the following setup:

1. The number of mixture components Q is set to 3
2. The weights $\boldsymbol{\alpha}$ is initialized as $[30, 10, 5]$
3. The ground-truth frequency is pre-selected as:

$$\boldsymbol{\mu} = \begin{bmatrix} 0.1 & 0.1 \\ 0.3 & 0.3 \\ 0.5 & 0.5 \end{bmatrix}$$

4. The bandwidth is fixed to a small number, $\sigma = 0.001$

Simulation: Data Set

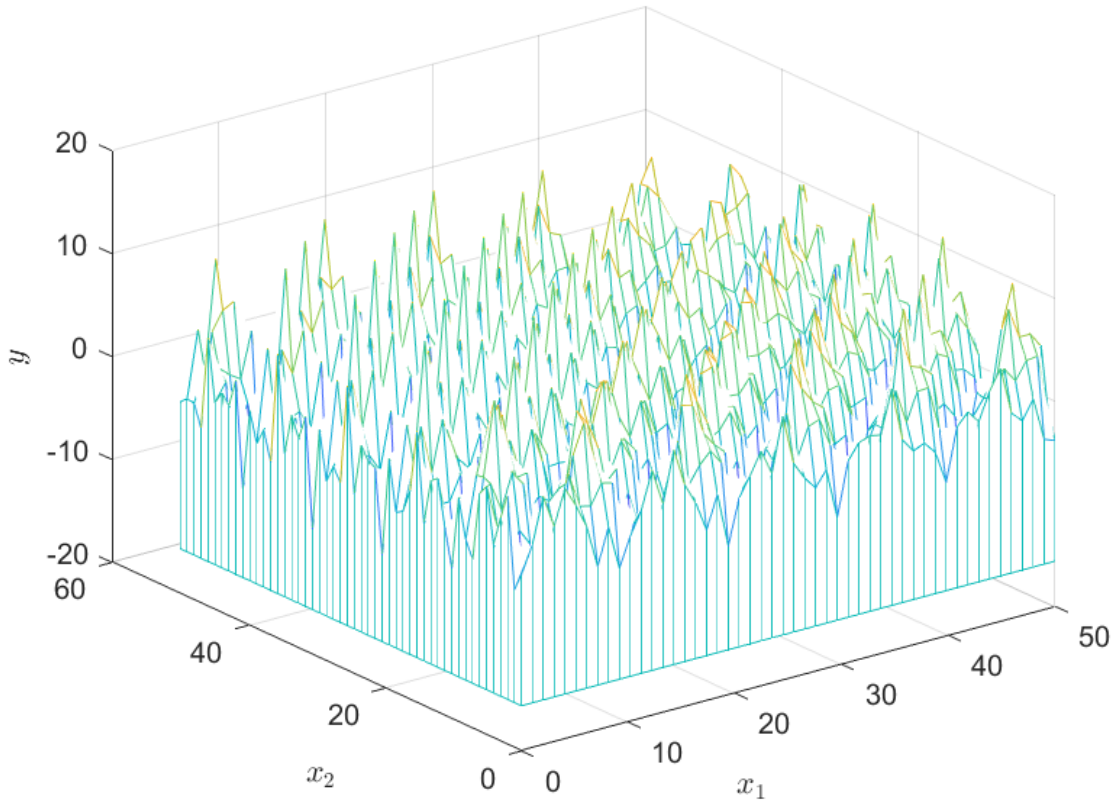

Figure: The simulated data set from a zero-mean Gaussian Process with SM kernel evaluated at $x_1 \in [0, 50]$ and $x_2 \in [0, 50]$ with 50 points each.

Simulation: Algorithm Setup

For the simulation, the algorithm setup is as follows:

1. The grid points are uniformly selected
2. The number of grid points is set to 500
3. The bandwidth σ is fixed to 0.001
4. The weights $\boldsymbol{\alpha}$ is initialized to $\mathbf{0}$
5. The noise variance parameter σ_e^2 is estimated using the cross-validation filter type method[9]
6. The MM method is used in the optimization process

[9] D. Garcia, "Robust smoothing of gridded data in one and higher dimensions with missing values," *Computational Statistics & Data Analysis*, vol. 54, no. 4, pp. 1167–1178, Apr. 2010.

Simulation: Results

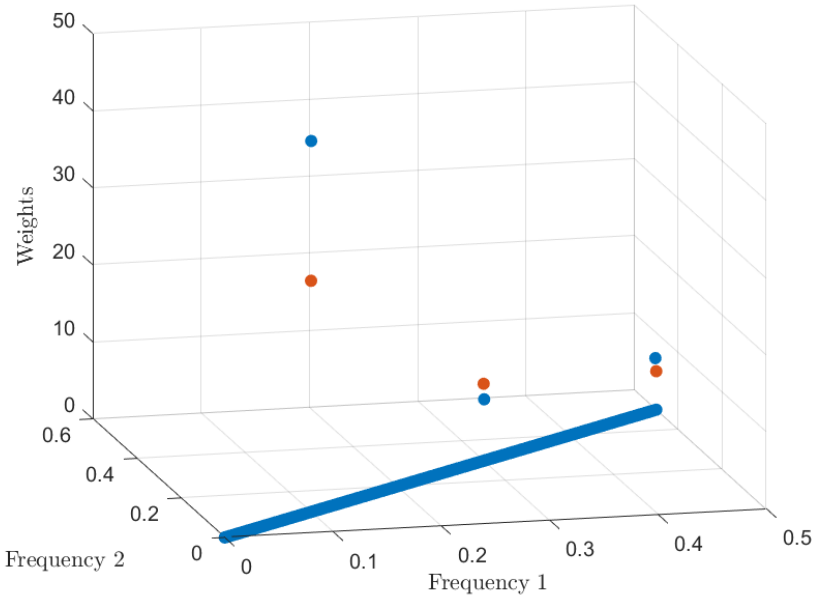

Figure: Estimated weights and frequency optimized using the MM method (blue points) compared with the ground truth (orange points)

Key observation: The solutions are sparse and only have significant non-zero weights on the ground truth frequencies

References I

[1] F. Yin, L. Pan, T. Chen, S. Theodoridis, Z.-Q. T. Luo, and A. M. Zoubir, "Linear multiple low-rank kernel based stationary gaussian processes regression for time series," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5260–5275, 2020.

[2] C. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," English, in *Advances in Neural Information Processing Systems 13 (NIPS 2000)*, T. Leen, T. Dietterich, and V. Tresp, Eds., MIT Press, 2001, pp. 682–688.

[3] A. Rahimi and B. Recht, "Random features for large-scale kernel machines", ser. NIPS'07, Vancouver, British Columbia, Canada: Curran Associates Inc., 2007, pp. 1177–1184.

References II

[4] M. Lázaro-Gredilla, J. Quiñnero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal, "Sparse spectrum gaussian process regression," *Journal of Machine Learning Research*, vol. 11, no. 63, pp. 1865–1881, 2010.

[5] D. Garcia, "Robust smoothing of gridded data in one and higher dimensions with missing values," *Computational Statistics & Data Analysis*, vol. 54, no. 4, pp. 1167–1178, Apr. 2010.

End

Q & A

Thanks for listening, any questions?

Questions

Actually, we want to ask the author, i.e. *Prof. Feng Yin*, some questions about this paper:

1. For the GSM kernel, is it possible to choose a suitable m automatically?
2. For 1-D GSM kernel, what about fixing μ and generating σ?
3. Question about experiments: What is the criterion of PFR? How can we judge that one run gets stuck at a bad local minimum?
4. Question about experiments: Is it a fair comparison?
5. Question about experiments: The graph of experiment results of random Fourier feature approximation (page 64)

Table of Contents

Nyström Approximation

- When using the Nyström approximation, the memory usage for storing $\tilde{\mathbf{L}}_i$ can be reduced to $\tilde{p}/n \times 100\%$ of the original amount for storing for storing $\mathbf{L}_i$
- The computational complexity for performing the eigendecomposition is also reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(p^3)$

Note:

- $\tilde{p}$ denotes the effective number of eigenvalues of the corresponding kernel matrix
- p denotes the size of the smaller subset of the training data
- We have $\tilde{p} \leq p \leq n$

Random Fourier Feature Approximation

- When using the Random Fourier feature approximation, the memory usage for storing $\tilde{\mathbf{L}}_i$ can be reduced to $2R/n \times 100\%$ of the original amount for storing for storing $\mathbf{L}_i$
- The computational complexity mainly comes from sampling a univariate Gaussian distribution which remains low

Note:

- $2R$ denotes the effective number of random frequencies/features

Benefits of Nyström Approximation

Experiment setup:

- 1-D GSMGP:
  1. Uniformly generate m frequency parameter $\mu_i \in [0, 0.5)$ and fix $\sigma = 0.001$; Initialize weights $\alpha_i \sim \mathcal{N}(\mu_\alpha = 0, \sigma_\alpha^2 = 10)$ and then $\alpha_i = max(\alpha_i, 0)$
  2. Optimize using MM method
- Randomly sample only 5% of the complete training inputs for constructing a Nyström approximation of every sub-kernel matrix $\boldsymbol{K}_i, i = 1, 2, ..., m$

Benefits of Nyström Approximation

Name	GSM MSE	GSM CT	NY-GSM MSE	NY-GSM CT
ECG	1.3E-02	122s	1.3E-02	116s
CO2	9.3E-01	24s	9.3E-01	22s
Electricity	3.0E+03	0.9s	3.0E+03	0.2s
Employment	6.8E+01	12s	6.8E+01	5s
Hotel	4.3E+02	3s	4.3E+02	1s
Passenger	2.4E+02	8s	2.9E+02	3s
Clay	8.5E+01	60s	8.5E+01	50s
Unemploy.	2.3E+03	8s	2.3E+03	3s

Table: Prediction MSE generated by the 1-D GSM kernel versus its Nyström approximation, short as NY-GSM.

Note: The total computation time is not reduced much due to the low-ranked sub-kernel matrices (refer to the property 2 of page 16)

Benefits of Nyström Approximation

Name	max rank GSM sub-kernels	min rank sub-kernels	mean rank sub-kernels
ECG	34	17	33
CO2	27	13	25
Electricity	14	7	13
Employment	16	8	15
Hotel	14	7	13
Passenger	14	7	13
Clay	26	13	25
Unemployment	24	12	23

Table: Maximum rank, minimum rank, and mean rank of the selected $m = 500$ GSM sub-kernel matrices used in the above experiments.

Note: The computational complexity of the MM method is approximately $\mathcal{O}(mn^{3/2})$ instead of the worst case $\mathcal{O}(mn^3)$

Benefits of Random Fourier Feature Approximation[10]

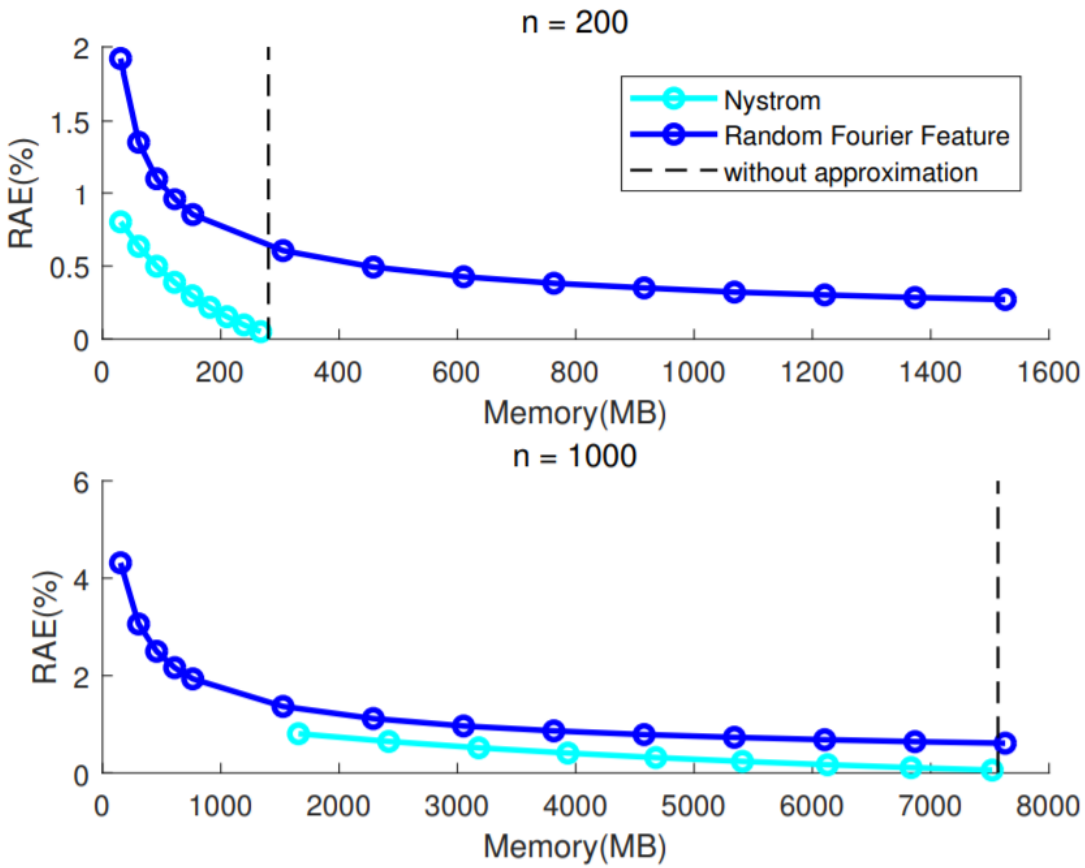

Figure: Relative approximation error (RAE) v.s. storage for Nyström approximation and random Fourier feature (RFF) approximation

[10] Relevant contents can be found in the page 3-4 of the supplement https://ieeexplore.ieee.org/ielx7/78/8933520/9189863/supp1-3023008.pdf?arnumber=9189863

Benefits of Random Fourier Feature Approximation

- For large data sets, RFF approximation may require less memory than the Nyström approximation in order to achieve the same small value of RAE

$$RAE = \frac{\|\boldsymbol{K} - \tilde{\boldsymbol{K}}\|_F}{\|\boldsymbol{K}\|_F}$$

  where $\boldsymbol{K}$ is the exact kernel matrix and $\tilde{\boldsymbol{K}} = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{L}}^T$ is its approximation
- The number of random features needed for constructing a good approximation is not sensitive to the sample size
- The number of data points needed by the Nyström approximation increases with the sample size