Deep Neural Networks for Large-scale Complex Spatial and Spatio-temporal Processes

PhD defense presentation by Pratik Nag
Supervised by Dr. Ying Sun (Chair)

Committee Members: Dr. Paula Moraga, Dr. Mohamed H. Elhoseiny, Dr. Veronica Berrocal

May 29, 2024

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of Science and Technology

EnvStat
@ KAUST

Outline

- Motivation
- Contributions
- Project 1: Bivariate DeepKriging for Large-scale Spatial Interpolation of Wind Fields
- Project 2: Spatio-Temporal DeepKriging for Probabilistic Interpolation and Forecasting
- Project 3: Efficient large-scale Nonstationary Spatial Covariance Function Estimation Using Convolutional Neural Networks
- Project 4: Spatial Normalizing Flows for Nonstationary Gaussian Processes

Motivation

- Spatial and spatio-temporal interpolation and forecasting is a challenging task because space-time data in environmental science are

Motivation

- Spatial and spatio-temporal interpolation and forecasting is a challenging task because space-time data in environmental science are
  - large.

Motivation

- Spatial and spatio-temporal interpolation and forecasting is a challenging task because space-time data in environmental science are
  - large.
  - exhibit complex spatio-temporal dependence.

Motivation

- Spatial and spatio-temporal interpolation and forecasting is a challenging task because space-time data in environmental science are
  - large.
  - exhibit complex spatio-temporal dependence.
  - may not be stationary.

Motivation

- Spatial and spatio-temporal interpolation and forecasting is a challenging task because space-time data in environmental science are
  - large.
  - exhibit complex spatio-temporal dependence.
  - may not be stationary.
- In spatial statistics, Kriging is widely used for spatial prediction and forecasting.

Motivation

- Spatial and spatio-temporal interpolation and forecasting is a challenging task because space-time data in environmental science are
  - large.
  - exhibit complex spatio-temporal dependence.
  - may not be stationary.
- In spatial statistics, Kriging is widely used for spatial prediction and forecasting.
- Features of Kriging predictor:

Motivation

- Spatial and spatio-temporal interpolation and forecasting is a challenging task because space-time data in environmental science are
  - large.
  - exhibit complex spatio-temporal dependence.
  - may not be stationary.
- In spatial statistics, Kriging is widely used for spatial prediction and forecasting.
- Features of Kriging predictor:
  - it is the best linear unbiased predictor (BLUP).

Motivation

- Spatial and spatio-temporal interpolation and forecasting is a challenging task because space-time data in environmental science are
    - large.
    - exhibit complex spatio-temporal dependence.
    - may not be stationary.
- In spatial statistics, Kriging is widely used for spatial prediction and forecasting.
- Features of Kriging predictor:
    - it is the best linear unbiased predictor (BLUP).
    - it involves modeling the mean and the covariance function of a spatial and spatio-temporal process.

Challenges

- Drawbacks of Kriging prediction:

Challenges

- Drawbacks of Kriging prediction:
  - it is typically not optimal for non-Gaussian data, e.g., skewed, heavy-tailed, count, or categorical data.

Challenges

- Drawbacks of Kriging prediction:
  - it is typically not optimal for non-Gaussian data, e.g., skewed, heavy-tailed, count, or categorical data.
  - it is computationally expensive for massive datasets (for both estimation and prediction) as the Cholesky factorization for a matrix of size $n \times n$ has $\mathcal{O}(n^3)$ performance complexity and $\mathcal{O}(n^2)$ memory space complexity.

Challenges

- Drawbacks of Kriging prediction:
  - it is typically not optimal for non-Gaussian data, e.g., skewed, heavy-tailed, count, or categorical data.
  - it is computationally expensive for massive datasets (for both estimation and prediction) as the Cholesky factorization for a matrix of size $n \times n$ has $\mathcal{O}(n^3)$ performance complexity and $\mathcal{O}(n^2)$ memory space complexity.
- Deep learning provides a computationally scalable methodology for a variety of data types and non-linear prediction. However, prediction uncertainty is an issue.

Contributions

This thesis aim at developing novel methodologies through deep learning for spatio-temporal statistics.

- Project 1 proposes a spatially dependent neural network (**Biv.DeepKriging**) to perform bivariate spatial prediction and give prediction uncertainties.

Contributions

This thesis aim at developing novel methodologies through deep learning for spatio-temporal statistics.

- Project 1 proposes a spatially dependent neural network (**Biv.DeepKriging**) to perform bivariate spatial prediction and give prediction uncertainties.
- Project 2 proposes a spatio-temporal deep learning framework (**Space-Time.DeepKriging**) for large-scale interpolation and probabilistic forecasting (**QConvLSTM**).

Contributions

This thesis aim at developing novel methodologies through deep learning for spatio-temporal statistics.

- Project 1 proposes a spatially dependent neural network (**Biv.DeepKriging**) to perform bivariate spatial prediction and give prediction uncertainties.
- Project 2 proposes a spatio-temporal deep learning framework (**Space-Time.DeepKriging**) for large-scale interpolation and probabilistic forecasting (**QConvLSTM**).
- Project 3 implements the nonstationary Matérn kernel in `ExaGeoSTAT` using HPC and Convolutional Neural Networks (**ConvNet**) for large datasets.

Contributions

This thesis aim at developing novel methodologies through deep learning for spatio-temporal statistics.

- Project 1 proposes a spatially dependent neural network (**Biv.DeepKriging**) to perform bivariate spatial prediction and give prediction uncertainties.
- Project 2 proposes a spatio-temporal deep learning framework (**Space-Time.DeepKriging**) for large-scale interpolation and probabilistic forecasting (**QConvLSTM**).
- Project 3 implements the nonstationary Matérn kernel in `ExaGeoSTAT` using HPC and Convolutional Neural Networks (**ConvNet**) for large datasets.
- Project 4 proposes a generalized spatial warping function called Spatial Normalizing Flows to model complex nonstationary fields.

Project 1: Bivariate DeepKriging for Large-scale Spatial Interpolation of Wind Fields

Background

- The DeepKriging as proposed by Chen et al. (2024)[a] uses basis functions as embedding layer for the deep neural network to model the univariate spatial process.

[a]Chen, W., Y. Li, B. J. Reich, and Y. Sun (2024). Deepkriging: Spatially dependent deep neural networks for spatial prediction. Accepted, Statistica Sinica, to appear.

Background

- The DeepKriging as proposed by Chen et al. (2024)[a] uses basis functions as embedding layer for the deep neural network to model the univariate spatial process.
- They also propose a histogram-based prediction interval computation methodology.

[a] Chen, W., Y. Li, B. J. Reich, and Y. Sun (2024). Deepkriging: Spatially dependent deep neural networks for spatial prediction. Accepted, Statistica Sinica, to appear.

Background

- The DeepKriging as proposed by Chen et al. (2024)[a] uses basis functions as embedding layer for the deep neural network to model the univariate spatial process.
- They also propose a histogram-based prediction interval computation methodology.
- This project is an extension of DeepKriging for bivariate spatial processes.

[a]Chen, W., Y. Li, B. J. Reich, and Y. Sun (2024). Deepkriging: Spatially dependent deep neural networks for spatial prediction. Accepted, Statistica Sinica, to appear.

Background

- The DeepKriging as proposed by Chen et al. (2024)[a] uses basis functions as embedding layer for the deep neural network to model the univariate spatial process.
- They also propose a histogram-based prediction interval computation methodology.
- This project is an extension of DeepKriging for bivariate spatial processes.
- A novel data-driven prediction interval mechanism is also devised which addresses the shortcomings of the prediction interval proposed by Chen et al. (2024).

[a] Chen, W., Y. Li, B. J. Reich, and Y. Sun (2024). Deepkriging: Spatially dependent deep neural networks for spatial prediction. Accepted, Statistica Sinica, to appear.

Theory of Bivariate DeepKriging

- let $\{\mathbf{Y}(\mathbf{s}), \mathbf{s} \in D\}, D \subseteq \mathbb{R}^p$, be a bivariate spatial process. A realization of the process $\mathbf{Z}(\mathbf{s}_i)$ is modeled as $\mathbf{Z}(\mathbf{s}_i) = \mathbf{Y}(\mathbf{s}_i) + \epsilon(\mathbf{s}_i)$ with nugget $\epsilon(\mathbf{s}_i)$, observed at locations $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N$.

Theory of Bivariate DeepKriging

- let $\{\mathbf{Y}(\mathbf{s}), \mathbf{s} \in D\}, D \subseteq \mathbb{R}^p$, be a bivariate spatial process. A realization of the process $\mathbf{Z}(\mathbf{s}_i)$ is modeled as $\mathbf{Z}(\mathbf{s}_i) = \mathbf{Y}(\mathbf{s}_i) + \epsilon(\mathbf{s}_i)$ with nugget $\epsilon(\mathbf{s}_i)$, observed at locations $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N$.
- The spatial model with covariates $\mathbf{X}(\mathbf{s}_i)$ can be written as $\mathbf{Y}(\mathbf{s}_i) = f(\mathbf{X}(\mathbf{s}_i)) + \gamma(\mathbf{s}_i)$, where $f(\cdot)$ is a nonlienar function and $\gamma(\mathbf{s}_i)$ is the underlying zero-mean spatial process.

Theory of Bivariate DeepKriging

- let $\{\mathbf{Y}(\mathbf{s}), \mathbf{s} \in D\}, D \subseteq \mathbb{R}^p$, be a bivariate spatial process. A realization of the process $\mathbf{Z}(\mathbf{s}_i)$ is modeled as $\mathbf{Z}(\mathbf{s}_i) = \mathbf{Y}(\mathbf{s}_i) + \epsilon(\mathbf{s}_i)$ with nugget $\epsilon(\mathbf{s}_i)$, observed at locations $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N$.
- The spatial model with covariates $\mathbf{X}(\mathbf{s}_i)$ can be written as $\mathbf{Y}(\mathbf{s}_i) = f(\mathbf{X}(\mathbf{s}_i)) + \gamma(\mathbf{s}_i)$, where $f(\cdot)$ is a nonlienar function and $\gamma(\mathbf{s}_i)$ is the underlying zero-mean spatial process.
- With the above formulation the theoretical backing of Bivariate DeepKriging (**Biv.DeepKriging**) is as follows:

Theorem

For a co-located bivariate spatial process, assuming that the latent variables are constructed with the same sets of basis functions, the Linear Model of Co-regionalization (LMC)[a] represents a special case of **Biv.DeepKriging**.

[a] Genton, M. G. and W. Kleiber (2015). Cross-covariance functions for multivariate geo-statistics. Statistical Science 30 (2), 147–163.

Bivariate DeepKriging: A Spatially Dependent Deep Neural Network

- Using the multivariate Karhunen-Loéve theorem[a] the spatial process $\gamma(\mathbf{s}_i)$ can be written of the form $\gamma(\mathbf{s}_i) \approx \sum_{b=1}^{K} \{w_{b,1}\phi_{b,1}(\mathbf{s}), w_{b,2}\phi_{b,2}(\mathbf{s})\}^T$ where $w_{b,u}$'s are independent random variables and $\phi_{b,u}(\mathbf{s}_i)$'s are the pairwise orthonormal basis functions corresponding to variable u, $u = 1, 2$.

[a] Paciorek, Christopher J., and Mark J. Schervish. "Spatial modelling using a new class of nonstationary covariance functions." Environmetrics: The official journal of the International Environmetrics Society 17.5 (2006): 483-506.

Bivariate DeepKriging: A Spatially Dependent Deep Neural Network

- Using the multivariate Karhunen-Loéve theorem[a] the spatial process $\gamma(\mathbf{s}_i)$ can be written of the form $\gamma(\mathbf{s}_i) \approx \sum_{b=1}^{K}\{w_{b,1}\phi_{b,1}(\mathbf{s}), w_{b,2}\phi_{b,2}(\mathbf{s})\}^T$ where $w_{b,u}$'s are independent random variables and $\phi_{b,u}(\mathbf{s}_i)$'s are the pairwise orthonormal basis functions corresponding to variable $u, u = 1, 2$.
- In this work the multi-resolution compactly supported Wendland radial basis function[b] is chosen to embedd the spatial locations.

[a] Paciorek, Christopher J., and Mark J. Schervish. "Spatial modelling using a new class of nonstationary covariance functions." Environmetrics: The official journal of the International Environmetrics Society 17.5 (2006): 483-506.

Bivariate DeepKriging: A Spatially Dependent Deep Neural Network

- Using the multivariate Karhunen-Loéve theorem[a] the spatial process $\gamma(\mathbf{s}_i)$ can be written of the form $\gamma(\mathbf{s}_i) \approx \sum_{b=1}^{K} \{w_{b,1}\phi_{b,1}(\mathbf{s}), w_{b,2}\phi_{b,2}(\mathbf{s})\}^T$ where $w_{b,u}$'s are independent random variables and $\phi_{b,u}(\mathbf{s}_i)$'s are the pairwise orthonormal basis functions corresponding to variable $u, u = 1, 2$.
- In this work the multi-resolution compactly supported Wendland radial basis function[b] is chosen to embedd the spatial locations.
- The idea of **Biv.DeepKriging**[c]: Through the basis function representation transform the spatial problem into a multi output regression problem by transforming the coordinate $\mathbf{s}$ to K basis functions.

[a] Paciorek, Christopher J., and Mark J. Schervish. "Spatial modelling using a new class of nonstationary covariance functions." Environmetrics: The official journal of the International Environmetrics Society 17.5 (2006): 483-506.

Bivariate DeepKriging: A Spatially Dependent Deep Neural Network

- Using the multivariate Karhunen-Loéve theorem[a] the spatial process $\gamma(\mathbf{s}_i)$ can be written of the form $\gamma(\mathbf{s}_i) \approx \sum_{b=1}^{K}\{w_{b,1}\phi_{b,1}(\mathbf{s}), w_{b,2}\phi_{b,2}(\mathbf{s})\}^T$ where $w_{b,u}$'s are independent random variables and $\phi_{b,u}(\mathbf{s}_i)$'s are the pairwise orthonormal basis functions corresponding to variable u, $u = 1, 2$.
- In this work the multi-resolution compactly supported Wendland radial basis function[b] is chosen to embedd the spatial locations.
- The idea of **Biv.DeepKriging**[c]: Through the basis function representation transform the spatial problem into a multi output regression problem by transforming the coordinate $\mathbf{s}$ to K basis functions.
- We pass the covariates and the bases together $\mathbf{X}_{\phi}(\mathbf{s}_i) = (\phi(\mathbf{s}_i)^T, \mathbf{X}_{vec}(\mathbf{s}_i)^T)^T$ as input to the DNN.

[a] Paciorek, Christopher J., and Mark J. Schervish. "Spatial modelling using a new class of nonstationary covariance functions." Environmetrics: The official journal of the International Environmetrics Society 17.5 (2006): 483-506.

Loss Function

- The optimal neural network based predictor is obtained as $\mathbf{f}_{NN}^{opt}(\mathbf{X}_{\phi}(\mathbf{s}_0)) = \text{argmin}_{\mathbf{f}_{NN}} R(\mathbf{f}_{NN}(\mathbf{X}_{\phi}(\mathbf{s}_0))|\mathbf{Z}_{vec})$.

Loss Function

- The optimal neural network based predictor is obtained as $\mathbf{f}_{NN}^{opt}(\mathbf{X}_\phi(\mathbf{s}_0)) = \text{argmin}_{\mathbf{f}_{NN}} R(\mathbf{f}_{NN}(\mathbf{X}_\phi(\mathbf{s}_0))|\mathbf{Z}_{vec})$.

- Here $R(\cdot)$ is given as

$$R\{\mathbf{f}_{NN}(\mathbf{X}_\phi(\mathbf{s}))|\mathbf{Z}_{vec}\} = \frac{1}{N}\sum_{n=1}^{N} M(\mathbf{s}_n),$$

where $M(\mathbf{s}_n) = \frac{w_1 \times (f_{NN_1}(\mathbf{X}_\phi(\mathbf{s})|\boldsymbol{\theta}) - Z_1(\mathbf{s}_n))^2 + w_2 \times (f_{NN_2}(\mathbf{X}_\phi(\mathbf{s})|\boldsymbol{\theta}) - Z_2(\mathbf{s}_n))^2}{2}$. and $w_u \propto \sigma_u^2, u = 1, 2$. We have chosen $w_u = \frac{1}{\sigma_u^2}$. Here σ_u^2 is unknown and can be estimated through the sample variance of the u-th variable.

Prediction Uncertainty (Prediction Mean)

- For the bivariate spatial prediction problem, the prediction at an unobserved location $\mathbf{s}_0$ can be expressed as

$$\hat{\mathbf{Z}}(\mathbf{s}_0) = \mathbf{f}_{NN}^{opt}(\mathbf{X}_\phi(\mathbf{s}_0)) + \epsilon(\mathbf{s}_0).$$

Prediction Uncertainty (Prediction Mean)

- For the bivariate spatial prediction problem, the prediction at an unobserved location $\mathbf{s}_0$ can be expressed as

$$\hat{\mathbf{Z}}(\mathbf{s}_0) = \mathbf{f}_{NN}^{opt}(\mathbf{X}_\phi(\mathbf{s}_0)) + \epsilon(\mathbf{s}_0).$$

- By employing ensembles, we can generate B replications of $\hat{\mathbf{Z}}(\mathbf{s}_0)$ at $\mathbf{s}_0$. Consequently, the prediction can be articulated as

$$\begin{aligned}\hat{\mathbf{Z}}(\mathbf{s}_0)^B &= \left(\frac{1}{B}\sum_{i=1}^{B}\hat{Z}_1(\mathbf{s}_0)_i, \frac{1}{B}\sum_{i=1}^{B}\hat{Z}_2(\mathbf{s}_0)_i\right)^T \\ &= \left(\frac{1}{B}\sum_{i=1}^{B}\left(f_{NN_1}^{opt}(\mathbf{X}_\phi(\mathbf{s}_0)) + \epsilon_1(\mathbf{s}_0)\right)_i, \frac{1}{B}\sum_{i=1}^{B}\left(f_{NN_2}^{opt}(\mathbf{X}_\phi(\mathbf{s}_0)) + \epsilon_2(\mathbf{s}_0)\right)_i\right)^T.\end{aligned}$$

Prediction Uncertainty (Prediction Mean)

- For the bivariate spatial prediction problem, the prediction at an unobserved location $\mathbf{s}_0$ can be expressed as

$$\hat{\mathbf{Z}}(\mathbf{s}_0) = \mathbf{f}_{NN}^{opt}(\mathbf{X}_\phi(\mathbf{s}_0)) + \epsilon(\mathbf{s}_0).$$

- By employing ensembles, we can generate B replications of $\hat{\mathbf{Z}}(\mathbf{s}_0)$ at $\mathbf{s}_0$. Consequently, the prediction can be articulated as

$$\hat{\mathbf{Z}}(\mathbf{s}_0)^B = \left(\frac{1}{B}\sum_{i=1}^{B}\hat{Z}_1(\mathbf{s}_0)_i, \frac{1}{B}\sum_{i=1}^{B}\hat{Z}_2(\mathbf{s}_0)_i\right)^T$$
$$= \left(\frac{1}{B}\sum_{i=1}^{B}\left(f_{NN_1}^{opt}(\mathbf{X}_\phi(\mathbf{s}_0)) + \epsilon_1(\mathbf{s}_0)\right)_i, \frac{1}{B}\sum_{i=1}^{B}\left(f_{NN_2}^{opt}(\mathbf{X}_\phi(\mathbf{s}_0)) + \epsilon_2(\mathbf{s}_0)\right)_i\right)^T.$$

- Employing the multidimensional Central Limit Theorem $\hat{\mathbf{Z}}(\mathbf{s}_0)^B$ follows bivariate normal distribution.

Prediction Uncertainty (Prediction Variance)

- The variance term associated with $Z_u(\mathbf{s}_0)$, for $u = 1, 2$, is $\sigma^2(Z_u(\mathbf{s}_0)) = \text{Var}(Y_u(\mathbf{s}_0)) + \text{Var}(\epsilon_u(\mathbf{s}_0))$, assuming independence between $Y_u(\mathbf{s}_0)$ and $\epsilon_u(\mathbf{s}_0)$.

Prediction Uncertainty (Prediction Variance)

- The variance term associated with $Z_u(\mathbf{s}_0)$, for $u = 1, 2$, is $\sigma^2(Z_u(\mathbf{s}_0)) = \text{Var}(Y_u(\mathbf{s}_0)) + \text{Var}(\epsilon_u(\mathbf{s}_0))$, assuming independence between $Y_u(\mathbf{s}_0)$ and $\epsilon_u(\mathbf{s}_0)$.
- We can estimate $\text{Var}(Y_u(\mathbf{s}_0))$ as

$$\widehat{Var(Y_u(\mathbf{s}_0))} = \frac{1}{B-1}\sum_{i=1}^{B} f_{NN_u}^{opt}(\mathbf{X}\phi(\mathbf{s}_0))_i^2 - \left(\frac{1}{B}\sum_{i=1}^{B} f_{NN_u}^{opt}(\mathbf{X}_\phi(\mathbf{s}_0))_i\right)^2.$$

Prediction Uncertainty (Prediction Variance)

- The variance term associated with $Z_u(\mathbf{s}_0)$, for $u = 1, 2$, is $\sigma^2(Z_u(\mathbf{s}_0)) = \text{Var}(Y_u(\mathbf{s}_0)) + \text{Var}(\epsilon_u(\mathbf{s}_0))$, assuming independence between $Y_u(\mathbf{s}_0)$ and $\epsilon_u(\mathbf{s}_0)$.
- We can estimate $\text{Var}(Y_u(\mathbf{s}_0))$ as

$$\widehat{Var(Y_u(\mathbf{s}_0))} = \frac{1}{B-1}\sum_{i=1}^{B} f_{NN_u}^{opt}(\mathbf{X}\phi(\mathbf{s}_0))_i^2 - \left(\frac{1}{B}\sum_{i=1}^{B} f_{NN_u}^{opt}(\mathbf{X}_\phi(\mathbf{s}_0))_i\right)^2.$$

- It can be shown that the noise variance will be the following

$$r_u^2(\mathbf{s}_0) = \max\{(Z_u(\mathbf{s}_0) - \frac{1}{B}\sum_{i=1}^{B} f_{NN_u}^{\text{opt}}(\mathbf{X}_\phi(\mathbf{s}_0))_i)^2 - \text{Var}\widehat{(Y_u(\mathbf{s}_0))}, 0\}.$$

Prediction Uncertainty (Prediction Variance)

- The variance term associated with $Z_u(\mathbf{s}_0)$, for $u = 1, 2$, is $\sigma^2(Z_u(\mathbf{s}_0)) = \text{Var}(Y_u(\mathbf{s}_0)) + \text{Var}(\epsilon_u(\mathbf{s}_0))$, assuming independence between $Y_u(\mathbf{s}_0)$ and $\epsilon_u(\mathbf{s}_0)$.
- We can estimate $\text{Var}(Y_u(\mathbf{s}_0))$ as

$$\widehat{Var(Y_u(\mathbf{s}_0))} = \frac{1}{B-1}\sum_{i=1}^{B} f^{opt}_{NN_u}(\mathbf{X}\phi(\mathbf{s}_0))_i^2 - \left(\frac{1}{B}\sum_{i=1}^{B} f^{opt}_{NN_u}(\mathbf{X}_\phi(\mathbf{s}_0))_i\right)^2.$$

- It can be shown that the noise variance will be the following

$$r_u^2(\mathbf{s}_0) = \max\{(Z_u(\mathbf{s}_0) - \frac{1}{B}\sum_{i=1}^{B} f^{\text{opt}}_{NN_u}(\mathbf{X}_\phi(\mathbf{s}_0))_i)^2 - \text{Var}\widehat{(Y_u(\mathbf{s}_0))}, 0\}.$$

- Computation of $r_u^2(\mathbf{s}_0)$ as defined previously is infeasible as we do not have $Z_u(\mathbf{s}_0)$. Hence we estimate $r_u^2(\mathbf{s}_0)$ through $\hat{r}_u^2(\mathbf{s}_0) = \frac{1}{G}\sum_{\mathbf{s}_g \in D_{20}} r_u^2(\mathbf{s}_g)$, such that $\mathbf{s}_g$'s are the nearest G locations to $\mathbf{s}_0$.

Prediction Uncertainty: Prediction Interval

- Then for a given test data location $\mathbf{s}_0$ the prediction interval is

$$\hat{Z}_u(\mathbf{s}_0)^B \pm t_{(1-\alpha/2),\mathrm{df}} \sqrt{\frac{1}{B}\left(\widehat{\mathrm{Var}(Y_u(\mathbf{s}_0))} + \hat{r}_u^2(\mathbf{s}_0)\right)}, \; u = 1, 2,$$

where $t_{(1-\alpha/2),\mathrm{df}}$ represents the $1-\alpha/2$ quantile of the t-distribution with df degrees of freedom, $\mathrm{df} = N - p$, where p denotes the number of estimated parameters.

Algorithm for Prediction Interval

Algorithm 1 Prediction Intervals Algorithm

Split $\mathbf{D}$ into $\mathbf{D_1}$ and $\mathbf{D_2}$ equally.
Further split $\mathbf{D_1}$ into $\mathbf{D_{11}}$ and $\mathbf{D_{12}}$.
Train a deep neural network (**DNN**) of L layers on $\mathbf{D_{11}}$.
Take $\mathbf{B}$ random samples $\{\mathbf{D_1^1}, \mathbf{D_1^2}, ..., \mathbf{D_1^B}\}$ from $\mathbf{D_1}$.
for $i \leftarrow 1$ to $\mathbf{B}$ **do**
    Fix the weights of the first L_0 layers of the **DNN** and train the last $L - L_0$ layers on $\mathbf{D_1^i}$.
    Train the **DNN** on $\mathbf{D_1^i}$ and store the result. Denote it as $\mathbf{f}_{NN}^{opt}(\mathbf{X}_\phi(\mathbf{s}))_i$.
end for
for location $\mathbf{s}_k$ in $\mathbf{D_2}$ **do**
    Calculate $\hat{Z}_u(\mathbf{s}_k)^B$(6) and $\widehat{Var(Y_u(\mathbf{s}_k))}$ (7).
    Calculate $r_u^2(\mathbf{s}_k)$ (8).
end for
For test location $\mathbf{s}_0$, obtain the set $\mathbf{D_{20}} = \{\mathbf{s}_k : \mathbf{s}_k \in \mathbf{D_2}\}$ of the nearest G locations from $\mathbf{s}_0$.
Calculate $\hat{Z}_u(\mathbf{s}_0)^B$(6), $\widehat{Var(Y_u(\mathbf{s}_k))}$ (7), and $\hat{r}_u^2(\mathbf{s}_0)$ (10).
Calculate the prediction interval as defined in (9).
$\triangleright$ Where u stands for the u-th variable, $u = 1, 2$.

Simulation Studies

- Three separate simulation scenarios are devised:
    - Gaussian with parsimonious Matérn covariance.
    - Non-Gaussian process with covariates : Gaussian process is generated and it is then transformed by the Tukey G and H transformation to yield non-Gaussian field
    - Nonstationary process : Nonlinear combinations of basis functions are taken into consideration for this process generation.
- Each simulation scenario is replicated 100 times.

Simulation Studies

The proposed model (**Biv.DeepKriging**) is compared against Gaussian kriging with parsimonious Matérn covariance(**CoKriging.Matérn**) and Linear Model of Coregionalization (**CoKriging.LMC**) respectively.

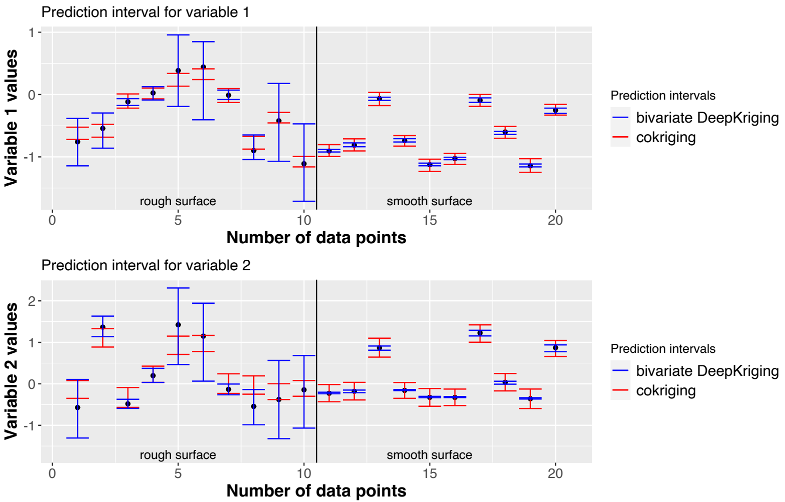
Different metrics such as the Mean Square Prediction Error (MSPE), Prediction Interval Coverage Probability (PICP) and Mean Prediction Interval Width (MPIW) are considered for comparing the predictions and the prediction intervals.

Table: Comparison on both the variables over different simulation settings.

Simulation type	Models	**MSPE$_1$**	**SE$_1$**	**PICP$_1$**	**MPIW$_1$**	**MSPE$_2$**	**SE$_2$**	**PICP$_2$**	**MPIW$_2$**
Gaussian	**CoKriging.Matérn$_{true}$**	0.23	0.11	0.95	0.87	0.21	0.09	0.95	0.92
	Biv.DeepKriging	0.24	0.19	0.94	0.98	0.24	0.18	0.95	1.07
non-Gaussian	**CoKriging.Matérn**	3.48 ($\times10^3$)	0.29 ($\times10^3$)	0.27	6.21	0.98 ($\times10^3$)	0.49 ($\times10^3$)	0.26	6.16
	CoKriging.LMC	87.4	12.13	0.58	11.2	94.5	21.9	0.51	9.99
	Biv.DeepKriging	32.7	11.6	0.94	29.9	23.8	9.11	0.94	29.4
non-stationary	**CoKriging.Matérn**	1.95	0.75	0.92	3.53	0.13	0.02	0.09	1.01
	CoKriging.LMC	1.26	0.14	0.92	3.49	0.14	0.02	0.10	1.33
	Biv.DeepKriging	7.52 ($\times10^{-4}$)	1.01 ($\times10^{-4}$)	0.96	0.16	6.83 ($\times10^{-4}$)	1.08 ($\times10^{-4}$)	0.95	0.19

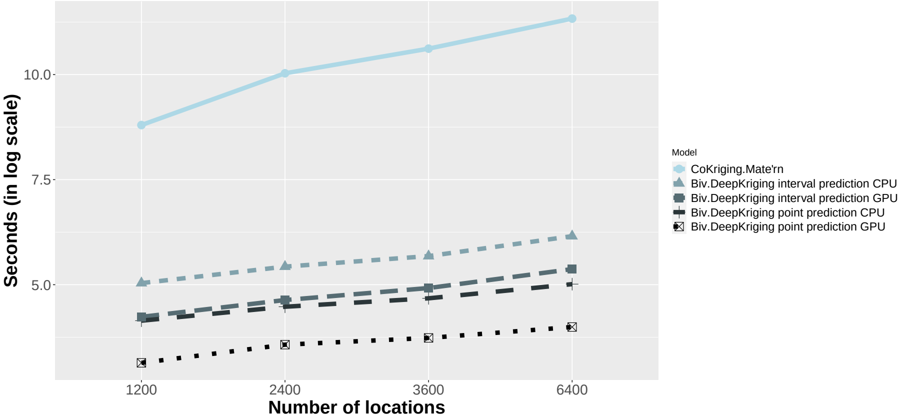
Simulation studies: Prediction intervals

Figure: Prediction interval for variable 1 and variable 2 for the nonstationary simulation.



Simulation Studies: Computation Time

Figure: Total computation time (in secconds) for different models in log scale for different number of locations



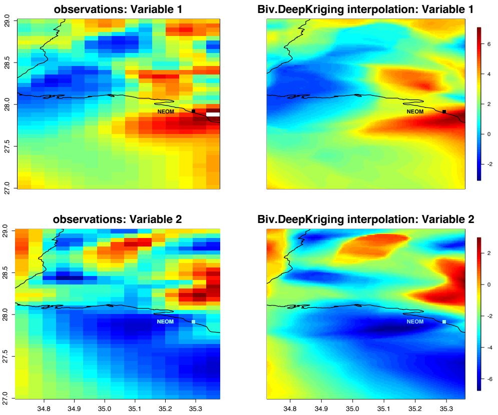
Application on Wind Data

- The U and V components of wind over the Middle East, encompassing 506,771 locations, are considered for this study.
- For **CoKriging.Matérn** total computation time was 2.18 days where as for **Biv.DeepKriging** it took **16.81 minutes** for point prediction and **55.61 minutes** for interval prediction.

Models	$\mathbf{RMSPE}_1$	$\mathbf{RMSPE}_2$
CoKriging.Matérn	0.882	4.066
Biv.DeepKriging$_{147000}$	0.488	0.438
Biv.DeepKriging$_{450000}$	**0.394**	**0.392**

Models	$PICP_1$	$PICP_2$	$MPIW_1$	$MPIW_2$
CoKriging.Matérn	0.601	0.734	1.671	1.343
Biv.DeepKriging	**0.971**	**0.950**	**1.226**	**1.340**

Spatial Downscaling



- A high resolution interpolation is given ($1km \times 1km$) for the region near NEOM, an upcoming smart city in Saudi Arabia. This downscaling can help understand the wind pattern better and can potentially help in wind energy setup in the area.

Conclusion

- The proposed framework, **Biv.DeepKriging**, which generalizes the Linear Model of Coregionalization, is suitable for modeling bivariate non-Gaussian and nonstationary spatial fields.

Conclusion

- The proposed framework, **Biv.DeepKriging**, which generalizes the Linear Model of Coregionalization, is suitable for modeling bivariate non-Gaussian and nonstationary spatial fields.

- The proposed method is also computationally scalable and can be implemented for large-scale datasets.

Conclusion

- The proposed framework, **Biv.DeepKriging**, which generalizes the Linear Model of Coregionalization, is suitable for modeling bivariate non-Gaussian and nonstationary spatial fields.
- The proposed method is also computationally scalable and can be implemented for large-scale datasets.
- The proposed prediction interval technique does not rely on the distribution of the data and can be applied to any kind of application beyond spatial modeling.

Project 2: Spatio-Temporal DeepKriging for Probabilistic Interpolation and Forecasting

Background

- This project is an extension of DeepKriging for spatio-temporal scenario, where interpolation and forecasting is done through a 2-stage modeling framework. The project also proposes a novel implementation of quantile neural networks to obtain prediction uncertainty.

Background

- This project is an extension of DeepKriging for spatio-temporal scenario, where interpolation and forecasting is done through a 2-stage modeling framework. The project also proposes a novel implementation of quantile neural networks to obtain prediction uncertainty.
- Consider the real valued spatio-temporal random field $\{Y(\mathbf{s}, t), \mathbf{s} \in D, t \in \mathcal{T}\}, D \subseteq \mathbb{R}^p, \mathcal{T} \subseteq \mathbb{R}$. Assuming the data is observed at N locations and K time points, the realizations can be given as $\mathbf{Z}_{N,K} = \{Z(\mathbf{s}_1, t_1), Z(\mathbf{s}_2, t_1), \dots, Z(\mathbf{s}_N, t_K)\}$ such that

$$Z(\mathbf{s}_i, t_j) = Y(\mathbf{s}_i, t_j) + \epsilon(\mathbf{s}_i, t_j).$$

Background

- This project is an extension of DeepKriging for spatio-temporal scenario, where interpolation and forecasting is done through a 2-stage modeling framework. The project also proposes a novel implementation of quantile neural networks to obtain prediction uncertainty.
- Consider the real valued spatio-temporal random field $\{Y(\mathbf{s}, t), \mathbf{s} \in D, t \in \mathcal{T}\}, D \subseteq \mathbb{R}^p, \mathcal{T} \subseteq \mathbb{R}$. Assuming the data is observed at N locations and K time points, the realizations can be given as $\mathbf{Z}_{N,K} = \{Z(\mathbf{s}_1, t_1), Z(\mathbf{s}_2, t_1), \ldots, Z(\mathbf{s}_N, t_K)\}$ such that

$$Z(\mathbf{s}_i, t_j) = Y(\mathbf{s}_i, t_j) + \epsilon(\mathbf{s}_i, t_j).$$

- Given observations $\mathbf{Z}_{N,K}$, two common goals of spatio-temporal prediction are probabilistic interpolation, i.e., predict the true process $Y(\mathbf{s}_0, t)$ at unobserved spatial location $\mathbf{s}_0$, and forecasting, i.e., predict $Y(\mathbf{s}_0, t_{K+u})$ at unobserved location $\mathbf{s}_0$ at a future time point t_{K+u}.

Optimal Predictor for Probabilistic Interpolation

- The optimal predictor can written as:

$$\hat{Y}_{\tau}^{opt}((\mathbf{s}_0, t)|\mathbf{Z}_{N,K}) = \underset{\hat{Y}}{\text{argmin}}\, R_1(\hat{Y}_{\tau}(\mathbf{s}_0, t)|\mathbf{Z}_{N,K}),$$

where $R_1(\cdot)$ represents the true risk function necessary for obtaining the τ-th quantile prediction.

Optimal Predictor for Probabilistic Interpolation

- The optimal predictor can written as:

$$\hat{Y}_\tau^{opt}((\mathbf{s}_0, t)|\mathbf{Z}_{N,K}) = \underset{\hat{Y}}{\text{argmin}}\, R_1(\hat{Y}_\tau(\mathbf{s}_0, t)|\mathbf{Z}_{N,K}),$$

where $R_1(\cdot)$ represents the true risk function necessary for obtaining the τ-th quantile prediction.

- An estimation for $R_1(\cdot)$ can be expressed through the quantile loss function, defined as:

$$R_1^{emp}(\hat{Y}_\tau(\mathbf{s}, t)|\mathbf{Z}_{N,K}) = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\rho_\tau(\hat{Y}_\tau(\mathbf{s}_n, t_k) - Z(\mathbf{s}_n, t_k)),$$

where $\rho_\tau(v) = v(\tau - I(v < 0))$ and $\tau \in (0, 1)$ is quantile level.

Space-Time.DeepKriging: DNN for Interpolation

- Similar to DeepKriging[a] a single-output deep neural network structure (**Space-Time. DeepKriging**) is used to build the spatio-temporal DeepKriging framework with basis functions as inputs.

[a]Chen, W., Y. Li, B. J. Reich, and Y. Sun (2024). Deepkriging: Spatially dependent deep neural networks for spatial prediction. Accepted, Statistica Sinica, to appear.

Space-Time.DeepKriging: DNN for Interpolation

- Similar to DeepKriging[a] a single-output deep neural network structure (**Space-Time. DeepKriging**) is used to build the spatio-temporal DeepKriging framework with basis functions as inputs.
- Wendland's compactly supported radial basis functions are used for spatial location embedding and Gaussian radial bases are used for temporal embedding.

[a] Chen, W., Y. Li, B. J. Reich, and Y. Sun (2024). Deepkriging: Spatially dependent deep neural networks for spatial prediction. Accepted, Statistica Sinica, to appear.

Space-Time.DeepKriging: DNN for Interpolation

- Similar to DeepKriging[a] a single-output deep neural network structure (**Space-Time. DeepKriging**) is used to build the spatio-temporal DeepKriging framework with basis functions as inputs.
- Wendland's compactly supported radial basis functions are used for spatial location embedding and Gaussian radial bases are used for temporal embedding.
- Hence $\hat{Y}_\tau(\cdot,\cdot)$ can be expressed through the DNN as:

$$\hat{Y}_\tau(\mathbf{s}, t) = \Psi(\tau, f_{NN_\tau}(\mathbf{X}_\phi(\mathbf{s}, t))),$$

where $\mathbf{X}_\phi(\mathbf{s}, t)$ is the set of stacked basis functions, $f_{NN_\tau}(\mathbf{X}_\phi(\mathbf{s}, t))$ is the DNN output at quantile level τ, and $\Psi(\cdot,\cdot)$ is the activation function of the output layer.

[a] Chen, W., Y. Li, B. J. Reich, and Y. Sun (2024). Deepkriging: Spatially dependent deep neural networks for spatial prediction. Accepted, Statistica Sinica, to appear.

The Output Layer Activation $\Psi(\cdot,\cdot)$

- In theory, quantile regression lines are expected not to intersect; however, unconstrained optimization of $R_1^{emp}(\hat{Y}_\tau(\mathbf{s},t)|\mathbf{Z}_{N,K})$ may inadvertently introduce crossing issues.

The Output Layer Activation $\Psi(\cdot, \cdot)$

- In theory, quantile regression lines are expected not to intersect; however, unconstrained optimization of $R_1^{emp}(\hat{Y}_\tau(\mathbf{s}, t)|\mathbf{Z}_{N,K})$ may inadvertently introduce crossing issues.

- To avoid quantile cross-over, the following activation function for the output layer is proposed:

$$\Psi(\tau, x) = \begin{cases} x & \text{for } \tau = 0.5 \\ f_{Constant} + \frac{\lambda(\tau - 0.5)}{1+e^{-x}} & \text{for } \tau > 0.5 \\ f_{Constant} - \frac{\lambda(0.5-\tau)}{1+e^{-x}} & \text{for } \tau < 0.5, \end{cases}$$

The Output Layer Activation $\Psi(\cdot, \cdot)$

- In theory, quantile regression lines are expected not to intersect; however, unconstrained optimization of $R_1^{emp}(\hat{Y}_\tau(\mathbf{s}, t)|\mathbf{Z}_{N,K})$ may inadvertently introduce crossing issues.

- To avoid quantile cross-over, the following activation function for the output layer is proposed:

$$\Psi(\tau, x) = \begin{cases} x & \text{for } \tau = 0.5 \\ f_{Constant} + \frac{\lambda(\tau - 0.5)}{1+e^{-x}} & \text{for } \tau > 0.5 \\ f_{Constant} - \frac{\lambda(0.5-\tau)}{1+e^{-x}} & \text{for } \tau < 0.5, \end{cases}$$

- Here $f_{Constant}$ is the model output for quantile level 0.5, λ is the hyperparameter proportional to the variance of the data.

Probabilistic Forecasting

- The Long short-term memory (LSTM) network is used to perform quantile based forecast of the time series at time point t_{K+u} (call it **QLSTM**).

Probabilistic Forecasting

- The Long short-term memory (LSTM) network is used to perform quantile based forecast of the time series at time point t_{K+u} (call it **QLSTM**).
- Here $\hat{Y}_\tau(\mathbf{s}_0, t_{K+u}) = f_{NN_\tau}^{LSTM}\widehat{(\mathbf{s}_0, t_{K+u})}$, where $f_{NN_\tau}^{LSTM}\widehat{(\mathbf{s}_0, t_{K+u})}$ is a multi-layer stacked LSTM network.

Probabilistic Forecasting

- The Long short-term memory (LSTM) network is used to perform quantile based forecast of the time series at time point t_{K+u} (call it **QLSTM**).
- Here $\hat{Y}_\tau(\mathbf{s}_0, t_{K+u}) = \widehat{f^{LSTM}_{NN_\tau}(\mathbf{s}_0, t_{K+u})}$, where $\widehat{f^{LSTM}_{NN_\tau}(\mathbf{s}_0, t_{K+u})}$ is a multi-layer stacked LSTM network.
- Although **QLSTM** is highly effective for capturing temporal dependence, it does not use information from other locations.

Probabilistic Forecasting

- The Long short-term memory (LSTM) network is used to perform quantile based forecast of the time series at time point t_{K+u} (call it **QLSTM**).
- Here $\hat{Y}_\tau(\mathbf{s}_0, t_{K+u}) = \widehat{f_{NN_\tau}^{LSTM}(\mathbf{s}_0, t_{K+u})}$, where $\widehat{f_{NN_\tau}^{LSTM}(\mathbf{s}_0, t_{K+u})}$ is a multi-layer stacked LSTM network.
- Although **QLSTM** is highly effective for capturing temporal dependence, it does not use information from other locations.
- For space-time data, this project propose the convolutional LSTM which includes data from other locations by passing the CNN layer as the input to the LSTM layer (call it **QConvLSTM**).

Optimal Predictor for **QConvLSTM**

- The optimal predictor can written as:

$$\hat{Y}_{\tau}^{opt}((\mathbf{s}_0, t_{K+u})|\mathbf{Z}_{N,K}) = \underset{\hat{Y}}{\operatorname{argmin}}\, R_2(\hat{Y}_{\tau}(\mathbf{s}_0, t)|\mathbf{Z}_{N,K}),$$

where $R_2(\cdot)$ represents the true risk function.

Optimal Predictor for **QConvLSTM**

- The optimal predictor can written as:

$$\hat{Y}_{\tau}^{opt}((\mathbf{s}_0, t_{K+u})|\mathbf{Z}_{N,K}) = \underset{\hat{Y}}{\operatorname{argmin}}\, R_2(\hat{Y}_{\tau}(\mathbf{s}_0, t)|\mathbf{Z}_{N,K}),$$

  where $R_2(\cdot)$ represents the true risk function.
- An estimarte of $R_2(\cdot)$ can be written as:

$$R_2^{emp}(\hat{Y}_{\tau}(\mathbf{s}_0, t)|\mathbf{Z}_{N,K}) = \frac{1}{K}\sum_{k=1}^{K} \rho_{\tau}(f_{NN_{\tau}}^{Conv}(\mathbf{s}_0, t_k) - \mathbf{X}_k^{NN}),$$

  where $f_{NN_{\tau}}^{Conv}(\mathbf{s}_0, t_k)$ is the output of **QConvLSTM**.

Optimal Predictor for **QConvLSTM**

- The optimal predictor can written as:

$$\hat{Y}_{\tau}^{opt}((\mathbf{s}_0, t_{K+u})|\mathbf{Z}_{N,K}) = \underset{\hat{Y}}{\text{argmin}}\, R_2(\hat{Y}_{\tau}(\mathbf{s}_0, t)|\mathbf{Z}_{N,K}),$$

  where $R_2(\cdot)$ represents the true risk function.

- An estimarte of $R_2(\cdot)$ can be written as:

$$R_2^{emp}(\hat{Y}_{\tau}(\mathbf{s}_0, t)|\mathbf{Z}_{N,K}) = \frac{1}{K}\sum_{k=1}^{K} \rho_{\tau}(f_{NN_{\tau}}^{Conv}(\mathbf{s}_0, t_k) - \mathbf{X}_k^{NN}),$$

  where $f_{NN_{\tau}}^{Conv}(\mathbf{s}_0, t_k)$ is the output of **QConvLSTM**.

- Here $\mathbf{X}^{NN} = \{\widehat{f_{NN_{\tau}}}(\mathbf{X}_{\phi}(\mathbf{s_0}, t_1)), \cdots, \widehat{f_{NN_{\tau}}}(\mathbf{X}_{\phi}(\mathbf{s_0}, t_K))\}$.

Optimal Predictor for **QConvLSTM**

- The optimal predictor can written as:

$$\hat{Y}_{\tau}^{opt}((\mathbf{s}_0, t_{K+u})|\mathbf{Z}_{N,K}) = \underset{\hat{Y}}{\operatorname{argmin}}\, R_2(\hat{Y}_{\tau}(\mathbf{s}_0, t)|\mathbf{Z}_{N,K}),$$

  where $R_2(\cdot)$ represents the true risk function.
- An estimarte of $R_2(\cdot)$ can be written as:

$$R_2^{emp}(\hat{Y}_{\tau}(\mathbf{s}_0, t)|\mathbf{Z}_{N,K}) = \frac{1}{K}\sum_{k=1}^{K}\rho_{\tau}(f_{NN_{\tau}}^{Conv}(\mathbf{s}_0, t_k) - \mathbf{X}_k^{NN}),$$

  where $f_{NN_{\tau}}^{Conv}(\mathbf{s}_0, t_k)$ is the output of **QConvLSTM**.
- Here $\mathbf{X}^{NN} = \{\widehat{f_{NN_{\tau}}}(\mathbf{X}_{\phi}(\mathbf{s_0}, t_1)), \cdots, \widehat{f_{NN_{\tau}}}(\mathbf{X}_{\phi}(\mathbf{s_0}, t_K))\}$.
- Note that, the input to $f_{NN_{\tau}}^{Conv}(\mathbf{s}_0, t_k)$ here is:
  $\mathbf{X}^{NN_{CONV}} = \{\mathcal{A}(\mathbf{s}_0, t_1), \ldots, \mathcal{A}(\mathbf{s}_0, t_K)\}$, where $\mathcal{A}(\mathbf{s}_0, t_k)$ is a $r \times r$ matrix of interpolation over a gridded neighbourhood $N_{\mathbf{s}_0}$ around $\mathbf{s}_0$.

Simulation Studies

- The proposed **Space-Time.DeepKriging** won the KAUST competition in large-scale prediction on 100k and 1M space-time locations with double digit improvement in percentage for MSPE over competing methods such as the Veccia's approximation and block composite likelihood.

Simulation Studies

- The proposed **Space-Time.DeepKriging** won the KAUST competition in large-scale prediction on 100k and 1M space-time locations with double digit improvement in percentage for MSPE over competing methods such as the Veccia's approximation and block composite likelihood.

- We compare the method on a simulated nonstationary field with 50k space-time locations with other competitive methods that can be applied for large-scale interpolation and forecasting.

Numeric Results on Simulated Data

Table: Average MSPE of prediction for simulated data. Here SE stands for standard error of the predictions.

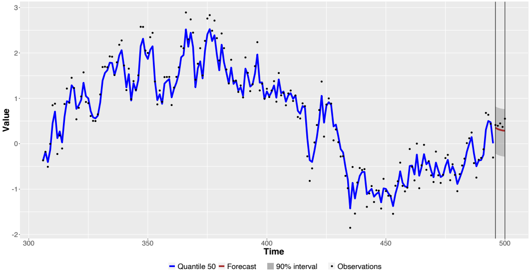
Models	MSPE	SE
Space-Time.DeepKriging	0.167	0.073
GpGp	0.746	0.288

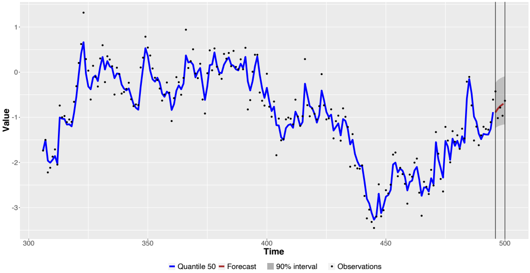
Table: Average MSPE, MPIW and PICP of forecast for simulated data.

Models	Avg.MSPE	SE	Avg.MPIW	SE	Avg.PICP
QConvLSTM	0.267	0.219	1.462	0.126	90.39
ARIMA	0.277	0.278	2.262	0.082	90.72
QLSTM	0.392	0.523	1.558	0.316	89.94
GpGp	0.839	0.358	-	-	-

Forecast on Simulated Data

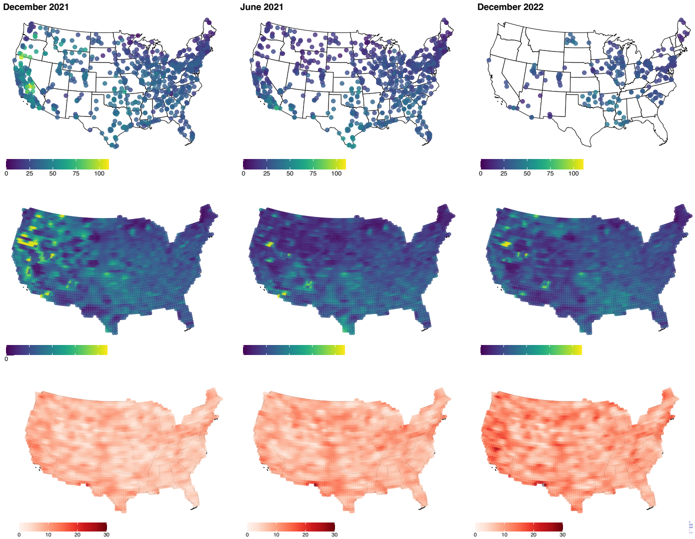
- Forecasting at specific observed locations using **QConvLSTM**.





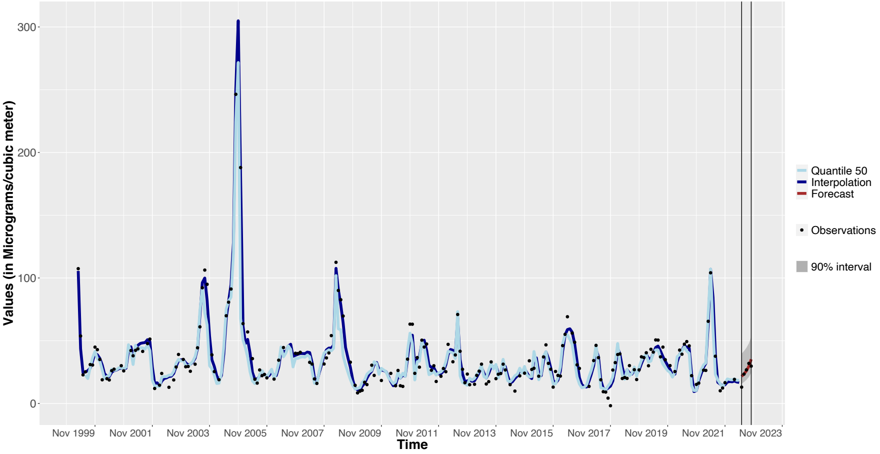
Application on $PM_{2.5}$: Interpolation

We also apply our method to the $PM_{2.5}$ data over USA with over 200,000 space-time locations.



Application on $PM_{2.5}$: Forecast for San Francisco

Forecast period for the last six observed months up to December 2022:



Conclusion

- In this study, without making any parametric assumptions about the underlying distribution of the data, a novel, easy-to-use methodology is established for interpolation as well as forecasting for spatio-temporal processes.

Conclusion

- In this study, without making any parametric assumptions about the underlying distribution of the data, a novel, easy-to-use methodology is established for interpolation as well as forecasting for spatio-temporal processes.
- In order to further enhance the deep learning-based spatio-temporal modeling architecture, semi-parametric quantile-based prediction intervals are included.

Conclusion

- In this study, without making any parametric assumptions about the underlying distribution of the data, a novel, easy-to-use methodology is established for interpolation as well as forecasting for spatio-temporal processes.
- In order to further enhance the deep learning-based spatio-temporal modeling architecture, semi-parametric quantile-based prediction intervals are included.
- The proposed method for spatio-temporal interpolation and forecasting is valid for general class of non-Gaussian and nonstationary spatio-temporal processes.

Conclusion

- In this study, without making any parametric assumptions about the underlying distribution of the data, a novel, easy-to-use methodology is established for interpolation as well as forecasting for spatio-temporal processes.
- In order to further enhance the deep learning-based spatio-temporal modeling architecture, semi-parametric quantile-based prediction intervals are included.
- The proposed method for spatio-temporal interpolation and forecasting is valid for general class of non-Gaussian and nonstationary spatio-temporal processes.
- The proposed approach can be easily extended to large datasets with minimum hardware support.

Project 3: Efficient Large-scale Nonstationary Spatial Covariance Function Estimation using Convolutional Neural Networks

Introduction

- In environmental and ecological applications nonstationarity assumption is more realistic and can be accounted through Covariance Nonstationarity .

Introduction

- In environmental and ecological applications nonstationarity assumption is more realistic and can be accounted through Covariance Nonstationarity.
- This project gives a novel approach for modeling the nonstationary Matérn covariance function through HPC and convolutional neural networks.

Nonstationary Matérn Covariance Function

- Let $Z(\cdot)$ be a spatial process observed over locations $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N \in D \subseteq \mathbb{R}^d$, where $Z(\mathbf{s}_i) = \mu(\mathbf{s}_i) + Y(\mathbf{s}_i) + \epsilon, \mathbf{s}_i \in D$ with the underlying GRF $Y(\mathbf{s}_i)$ having covariance function $C(\cdot, \cdot)$.

Nonstationary Matérn Covariance Function

- Let $Z(\cdot)$ be a spatial process observed over locations $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N \in D \subseteq \mathbb{R}^d$, where $Z(\mathbf{s}_i) = \mu(\mathbf{s}_i) + Y(\mathbf{s}_i) + \epsilon, \mathbf{s}_i \in D$ with the underlying GRF $Y(\mathbf{s}_i)$ having covariance function $C(\cdot, \cdot)$.
- The nonstationary Matérn covariance:

$$C^{NS}(\mathbf{s}_i, \mathbf{s}_j; \boldsymbol{\theta}) = \tau(\mathbf{s}_i)\tau(\mathbf{s}_j)\mathbb{1}_{ij}(\mathbf{s}_i, \mathbf{s}_j) + \frac{\sigma(\mathbf{s}_i)\sigma(\mathbf{s}_j)|\Sigma(\mathbf{s}_i)|^{1/4}|\Sigma(\mathbf{s}_j)|^{1/4}}{\Gamma(\bar{\nu}(\mathbf{s}_i, \mathbf{s}_j))2^{\bar{\nu}(\mathbf{s}_i, \mathbf{s}_j)-1}}$$
$$\times \left|\frac{\Sigma(\mathbf{s}_i) + \Sigma(\mathbf{s}_j)}{2}\right|^{-1/2} \left(2\sqrt{\bar{\nu}(\mathbf{s}_i, \mathbf{s}_j)Q_{ij}}\right)^{\bar{\nu}(\mathbf{s}_i, \mathbf{s}_j)} \mathcal{K}_{\bar{\nu}(\mathbf{s}_i, \mathbf{s}_j)}\left(2\sqrt{\bar{\nu}(\mathbf{s}_i, \mathbf{s}_j)Q_{ij}}\right),$$

and $\nu_{ij} = \frac{\nu(\mathbf{s}_i)+\nu(\mathbf{s}_j)}{2}$, where $\boldsymbol{\theta}(\mathbf{s}_i) = \{\Sigma(\mathbf{s}_i), \sigma(\mathbf{s}_i), \tau^2(\mathbf{s}_i), \nu(\mathbf{s}_i)\}$ are spatially varying parameters that control nonstationarity.

Nonstationary Matérn Covariance Function

- Let $Z(\cdot)$ be a spatial process observed over locations $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N \in D \subseteq \mathbb{R}^d$, where $Z(\mathbf{s}_i) = \mu(\mathbf{s}_i) + Y(\mathbf{s}_i) + \epsilon, \mathbf{s}_i \in D$ with the underlying GRF $Y(\mathbf{s}_i)$ having covariance function $C(\cdot, \cdot)$.
- The nonstationary Matérn covariance:

$$C^{NS}(\mathbf{s}_i, \mathbf{s}_j; \boldsymbol{\theta}) = \tau(\mathbf{s}_i)\tau(\mathbf{s}_j)\mathbb{1}_{ij}(\mathbf{s}_i, \mathbf{s}_j) + \frac{\sigma(\mathbf{s}_i)\sigma(\mathbf{s}_j)|\Sigma(\mathbf{s}_i)|^{1/4}|\Sigma(\mathbf{s}_j)|^{1/4}}{\Gamma(\bar{\nu}(\mathbf{s}_i, \mathbf{s}_j))2^{\bar{\nu}(\mathbf{s}_i,\mathbf{s}_j)-1}} \times \left|\frac{\Sigma(\mathbf{s}_i) + \Sigma(\mathbf{s}_j)}{2}\right|^{-1/2} \left(2\sqrt{\bar{\nu}(\mathbf{s}_i, \mathbf{s}_j)Q_{ij}}\right)^{\bar{\nu}(\mathbf{s}_i,\mathbf{s}_j)} \mathcal{K}_{\bar{\nu}(\mathbf{s}_i,\mathbf{s}_j)}\left(2\sqrt{\bar{\nu}(\mathbf{s}_i, \mathbf{s}_j)Q_{ij}}\right),$$

  and $\nu_{ij} = \frac{\nu(\mathbf{s}_i)+\nu(\mathbf{s}_j)}{2}$, where $\boldsymbol{\theta}(\mathbf{s}_i) = \{\Sigma(\mathbf{s}_i), \sigma(\mathbf{s}_i), \tau^2(\mathbf{s}_i), \nu(\mathbf{s}_i)\}$ are spatially varying parameters that control nonstationarity.
- $\Sigma(\mathbf{s}_i)$ controls the spatial range and anisotropy, $\sigma(\mathbf{s}_i)$ controls the local standard deviation, $\tau^2(\mathbf{s}_i)$ controls the nugget effect, and $\nu(\mathbf{s}_i)$ controls the smoothness.

Nonstationary Matérn Covariance Function

- Let $Z(\cdot)$ be a spatial process observed over locations $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N \in D \subseteq \mathbb{R}^d$, where $Z(\mathbf{s}_i) = \mu(\mathbf{s}_i) + Y(\mathbf{s}_i) + \epsilon, \mathbf{s}_i \in D$ with the underlying GRF $Y(\mathbf{s}_i)$ having covariance function $C(\cdot, \cdot)$.
- The nonstationary Matérn covariance:

$$C^{NS}(\mathbf{s}_i, \mathbf{s}_j; \boldsymbol{\theta}) = \tau(\mathbf{s}_i)\tau(\mathbf{s}_j)\mathbb{1}_{ij}(\mathbf{s}_i, \mathbf{s}_j) + \frac{\sigma(\mathbf{s}_i)\sigma(\mathbf{s}_j)|\Sigma(\mathbf{s}_i)|^{1/4}|\Sigma(\mathbf{s}_j)|^{1/4}}{\Gamma(\bar{\nu}(\mathbf{s}_i, \mathbf{s}_j))2^{\bar{\nu}(\mathbf{s}_i,\mathbf{s}_j)-1}}$$
$$\times \left|\frac{\Sigma(\mathbf{s}_i) + \Sigma(\mathbf{s}_j)}{2}\right|^{-1/2} \left(2\sqrt{\bar{\nu}(\mathbf{s}_i, \mathbf{s}_j)Q_{ij}}\right)^{\bar{\nu}(\mathbf{s}_i,\mathbf{s}_j)} \mathcal{K}_{\bar{\nu}(\mathbf{s}_i,\mathbf{s}_j)}\left(2\sqrt{\bar{\nu}(\mathbf{s}_i, \mathbf{s}_j)Q_{ij}}\right),$$

and $\nu_{ij} = \frac{\nu(\mathbf{s}_i)+\nu(\mathbf{s}_j)}{2}$, where $\boldsymbol{\theta}(\mathbf{s}_i) = \{\Sigma(\mathbf{s}_i), \sigma(\mathbf{s}_i), \tau^2(\mathbf{s}_i), \nu(\mathbf{s}_i)\}$ are spatially varying parameters that control nonstationarity.
- $\Sigma(\mathbf{s}_i)$ controls the spatial range and anisotropy, $\sigma(\mathbf{s}_i)$ controls the local standard deviation, $\tau^2(\mathbf{s}_i)$ controls the nugget effect, and $\nu(\mathbf{s}_i)$ controls the smoothness.
- $Q_{ij} = (\mathbf{s}_i - \mathbf{s}_j)^T \left(\frac{\Sigma_i+\Sigma_j}{2}\right)^{-1} (\mathbf{s}_i - \mathbf{s}_j)$ is the Mahalanobis distance between points $\mathbf{s}_i$ and $\mathbf{s}_j$.

Existing Methods and Challenges

- Existing methods:

[a] Risser MD, Calder CA (2017). "Local Likelihood Estimation for Covariance Functions with Spatially-Varying Parameters: The convoSPAT Package for R." Journal of Statistical Software, 81(14), 1–32. doi:10.18637/jss.v081.i14.

Existing Methods and Challenges

- Existing methods:
  - The most common approach to modeling the nonstationary Matérn covariance is to divide the nonstationary field into subregions where the parameters are assumed to be stationary, and then construct the spatially varying parameter set using kernel smoothing.

[a] Risser MD, Calder CA (2017). "Local Likelihood Estimation for Covariance Functions with Spatially-Varying Parameters: The convoSPAT Package for R." Journal of Statistical Software, 81(14), 1–32. doi:10.18637/jss.v081.i14.

Existing Methods and Challenges

- Existing methods:
  - The most common approach to modeling the nonstationary Matérn covariance is to divide the nonstationary field into subregions where the parameters are assumed to be stationary, and then construct the spatially varying parameter set using kernel smoothing.
  - `ConvoSPAT`[a] is an R package widely used for modeling this covariance function.

[a] Risser MD, Calder CA (2017). "Local Likelihood Estimation for Covariance Functions with Spatially-Varying Parameters: The convoSPAT Package for R." Journal of Statistical Software, 81(14), 1–32. doi:10.18637/jss.v081.i14.

Existing Methods and Challenges

- Existing methods:
  - The most common approach to modeling the nonstationary Matérn covariance is to divide the nonstationary field into subregions where the parameters are assumed to be stationary, and then construct the spatially varying parameter set using kernel smoothing.
  - `ConvoSPAT`[a] is an R package widely used for modeling this covariance function.
- Challenges:

[a] Risser MD, Calder CA (2017). "Local Likelihood Estimation for Covariance Functions with Spatially-Varying Parameters: The convoSPAT Package for R." Journal of Statistical Software, 81(14), 1–32. doi:10.18637/jss.v081.i14.

Existing Methods and Challenges

- Existing methods:
  - The most common approach to modeling the nonstationary Matérn covariance is to divide the nonstationary field into subregions where the parameters are assumed to be stationary, and then construct the spatially varying parameter set using kernel smoothing.
  - `ConvoSPAT`[a] is an R package widely used for modeling this covariance function.
- Challenges:
  - `ConvoSPAT` cannot handle large datasets.

[a] Risser MD, Calder CA (2017). "Local Likelihood Estimation for Covariance Functions with Spatially-Varying Parameters: The convoSPAT Package for R." Journal of Statistical Software, 81(14), 1–32. doi:10.18637/jss.v081.i14.

Existing Methods and Challenges

- Existing methods:
  - The most common approach to modeling the nonstationary Matérn covariance is to divide the nonstationary field into subregions where the parameters are assumed to be stationary, and then construct the spatially varying parameter set using kernel smoothing.
  - `ConvoSPAT`[a] is an R package widely used for modeling this covariance function.
- Challenges:
  - `ConvoSPAT` cannot handle large datasets.
  - The subregion selection is subjective and not data-driven.

[a] Risser MD, Calder CA (2017). "Local Likelihood Estimation for Covariance Functions with Spatially-Varying Parameters: The convoSPAT Package for R." Journal of Statistical Software, 81(14), 1–32. doi:10.18637/jss.v081.i14.

Existing Methods and Challenges

- Existing methods:
  - The most common approach to modeling the nonstationary Matérn covariance is to divide the nonstationary field into subregions where the parameters are assumed to be stationary, and then construct the spatially varying parameter set using kernel smoothing.
  - ConvoSPAT[a] is an R package widely used for modeling this covariance function.
- Challenges:
  - ConvoSPAT cannot handle large datasets.
  - The subregion selection is subjective and not data-driven.
- This project addresses the aforementioned challenges by

[a] Risser MD, Calder CA (2017). "Local Likelihood Estimation for Covariance Functions with Spatially-Varying Parameters: The convoSPAT Package for R." Journal of Statistical Software, 81(14), 1–32. doi:10.18637/jss.v081.i14.

Existing Methods and Challenges

- Existing methods:
  - The most common approach to modeling the nonstationary Matérn covariance is to divide the nonstationary field into subregions where the parameters are assumed to be stationary, and then construct the spatially varying parameter set using kernel smoothing.
  - `ConvoSPAT`[a] is an R package widely used for modeling this covariance function.
- Challenges:
  - `ConvoSPAT` cannot handle large datasets.
  - The subregion selection is subjective and not data-driven.
- This project addresses the aforementioned challenges by
  - Implementing the nonstationary covariance function in `ExaGeoStat` to handle large datasets.

[a] Risser MD, Calder CA (2017). "Local Likelihood Estimation for Covariance Functions with Spatially-Varying Parameters: The convoSPAT Package for R." Journal of Statistical Software, 81(14), 1–32. doi:10.18637/jss.v081.i14.

Existing Methods and Challenges

- Existing methods:
  - The most common approach to modeling the nonstationary Matérn covariance is to divide the nonstationary field into subregions where the parameters are assumed to be stationary, and then construct the spatially varying parameter set using kernel smoothing.
  - `ConvoSPAT`[a] is an R package widely used for modeling this covariance function.
- Challenges:
  - `ConvoSPAT` cannot handle large datasets.
  - The subregion selection is subjective and not data-driven.
- This project addresses the aforementioned challenges by
  - Implementing the nonstationary covariance function in `ExaGeoStat` to handle large datasets.
  - Developing a CNN-based cluster mechanism for data-driven subregion selection, where the model also serves as a stationary-nonstationary classifier.

[a] Risser MD, Calder CA (2017). "Local Likelihood Estimation for Covariance Functions with Spatially-Varying Parameters: The convoSPAT Package for R." Journal of Statistical Software, 81(14), 1–32. doi:10.18637/jss.v081.i14.

Maximum Likelihood Parameter Estimation with ExaGeoStat

- This work employs the `ExaGeoStat`[a] software to facilitate scalable parameter estimation, designed explicitly for modeling large-scale geospatial data circumventing the need for approximated likelihoods.

[a] Abdulah, S., H. Ltaief, Y. Sun, M. G. Genton, and D. E. Keyes (2018). ExaGeoStat: A high performance unified software for geostatistics on manycore systems. IEEE Transactions on Parallel and Distributed Systems 29 (12), 2771–2784.

Maximum Likelihood Parameter Estimation with ExaGeoStat

- This work employs the `ExaGeoStat`[a] software to facilitate scalable parameter estimation, designed explicitly for modeling large-scale geospatial data circumventing the need for approximated likelihoods.

- `ExaGeoStat` estimates the statistical parameters of a given geospatial domain in parallel and at large-scale. `ExaGeoStat` has distributed memory support, enabling one to perform parallel computing.

[a] Abdulah, S., H. Ltaief, Y. Sun, M. G. Genton, and D. E. Keyes (2018). ExaGeoStat: A high performance unified software for geostatistics on manycore systems. IEEE Transactions on Parallel and Distributed Systems 29 (12), 2771–2784.

Kernel Smoothing

- To tackle the issue of many unknown parameters, the kernel smoothing approach is implemented in `ExaGeoStat` to represent the spatially varying parameters through smooth functions.

Kernel Smoothing

- To tackle the issue of many unknown parameters, the kernel smoothing approach is implemented in `ExaGeoStat` to represent the spatially varying parameters through smooth functions.
- The spatially varying parameters are modeled as follows:

Kernel Smoothing

- To tackle the issue of many unknown parameters, the kernel smoothing approach is implemented in `ExaGeoStat` to represent the spatially varying parameters through smooth functions.
- The spatially varying parameters are modeled as follows:
- Let K represent the total number of subregions and $\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_K$ be the anchor locations to those subregions, which are the centers of different subregions. The spatially varying parameters are defined as:

$$\boldsymbol{\theta}(\boldsymbol{s}_i) = \sum_{k=1}^{K} W(\boldsymbol{s}_i, \boldsymbol{S}_k)\, \boldsymbol{\theta}(\boldsymbol{S}_k).$$

Kernel Smoothing

- Let K represent the total number of subregions and $\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_k$ be the anchor locations to those subregions, which are the centers of different subregions. The spatially varying parameters are defined as:

$$\boldsymbol{\theta}\left(\boldsymbol{s}_i\right) = \sum_{k=1}^{K} W\left(\boldsymbol{s}_i, \boldsymbol{S}_k\right) \boldsymbol{\theta}\left(\boldsymbol{S}_k\right),$$

- For any $\mathbf{s}_i \in D \subset \mathbb{R}^d$, where $\mathbf{S}_k$'s are anchor locations and $\boldsymbol{\theta}(S_k)$'s are the parameter values at those anchor locations.

$$W\left(\mathbf{s}_i, \boldsymbol{S}_k\right) = \frac{K\left(\mathbf{s}_i, \boldsymbol{S}_k\right)}{\sum_k K\left(\mathbf{s}_i, \boldsymbol{S}_k\right)}$$

and the Gaussian kernel $K\left(\mathbf{s}_i, \boldsymbol{S}_k\right) = exp\left(\left(-\|\mathbf{s}_i - \boldsymbol{S}_k\|^2\right)/2h\right)$, where h is the bandwidth.

ConvNet for Nonstationarity Classification

- This project propose a Convolutional Neural Network based classifier (**ConvNet**) which can be used to distinguish between stationary and nonstationary random fields.

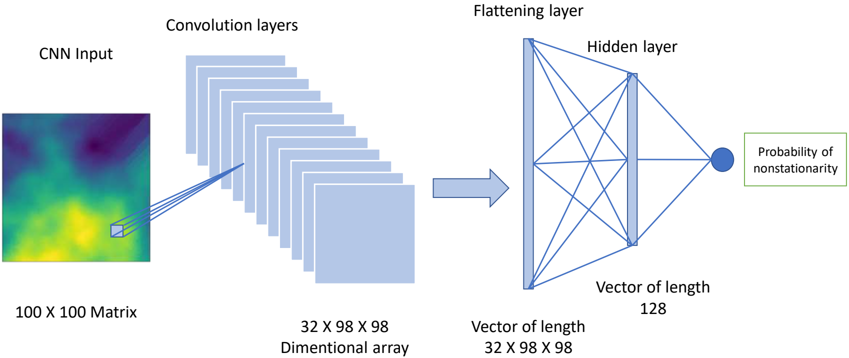


Figure: The structure of the CNN model, the flatten layer victories the CNN layer output, the final layer with softmax activation provides the probability of a particular class.

The **ConvNet** Training Phase

- Stationary and nonstationary datasets with different sizes are simulated for the CNN model training.

The **ConvNet** Training Phase

- Stationary and nonstationary datasets with different sizes are simulated for the CNN model training.
- Stationary datasets are generated using the stationary Matérn covariance while the nonstationary datasets are generated with the nonstationary Matérn covariance function.

The **ConvNet** Training Phase

- Stationary and nonstationary datasets with different sizes are simulated for the CNN model training.
- Stationary datasets are generated using the stationary Matérn covariance while the nonstationary datasets are generated with the nonstationary Matérn covariance function.
- To ensure generalized parameter settings different nonlinear functions representing the parameters in $\boldsymbol{\theta}$ are chosen.

The **ConvNet** Training Phase

- Stationary and nonstationary datasets with different sizes are simulated for the CNN model training.
- Stationary datasets are generated using the stationary Matérn covariance while the nonstationary datasets are generated with the nonstationary Matérn covariance function.
- To ensure generalized parameter settings different nonlinear functions representing the parameters in $\boldsymbol{\theta}$ are chosen.
- A simple pre-processing transformation is followed to transform the data into a regular 100×100 grid.

ConvNet for Subregion Selection

- A Clustering approach is considered here for selection of subregions.

ConvNet for Subregion Selection

- A Clustering approach is considered here for selection of subregions.
- The whole nonstationary region is first divided into K clusters.

ConvNet for Subregion Selection

- A Clustering approach is considered here for selection of subregions.
- The whole nonstationary region is first divided into K clusters.
- Each of the clusters/subregions are then pre-processed and transformed into 100×100 grid to pass to **ConvNet**.

ConvNet for Subregion Selection

- A Clustering approach is considered here for selection of subregions.
- The whole nonstationary region is first divided into K clusters.
- Each of the clusters/subregions are then pre-processed and transformed into 100×100 grid to pass to **ConvNet**.
- The **ConvNet** provides a nonstationarity index score for each.

ConvNet for Subregion Selection

- A Clustering approach is considered here for selection of subregions.
- The whole nonstationary region is first divided into K clusters.
- Each of the clusters/subregions are then pre-processed and transformed into 100×100 grid to pass to **ConvNet**.
- The **ConvNet** provides a nonstationarity index score for each.
- This process is followed for B number of iterations.

ConvNet for Subregion Selection

- A Clustering approach is considered here for selection of subregions.
- The whole nonstationary region is first divided into K clusters.
- Each of the clusters/subregions are then pre-processed and transformed into 100×100 grid to pass to **ConvNet**.
- The **ConvNet** provides a nonstationarity index score for each.
- This process is followed for B number of iterations.
- The optimal cluster is then chosen for which the combined nonstationarity index as obtained from the **ConvNet** model is the smallest.

Simulation Studies : Performance of The **ConvNet** Model

The **ConvNet** model obtains 97% and 98% accuracy respectively in successfully identifying the stationary and nonstationary random fields on test data.

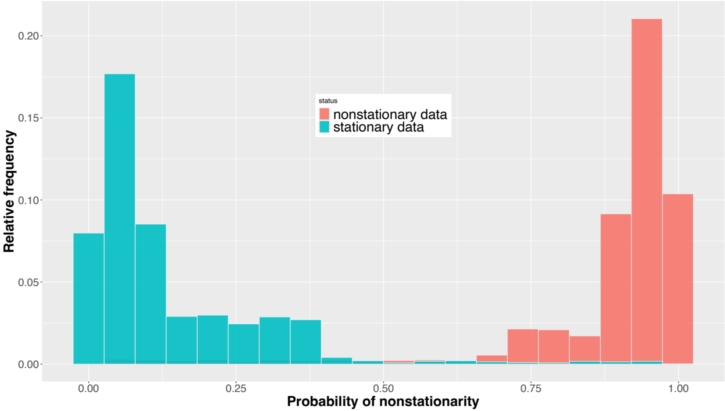


Figure: Histogram of the nonstationarity index for stationary testing data and nonstationary testing data.

Simulation Study: Parameter Estimation

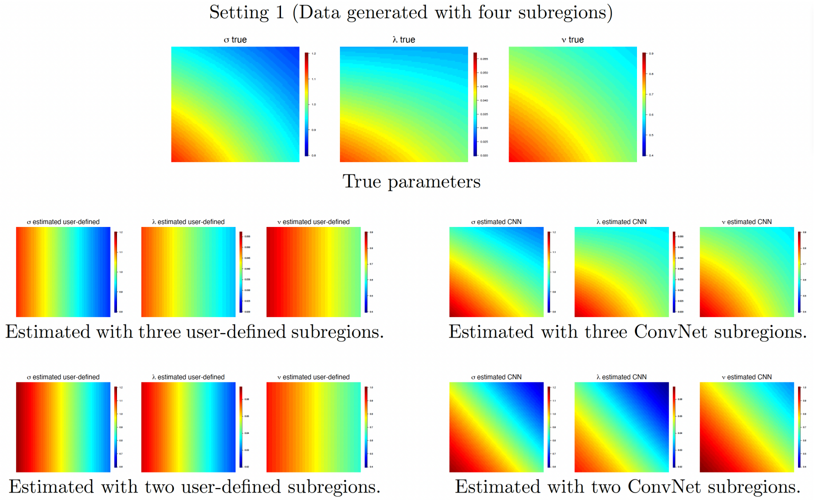


Figure: Heatmaps for true parameters and the average of the estimated parameters for different simulation scenarios.

Simulation Study: Parameter Estimation

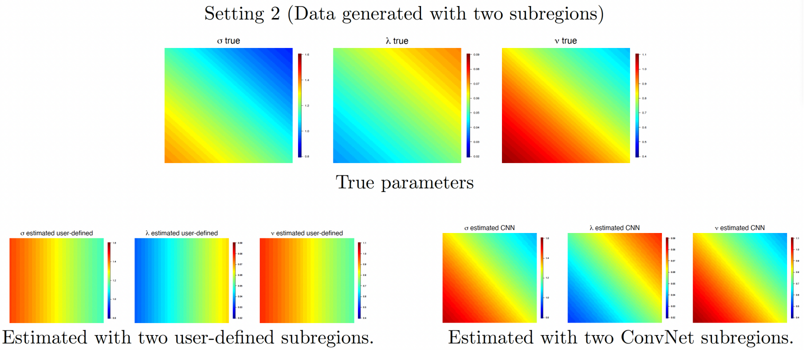
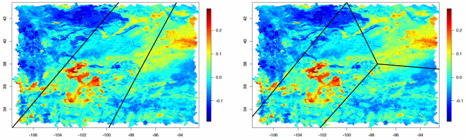


Figure: Heatmaps for true parameters and the average of the estimated parameters for different simulation scenarios.

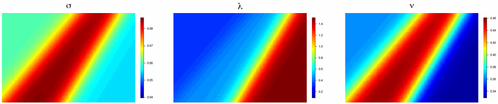
Soil Moisture Data Application

This method is applied to analyze soil moisture content data across the Mississippi Basin region in the United States with 200,000 locations. Based on AIC the performing model came out to be the three-subregion model.

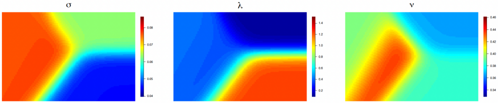
Soil Moisture Data



The black lines show the splits for three and four subregions, respectively.



Estimated with three ConvNet subregions.



Estimated with four ConvNet subregions.

Conclusion

- This project present **ConvNet**, an approach that is able to distinguish the stationary and nonstationary regions.

Conclusion

- This project present **ConvNet**, an approach that is able to distinguish the stationary and nonstationary regions.
- **ConvNet** is then coupled with a clustering mechanism to identify the stationary subregions in a given geospatial region.

Conclusion

- This project present **ConvNet**, an approach that is able to distinguish the stationary and nonstationary regions.
- **ConvNet** is then coupled with a clustering mechanism to identify the stationary subregions in a given geospatial region.
- `ExaGeoStat` framework is used along with the clustering mechanism for exact large-scale implementation of nonstationary Matérn kernel.

Project 4: Spatial Normalizing Flows for Nonstationary Gaussian Processes

Background

- Modeling complex environmental phenomena often involves selecting nonstationary and anisotropic covariance structures such as the non-stationary Matérn covariance as discussed in previous section.

Background

- Modeling complex environmental phenomena often involves selecting nonstationary and anisotropic covariance structures such as the non-stationary Matérn covariance as discussed in previous section.
- However, the choise of the covarince can pose challenges when the underlying spatial process is not well-understood.

Background

- Modeling complex environmental phenomena often involves selecting nonstationary and anisotropic covariance structures such as the non-stationary Matérn covariance as discussed in previous section.
- However, the choise of the covarince can pose challenges when the underlying spatial process is not well-understood.
- An alternative approach to model these intricate structures involves deformation of the spatial domain with the idea that a process that is highly nonstationary or anisotropic on the original domain could be stationary and isotropic on the warped domain.

Existing Methods and Challenges

- Key previous works:

[a] Sampson, P. D. and Guttorp, P. (1992) Nonparametric estimation of nonstationary spatial covariance structure. Journal of the American Statistical Association 87(417), 108–119.

[b] Zammit-Mangion, A., Ng, T. L. J., Vu, Q. and Filippone, M. (2021) Deep compositional spatial models. Journal of the American Statistical Association pp. 1–22.

Existing Methods and Challenges

- Key previous works:
  - Warping using multi-dimentional scaling[a].

[a] Sampson, P. D. and Guttorp, P. (1992) Nonparametric estimation of nonstationary spatial covariance structure. Journal of the American Statistical Association 87(417), 108–119.

[b] Zammit-Mangion, A., Ng, T. L. J., Vu, Q. and Filippone, M. (2021) Deep compositional spatial models. Journal of the American Statistical Association pp. 1–22.

Existing Methods and Challenges

- Key previous works:
  - Warping using multi-dimentional scaling[a].
  - Spatial Input-Warped Gaussian Processes[b]: a collection of simple warping units to model the deformation in deep learning framework.

[a] Sampson, P. D. and Guttorp, P. (1992) Nonparametric estimation of nonstationary spatial covariance structure. Journal of the American Statistical Association 87(417), 108–119.

[b] Zammit-Mangion, A., Ng, T. L. J., Vu, Q. and Filippone, M. (2021) Deep compositional spatial models. Journal of the American Statistical Association pp. 1–22.

Existing Methods and Challenges

- Key previous works:
  - Warping using multi-dimentional scaling[a].
  - Spatial Input-Warped Gaussian Processes[b]: a collection of simple warping units to model the deformation in deep learning framework.
- Challenges:

[a] Sampson, P. D. and Guttorp, P. (1992) Nonparametric estimation of nonstationary spatial covariance structure. Journal of the American Statistical Association 87(417), 108–119.

[b] Zammit-Mangion, A., Ng, T. L. J., Vu, Q. and Filippone, M. (2021) Deep compositional spatial models. Journal of the American Statistical Association pp. 1–22.

Existing Methods and Challenges

- Key previous works:
  - Warping using multi-dimentional scaling[a].
  - Spatial Input-Warped Gaussian Processes[b]: a collection of simple warping units to model the deformation in deep learning framework.
- Challenges:
  - Choise of warping functions are always tricky.

[a] Sampson, P. D. and Guttorp, P. (1992) Nonparametric estimation of nonstationary spatial covariance structure. Journal of the American Statistical Association 87(417), 108–119.

[b] Zammit-Mangion, A., Ng, T. L. J., Vu, Q. and Filippone, M. (2021) Deep compositional spatial models. Journal of the American Statistical Association pp. 1–22.

Existing Methods and Challenges

- Key previous works:
  - Warping using multi-dimentional scaling[a].
  - Spatial Input-Warped Gaussian Processes[b]: a collection of simple warping units to model the deformation in deep learning framework.
- Challenges:
  - Choise of warping functions are always tricky.
  - The work of Zammit-Mangion et al. (2021)[b] also has some limitations:The individual warping units as in Zammit-Mangion et al. (2021)[b] are rigid and is only suitable for two-dimentional spatial processes.

[a] Sampson, P. D. and Guttorp, P. (1992) Nonparametric estimation of nonstationary spatial covariance structure. Journal of the American Statistical Association 87(417), 108–119.

[b] Zammit-Mangion, A., Ng, T. L. J., Vu, Q. and Filippone, M. (2021) Deep compositional spatial models. Journal of the American Statistical Association pp. 1–22.

Existing Methods and Challenges

- Key previous works:
  - Warping using multi-dimentional scaling[a].
  - Spatial Input-Warped Gaussian Processes[b]: a collection of simple warping units to model the deformation in deep learning framework.
- Challenges:
  - Choise of warping functions are always tricky.
  - The work of Zammit-Mangion et al. (2021)[b] also has some limitations:The individual warping units as in Zammit-Mangion et al. (2021)[b] are rigid and is only suitable for two-dimentional spatial processes.
- This work explores the novel application of Neural Autoregressive Flows (NAFs) to model spatial warping.

[a] Sampson, P. D. and Guttorp, P. (1992) Nonparametric estimation of nonstationary spatial covariance structure. Journal of the American Statistical Association 87(417), 108–119.

[b] Zammit-Mangion, A., Ng, T. L. J., Vu, Q. and Filippone, M. (2021) Deep compositional spatial models. Journal of the American Statistical Association pp. 1–22.

The normalizing flow (NF) is an invertible function typically used to model transformations of random variables. We employ the normalizing flow to warp the spatial locations. A special type of NF is the auto-regressive flow (AF) which can be viewed as a triangular map $\boldsymbol{T}(.)$ where

$$T^{(k)}(s_1, \ldots, s_k) = S^{(k)}(s_k; \gamma_k(s_1, \ldots, s_{k-1}; \boldsymbol{\vartheta}_k)),$$

where, $\{s_1, \ldots, s_k\} \subseteq \mathbf{s}, k = 1, \ldots, d$, and γ_k is the k-th conditional network with parameters $\boldsymbol{\vartheta}_k$. The conditional network is a multivariate mapping that takes inputs $s_1, \ldots, s_{k-1}$ and gives outputs in the parameter space of $S^{(k)}$, i,e., $\gamma_k : D^{k-1} \rightarrow \mathbb{R}^{m_k}$, where m_k is the number of parameters that parameterize $S^{(k)}$.

Normalizing Flows

This work focuses on the class of Neural Autoregressive Flows (NAFs), proposed by Huang et al. (2018)[a]. We choose a class of functions commonly referred to as Deep Sigmoidal Flows (DSF). In this class, one single layer has $m_k = 3M$ parameters, where $M \geq 1$, and the k-th component has the form

$$S^{(k)}(s_k; \gamma_k) = \sigma^{-1}\left(\boldsymbol{w}_k^T \sigma(\boldsymbol{a}_k s_k + \boldsymbol{b}_k)\right),$$

where $\sigma^{-1}(\cdot)$ is the logit function and parameters $\gamma_k \equiv (\boldsymbol{w}_k^T, \boldsymbol{a}_k^T, \boldsymbol{b}_k^T)^T$ are neural network functions of length M with $\sum_{i=1}^{M} w_{ki} = 1$. This construction ensures monotonicity of the function $S^{(k)}(\cdot)$ and hence of the function $T^{(k)}(\cdot)$. Ultimately, this construction ensures that the multivariate mapping $\boldsymbol{T}(s_1, \ldots, s_k)$ will be injective.

[a] Huang, C.-W., D. Krueger, A. Lacoste, and A. Courville (2018). Neural autoregressive flows. In International Conference on Machine Learning, pp. 2078–2087. PMLR.

Deep Dense Sigmoidal Flows

- A multi layer perceptron (MLP) can be obtained by stacking multiple layers of these DSFs together. However, this architecture contains a bottleneck as the output of each layer has only one node.

Deep Dense Sigmoidal Flows

- A multi layer perceptron (MLP) can be obtained by stacking multiple layers of these DSFs together. However, this architecture contains a bottleneck as the output of each layer has only one node.
- The alternative framework is the Deep Dense Sigmoidal Flows (DDSFs). In this class, layer l of the flow has $m_k = M_l^2 + M_l M_{l-1} + 2M_l$ parameters, where $M_{l-1}, M_l \geq 1$ with

$$\begin{aligned} h_k^1 &= \sigma^{-1}\left(\boldsymbol{w}_k^1 \sigma(\boldsymbol{a}_k^1 \odot (\boldsymbol{u}_k^1 s_k) + \boldsymbol{b}_k^1)\right), \\ h_k^{l-1} &= \sigma^{-1}\left(\boldsymbol{W}_k^l \sigma(\boldsymbol{a}_k^l \odot (\boldsymbol{U}_k^l \boldsymbol{h}_k^{l-1}) + \boldsymbol{b}_k^l)\right), \quad l = 2, \ldots, L-1, \\ h_k^L &= \sigma^{-1}\left(\boldsymbol{w}_k^L \sigma(\boldsymbol{a}_k^L \odot (\boldsymbol{u}_k^L h_k^{L-1}) + \boldsymbol{b}_k^1)\right), \end{aligned}$$

$$S^{(k)}(s_k; \gamma_k) = h_k^L,$$

where, $\sum_{j=1}^{M_l} w_{kij} = 1$ and $\sum_{j=1}^{M_l} u_{kij} = 1$ corresponding to the i-th row of matrices $\boldsymbol{W}_k^l, \boldsymbol{U}_k^l$. Similar to DSF the parameters here are defined through γ_k.

Binary Masking

- The vector of locations $\mathbf{s}$ is passed through a single feed forward network to obtain γ_k. To enforce the autoregressive property, the feed forward function is modified by introducing binary mask $\mathbf{M_W}$:

$$\gamma_k = \boldsymbol{g}\left(\mathbf{b} + \left(\mathbf{W} \odot \mathbf{M_W}\right)\mathbf{s}\right)$$

Here, $\odot$ denotes elementwise multiplication, and the mask $\mathbf{M_W}$ ensure the autoregressive property.

Binary Masking

- The vector of locations **s** is passed through a single feed forward network to obtain γ_k. To enforce the autoregressive property, the feed forward function is modified by introducing binary mask $\mathbf{M_W}$:

$$\gamma_k = \boldsymbol{g}\left(\mathbf{b} + \left(\mathbf{W} \odot \mathbf{M_W}\right)\mathbf{s}\right)$$

  Here, $\odot$ denotes elementwise multiplication, and the mask $\mathbf{M_W}$ ensure the autoregressive property.

- Constraints on the maximum number of inputs to each hidden unit are encoded in the matrix masking the connections between input and hidden units:

$$M_{\mathbf{W}_{j,k}} = 1_{m(j) \geq k} = \begin{cases} 1 & \text{if } m(j) \geq k \\ 0 & \text{otherwise} \end{cases},$$

  for $k \in \{1, \ldots, D\}$ and $l \in \{1, \ldots, L\}$. Overall, the constraint is that the k^{th} output unit connects only to $\mathbf{s}_{<k}$ (not to $\mathbf{s}_{\geq k}$).

Loss Function and Training

- In this project, the aim is to model the nonstationary space through a stationary Gaussian process with a simple covariance structure such as the exponential covariance.

Loss Function and Training

- In this project, the aim is to model the nonstationary space through a stationary Gaussian process with a simple covariance structure such as the exponential covariance.
- A 2-stage training is employed to maximize the loss function.

Loss Function and Training

- In this project, the aim is to model the nonstationary space through a stationary Gaussian process with a simple covariance structure such as the exponential covariance.
- A 2-stage training is employed to maximize the loss function.
- In the first stage $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ is fixed, and $L(\mathbf{S}_{vec}, \boldsymbol{\theta}_0)$ is maximized based on the warped location $\mathbf{S}$.

Loss Function and Training

- In this project, the aim is to model the nonstationary space through a stationary Gaussian process with a simple covariance structure such as the exponential covariance.
- A 2-stage training is employed to maximize the loss function.
- In the first stage $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ is fixed, and $L(\mathbf{S}_{vec}, \boldsymbol{\theta}_0)$ is maximized based on the warped location $\mathbf{S}$.
- Training of the warping function is done through DDSFs.

Loss Function and Training

- In this project, the aim is to model the nonstationary space through a stationary Gaussian process with a simple covariance structure such as the exponential covariance.
- A 2-stage training is employed to maximize the loss function.
- In the first stage $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ is fixed, and $L(\mathbf{S}_{vec}, \boldsymbol{\theta}_0)$ is maximized based on the warped location $\mathbf{S}$.
- Training of the warping function is done through DDSFs.
- In the next stage the warping is fixed to $\mathbf{S}^{opt}$ that maximizes $L(\mathbf{S}^{opt}, \boldsymbol{\theta})$ and it is then maximized on $\boldsymbol{\theta}$, let us call the optimized parameters as $\boldsymbol{\theta}^{opt}$.

Loss Function and Training

- In this project, the aim is to model the nonstationary space through a stationary Gaussian process with a simple covariance structure such as the exponential covariance.
- A 2-stage training is employed to maximize the loss function.
- In the first stage $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ is fixed, and $L(\mathbf{S}_{vec}, \boldsymbol{\theta}_0)$ is maximized based on the warped location $\mathbf{S}$.
- Training of the warping function is done through DDSFs.
- In the next stage the warping is fixed to $\mathbf{S}^{opt}$ that maximizes $L(\mathbf{S}^{opt}, \boldsymbol{\theta})$ and it is then maximized on $\boldsymbol{\theta}$, let us call the optimized parameters as $\boldsymbol{\theta}^{opt}$.
- This process is repeated until $\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^{opt}\| < \psi$ for some small quantity ψ.

Simulation Studies

- Two one-dimentional simulations and one two-dimentional simulation is constructed to compare the model with other comparing models.

Simulation Studies

- Two one-dimentional simulations and one two-dimentional simulation is constructed to compare the model with other comparing models.
- One dimentional :
  Two cases are considered here on $G = [-0.5, 0.5]$, where the underlying processes are

Simulation Studies

- Two one-dimentional simulations and one two-dimentional simulation is constructed to compare the model with other comparing models.
- One dimentional :
  Two cases are considered here on $G = [-0.5, 0.5]$, where the underlying processes are
- $$Y^{(1,1)}(s) = \begin{cases} -0.5 & |s| > 0.2 \\ 0.5 & \text{otherwise} \end{cases},$$

Simulation Studies

- Two one-dimentional simulations and one two-dimentional simulation is constructed to compare the model with other comparing models.
- One dimentional :
  Two cases are considered here on $G = [-0.5, 0.5]$, where the underlying processes are
- $$Y^{(1,1)}(s) = \begin{cases} -0.5 & |s| > 0.2 \\ 0.5 & \text{otherwise} \end{cases},$$
- $$Y^{(1,2)}(s) = \begin{cases} \exp\left(4 + \frac{5}{2s(10s+5)}\right) & -0.5 < s < 0 \\ 1 & 0.2 \leq s \leq 0.3 \\ -1 & 0.3 < s \leq 0.4 \\ 0 & \text{otherwise.} \end{cases}$$

  with added Gaussian noise.

Simulation Studies

- Two one-dimentional simulations and one two-dimentional simulation is constructed to compare the model with other comparing models.

- Two dimentional :

  The warping function SWGIP as proposed in Zammit-Mangion et al. (2021)[a] is taken and compared with other approaches. Data is simulated in two dimensions from the underlying SIWGP on $G = [-0.5, 0.5]^2$, denoted as $Y^{(2,1)}(\cdot)$.

[a]Zammit-Mangion, A., Ng, T. L. J., Vu, Q. and Filippone, M. (2021) Deep compositional spatial models. Journal of the American Statistical Association pp. 1–22.

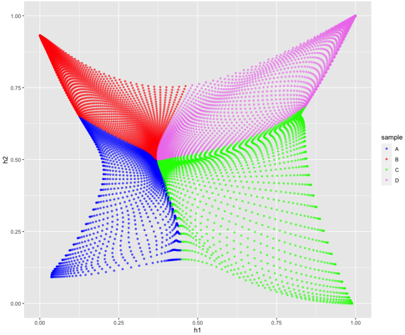
Simulation Studies: Results

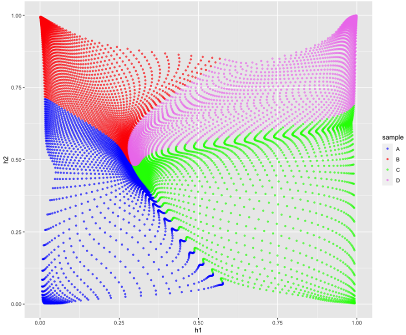
Table: Comparison on different simulation scenarios.

dataset	Models	**MSPE**	**PICP**	**MPIW**
$Y^{(1,1)}(s)$	$GP_{nonstat}$	0.033	0.93	0.47
	GP_{orig}	0.034	0.94	0.47
	GP_{warped}	0.033	0.94	0.45
$Y^{(1,2)}(s)$	$GP_{nonstat}$	0.021	0.90	0.522
	GP_{orig}	0.025	0.90	0.521
	GP_{warped}	0.016	0.93	0.423
$Y^{(2,1)}(\mathbf{s})$	$GP_{nonstat}$	0.348	0.96	2.822
	GP_{orig}	0.372	0.97	2.833
	GP_{warped}	0.063	0.97	1.171

The Estimated Warping in 2-dimension

The true (left) and the estimated (right) warpings for $Y^{(2,1)}(\cdot)$.





Conclusion

- Spatial normalizing flow represents a deep-learning-based model that excels in handling processes featuring highly complex nonstationary and anisotropic covariance structures.

Conclusion

- Spatial normalizing flow represents a deep-learning-based model that excels in handling processes featuring highly complex nonstationary and anisotropic covariance structures.
- Its construction inherently imposes a smooth injective constraint, thereby limiting the class of warpings and effectively mitigating the issue of "space-folding."

Conclusion

- Spatial normalizing flow represents a deep-learning-based model that excels in handling processes featuring highly complex nonstationary and anisotropic covariance structures.
- Its construction inherently imposes a smooth injective constraint, thereby limiting the class of warpings and effectively mitigating the issue of "space-folding."
- This method boasts extendability to higher dimensions, as the Neural Autoregressive Flows (NAF) architecture seamlessly accommodates multidimensional mappings.

Publications: Thesis Works

- **Pratik Nag**, Ying Sun, Brian J Reich. *Spatio-temporal DeepKriging for Interpolation and Probabilistic Forecasting*, Spatial Statistics (Oct, 2023), volume. 57, 100773, DOI 10.1016/j.spasta.2023.100773.
- **Pratik Nag**, Ying Sun, Brian J Reich. *Bivariate DeepKriging for Computationally Efficient Spatial Interpolation of Large-scale Wind Fields*, Arxiv : https://arxiv.org/abs/2307.08038.(In Revision at Technometrics).
- **Pratik Nag**, Sameh Abdulah, Yiping Hong, Ghulam Qadir, Ying Sun, Marc G. Genton. *Efficient Large-scale Nonstationary Spatial Covariance Function Estimation Using Convolutional Neural Networks*, Arxiv : https://arxiv.org/abs/2306.11487. (In Revision at Journal of Computational and Graphical Statistics (JCGS)).
- **Pratik Nag**, Andrew Zammit-Mangion, Ying Sun. *Spatial Normalizing Flows for Nonstationary Gaussian Processes*(in preparation).

Publications: Other Works

- **Pratik Nag**, Arnab Hazra, Rishikesh Yadav, Ying Sun. *Exploring the Efficacy of Statistical and Deep Learning Methods for Large Spatial Datasets: A Case Study*, JABES (2024). https://doi.org/10.1007/s13253-024-00602-4
- Sameh Abdulah, Faten Alamri, **Pratik Nag**, Ying Sun, Hatem Ltaief, David E. Keyes, Marc G. Genton, *The Second Competition on Spatial Statistics for Large Datasets*, J. data sci.(2022), 1-22, DOI 10.6339/22-JDS1076.
- Qinglei Cao, Sameh Abdulah, Rabab Alomairy,Yu Pei, **Pratik Nag**, George Bosilca, Jack Dongarra, Marc G. Genton,David E. Keyes, Hatem Ltaief, Ying Sun. *Reshaping Geostatistical Modeling and Prediction for Extreme-Scale Environmental Applications*, In2022 SC22: International Conference for High Performance Computing, Networking, Storage and Analysis (SC) 2022 Nov 3 (pp. 13-24). IEEE Computer Society.(Finalist for **Gordon Bell Prize**).
- **Pratik Nag**, Huixia Judy Wang, Ying Sun. *Indicator DeepKriging for Probabilistic Prediction of Spatial Processes*(in preparation).

Expressions of Gratitude

- I extend my deepest appreciation to Prof. Ying Sun for her unwavering support and invaluable mentorship throughout my academic journey.
- I am grateful to the members of my committee, Prof. Paula Moraga, Prof. Mohamed H. Elhoseiny, and Prof. Veronica Berrocal, for their willingness to be part of my Ph.D. defense evaluation.
- My sincere thanks go to the STAT faculty members, Prof. Marc Genton, Prof. Havard Rue, Prof. Raphael Huser, Prof. Hernando Ombao, and Prof. David Bolin, for their exceptional teaching.
- I am thankful for the collaboration and guidance provided by Prof. Andrew Zammit Mangion and Prof. Brian J. Reich, which has been instrumental in advancing our research.
- I deeply appreciate my colleagues and friends here at KAUST for their companionship and support throughout my Ph.D. journey.
- Last, but not the least, I am profoundly thankful to my parents and especially my friends back home for their unwavering presence and support during challenging times.