Spatial Neural Network with Transfer Learning

Hongjian Yang

Department of Statistics, North Carolina State University

1 Introduction

Spatial data is ubiquitous, encompassing a wide range of applications from environmental observations and biological measurements to more recent fields like computer vision. A critical challenge in the analysis of spatial data is spatial prediction, which involves estimating unobserved values based on nearby observations under the assumption of certain correlations. Among parametric algorithms, Kriging is particularly notable ((Matheron (1963))). Described as the best linear unbiased estimator (BLUE), Kriging employs a weighted average of nearby observations, with weights determined by a covariance function typically presumed to be stationary. However, this assumption does not hold in many real-world scenarios, such as data from satellites, monitoring stations, and urban streets, which tend to exhibit non-stationarity (Katzfuss (2013)). Moreover, Kriging faces computational challenges with large datasets, requiring the inversion of the covariance matrix, an operation with a computational complexity of $O(N^3)$ (Chen et al. (2020)).

A variety of algorithms have been proposed to address the issues highlighted above. Mao et al. (2022) introduced a model-free method for handling non-stationary spatial data, presenting a significant advancement in the field. Another popular alternative to Kriging is the Vecchia approximation, which simplifies the full Gaussian distribution by conditioning on neighboring values (Vecchia (1988)). Drawing on the concept of neighbor-based approximation, Wang et al. (2019) developed a nearest neighbor neural network specifically for spatial prediction. Building on these ideas, Chen et al. (2020) introduced the innovative Deep Kriging framework, which combines spatial basis functions with deep neural networks to model any spatial processes effectively.

While deep learning approaches have demonstrated significant promise in approximating spatial surfaces, they typically require a substantial amount of training data. Transfer learning emerges as an effective solution to this challenge. For instance, He et al. (2019) employed transfer learning to leverage a neural network pre-trained on the ImageNet dataset for Hyperspectral Image Classification (HSI). Similarly, Zhang and Liu (2012) utilized transfer

learning to adapt to a common latent representation for writer adaptation. Both the HSI and writer adaptation scenarios, which are limited by small target sample sizes, benefit from the integration of external information from larger datasets.

Spatial applications often grapple with the challenge of small sample sizes. For instance, in estimating $PM_{2.5}$ concentrations, only about 70 high-quality monitoring stations are available throughout North Carolina, and approximately 200 in California (Yang et al. (2023)). Despite this, a large volume of relatively low-quality data exists in these states. Since the spatial distribution of $PM_{2.5}$ tends to be stable, integrating data from these abundant but lower-quality stations could significantly enhance estimation accuracy. Drawing inspiration from the works of He et al. (2019) and Chen et al. (2020), this paper proposes a neural network-based transfer learning method that utilizes external information to improve spatial predictions in datasets with limited target data.

2 Method and Theoretical Properties

Let Y_i be the observation at spatial location $\mathbf{s}_i = (s_{i1}, s_{i2})$ for $i \in \{1, ..., n\}$. This paper assumes

$$Y_i = f(\mathbf{s}_i; \boldsymbol{\theta}) + \varepsilon_i, \tag{1}$$

where f is a spatial process that depends on parameters $\boldsymbol{\theta}$ and $\varepsilon_i \overset{iid}{\sim} \text{Normal}(0, \tau^2)$ is error.

The spatial process is modelled using a feed-forward neural network (FFNN) with input $\mathbf{s}_i$. The FFNN has two stages: in the first stage we deform the spatial coordinates using Radial Basis Function (RBF), and in the second stage the weights are applied to capture the underlying spatial strcture. Below, we describe the model with a single hidden layer in the first stage, and with seven hidden layers of 100 neurons in the second stage.

Following Chen et al. (2020), we first use an embedding layer expanding the spatial location into p known basis functions $K_1(\mathbf{s}), ..., K_p(\mathbf{s})$. In particular, we use the Wendland

basis function

$$\phi(d) = \begin{cases} (1-d)^6(35d^2+18d+3)/3, & d \in [0,1] \\ 0, & \text{otherwise.} \end{cases}$$

where d is the Euclidean distance between observations and knots. Adopting the idea in Nychka et al. (2015), we use a multi-resolution with the knots arranged on a rectangular grid. In particular, we used four level of resolutions, and at each level, let $u_i, i = 1, 2, 3, 4$ be a rectangular grid of points. The basis function is defined as $\phi^*(\mathbf{s}) = \phi(d) = \phi(||\mathbf{s} - u_i||)$. After the first stage, we have a basis function representation of $\mathbf{x} \in \mathbb{R}^{139}$.

The second stage of the neural network is defined as follows:

$$\begin{aligned} \mathbf{h}_1 &= \text{ReLU}(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \\ &\vdots \\ \mathbf{h}_7 &= \text{ReLU}(\mathbf{W}_7\mathbf{h}_6 + \mathbf{b}_7) \\ y &= \mathbf{W}_8\mathbf{h}_3 + b_8 \end{aligned}$$

where:

- $\mathbf{W}_1 \in \mathbb{R}^{100\times139}, \mathbf{b}_1 \in \mathbb{R}^{100}$ are the weights and biases of the first hidden layer.
- $\mathbf{W}_2 \dots \mathbf{W}_7 \in \mathbb{R}^{100\times100}, \mathbf{b}_2 \dots \mathbf{b}_7 \in \mathbb{R}^{100}$ are the weights and biases of the second and third hidden layers.
- $\mathbf{W}_8 \in \mathbb{R}^{1\times100}, b_8 \in \mathbb{R}$ are the weights and bias of the output layer.
- $\text{ReLU}(\cdot)$ is the Rectified Linear Unit activation function.
- $y \in \mathbb{R}$ is the output of the network.

A visual illustration of the architecture of the two-stage neural network are provided in the supplementary materials.

The neural network above is first trained on a large external data, and all parameters are transferred to the target data set, since we assume the distribution of the covariates X are the same.

Many recent advancement of transfer learning in NLP and Computer Vision area shows that the method proposed could improve the estimation performance. For example, Segment Anything Model (Kirillov et al. (2023)) learns from a millions of images and is the state-of-the-art in segmentation task. The basic idea behind both Segment Anything Model and the proposed approach is that the neural network is learning latent representation of the training feature, and can decode the learned feature and adapt to new tasks quickly.

3 Simulation

To evaluate the performance of the proposed method above, a simulation study is carried out using both stationary and non-stationary data on a unit square $[0,1]^2$.

The stationary data is a Matern spatial process as defined by Lindgren et al. (2011), where the correlation between two spatial location $\mathbf{s}_i, \mathbf{s}_j$ is

$$C(\mathbf{s}_i, \mathbf{s}_j) = \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)}(\kappa||\mathbf{s}_i - \mathbf{s}_j||)^\nu K_\nu(\kappa||\mathbf{s}_i - \mathbf{s}_j||)$$

where K_ν is the modified Bessel function of the second kind with smoothness factor $\nu = 1$, and $\kappa = \frac{\sqrt{8}}{\rho}$ where $\rho = 0.2$ is the spatial range, and $\sigma^2 = 1$ is the spatial range. The nugget error $\epsilon_i = 0.01$ is the i.i.d noise.

The non-stationary process is defined in Chen et al. (2020) with $Y_i = sin\{30(\bar{\mathbf{s}_i} - 0.9)^4\}cos\{2(\bar{\mathbf{s}_i} - 0.9)\} + (\bar{\mathbf{s}_i} - 0.9)/2$, where $\mathbf{s}_i = (\mathbf{s}_{i1}, \mathbf{s}_{i2})$ and $\bar{\mathbf{s}_i} = \frac{\mathbf{s}_{i1}+\mathbf{s}_{i2}}{2}$. To distinguish these two processes, in this paper we define the stationary process as Y_{iS} and the non-stationary process as Y_{iN}. An independent nugget variance of $\nu^2 = 1e^{-6}$ is added to the non-stationary process. Figure 1 shows the spatial surface of sample realizations.

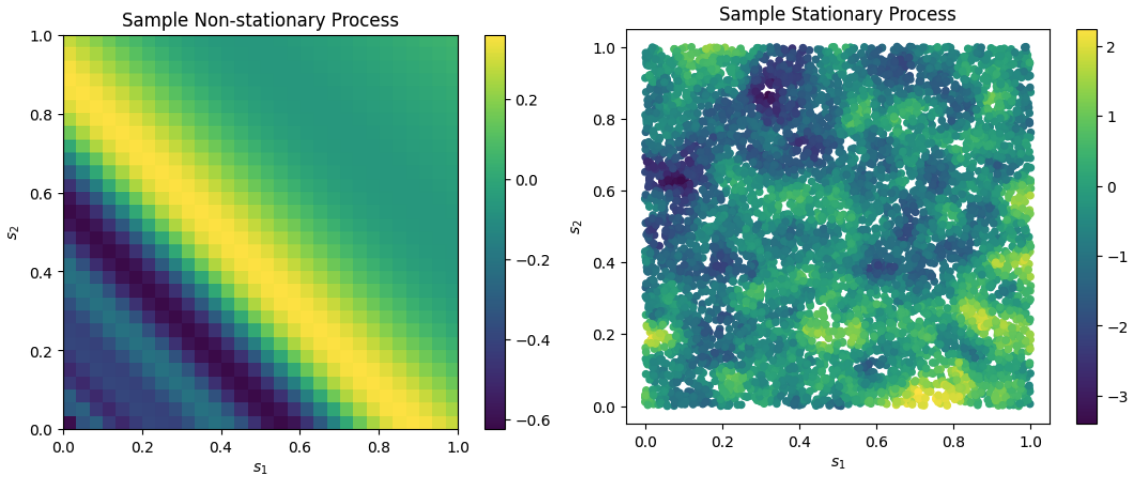

Figure 1: Simulation data

For each process, the source data set each consists of $N = 4900$ observations. We evaluate the performance on the target data set of $N = 25, 64, 100, 225$ observations. Figure 2 displays the MSE comparison of 1). source data pre-trained MSE on target data 2). target data set only, and 3). Kriging result on stationary data. Figure 3 compares the results on non-stationary data.

During the pre-training stage on external data, a total of 1500 epochs are used with a learning rate of 0.001. A trace plot with validation set is monitored to make sure the neural network has converged. During the tuning stage on target data, all parameters from the pre-training stage are updated. A total of 1000 epochs with a learning rate of 0.001 is used.

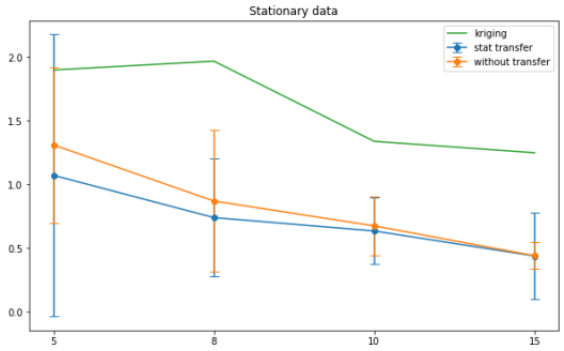

Figure 2: Stationary process MSE

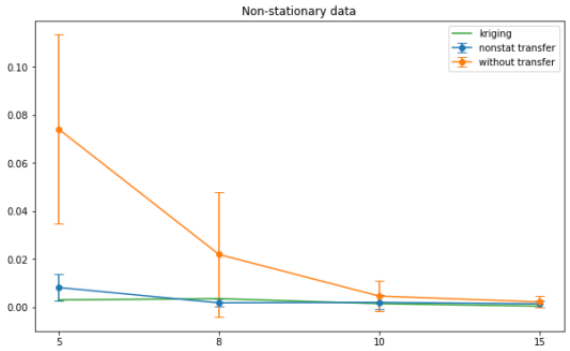

Figure 3: Non-stationary process MSE

4 Conclusion

As we can see from the figures above, for stationary data, the proposed approach outperforms both target only setting and the traditional Kriging approach. As target set gets larger, the proposed method and target only neural network converges.

For the non-stationary scenario, the proposed method significantly outperforms target only approach when the target sample size is less than 100, and similar to the stationary case, these two neural network converges when the sample size gets larger.

The proposed transfer learning approach is a first step towards spatial transfer learning. Following He et al. (2019), we tune all parameters in the model. One possible future extension is to add additional layers to the neural network, fix the first seven layers in the network, and only tune additional layers. Also, the fully connected neural network may not capture the spatial dependence well. It is possible to combine the 4N network proposed by Wang et al. (2019) or the graph neural network Klemmer et al. (2023).

References

Chen, W., Li, Y., Reich, B. J. and Sun, Y. (2020) Deepkriging: Spatially dependent deep neural networks for spatial prediction. *arXiv preprint arXiv:2007.11972*.

He, X., Chen, Y. and Ghamisi, P. (2019) Heterogeneous transfer learning for hyperspectral image classification based on convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, **58**, 3246–3263.

Katzfuss, M. (2013) Bayesian nonstationary spatial modeling for very large datasets. *Environmetrics*, **24**, 189–200.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y. et al. (2023) Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026.

Klemmer, K., Safir, N. S. and Neill, D. B. (2023) Positional encoder graph neural networks for geographic data. In *International Conference on Artificial Intelligence and Statistics*, 1379–1389. PMLR.

Lindgren, F., Rue, H. and Lindström, J. (2011) An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **73**, 423–498.

Mao, H., Martin, R. and Reich, B. J. (2022) Valid model-free spatial prediction. *Journal of the American Statistical Association*, 1–11.

Matheron, G. (1963) Principles of geostatistics. *Economic geology*, **58**, 1246–1266.

Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F. and Sain, S. (2015) A multiresolution gaussian process model for the analysis of large spatial datasets. *Journal of Computational and Graphical Statistics*, **24**, 579–599.

Vecchia, A. V. (1988) Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **50**, 297–312.

Wang, H., Guan, Y. and Reich, B. (2019) Nearest-neighbor neural networks for geostatistics. In *2019 international conference on data mining workshops (ICDMW)*, 196–205. IEEE.

Yang, H., Ruiz-Suarez, S., Reich, B. J., Guan, Y. and Rappold, A. G. (2023) A data-fusion approach to assessing the contribution of wildland fire smoke to fine particulate matter in california. *Remote Sensing*, **15**, 4246.

Zhang, X.-Y. and Liu, C.-L. (2012) Writer adaptation with style transfer mapping. *IEEE transactions on pattern analysis and machine intelligence*, **35**, 1773–1787.

5 Supplementary Materials

5.1 Neural Network Architecture