GeospaNN: An python package for geospatial neural networks

13 December 2024

Summary

Geostatistical models are widely used to analyse datasets with spatial information encountered frequently in the geosciences (climate, ecology, forestry). In geostatistics, the spatial linear mixed model (SPLMM)

$$Y = X\beta + \epsilon(s)$$

has long been the standard approach to model such data. SPLMM accounts for the fixed effects between observations Y and covariates X via the linear regression part, while accounting for the correlation across spatial locations s via the spatial error process $\epsilon(s)$. Typically, the spatial dependency embedded in $\epsilon(s)$ is modeled using a Gaussian Process, which provides a parsimonious and flexible solution to modeling spatial correlations and provide spatial predictions at new locations via kriging.

The linear regression part of SPLMM is restrictive. Recently, machine learning techniques like artificial neural networks have been increasingly incorporated into geostatistics to address complex and non-linear interactions among variables. However, few methods were originally designed for dependent data structures and cannot directly encode correlation. Even those modified to account for spatial dependencies, like adding a large number of spatial features as inputs to neural network, still face scalability limitations due to the expanded dimensionality from adding many features. To address these concerns, we introduce `geospaNN`, a Python software for analysis of geospatial data using neural networks. `geospaNN` implements NN-GLS, a novel adaptation of Neural Networks (NN) proposed in Zhan and Datta (2024). NN-GLS bridges ideas of geostatistical modeling and machine learning by embedding neural network with spatial mixed models, retaining the capacity to directly model spatial correlations. NN-GLS extends the SPLMM to non-linear scenarios by replacing $X\beta$ with $m(X)$, a non-linear function. NN-GLS simultaneously performs non-linear mean function estimation using a neural network and spatial predictions (including prediction intervals) using Gaussian Processes. For computational efficiency, `geospaNN` leverages the widely popular Nearest Neighbor Gaussian Process (NNGP) approximation (Datta et al. 2016). It is also, to our knowledge, the first Python package to implement NNGP for scalable covariance matrix computations in geostatistical mmodels. This is a standalone component of broad and independent utility.

Statement of Need

`geospaNN` is a Python package for geospatial analysis that uses NN-GLS, a novel and scalable class of neural networks explicitly designed to account for spatial correlation in the data. The package implements NN-GLS using PyTorch, an open-source library widely used for building machine learning models. As illustrated in Zhan and Datta (2024), `geospaNN` is effectively represented as a geographically-informed Graph Neural Network (GNN) by using NNGP covariance matrices. It can be embedded within the PyG (PyTorch Geometric) framework, which is designed for scalable execution of GNNs on irregular data structures like graphs. `geospaNN` is primarily intended for researchers and scientists in machine learning and spatial statistics, but can also be applied more generally to estimation and prediction tasks involving other data with dependency structures, like time-series. `geospaNN` provides user-friendly wrappers for data simulation, preprocessing, and model training, which significantly simplify the analytical pipeline.

The implementation of NNGP models within geospaNN is of independent importance. NNGP enables scalable covariance matrix inversions, which features extensively in geospatial models. There exists two widely popular R-packages spNNGP and BRISC for implementations of NNGP, but to our knowledge there is no Python implementation of NNGP. We thus offer an avenue to efficiently analyze massive geospatial datasets in Python using NNGP (linear models) and NN-GLS (non-linear models).

The goal of geospaNN is to provide a lightweight, efficient, and user-friendly machine learning tool for geospatial analysis that can directly model spatial correlation. According to simulations, the package can handle datasets with up to half a million observations in under an hour on a standard personal laptop (Macbook with an M1 8-core processor, 16 GB Unified RAM). A significant portion of geospaNN has already been used in articles such as Zhan and Datta (2024) and Heaton, Millane, and Rhodes (2024). In the future, we anticipate that geospaNN will play a significant role at the interface of machine learning and spatial statistics, serving as a foundation for both scientific and methodological explorations.

Statement of the field

The geospaNN Package

This section provides an overview of the geospaNN package, including the NN-GLS architecture and several technical details. For practical examples and detailed documentation, visit the geospaNN website at https://wentaozhan1998.github.io/geospaNN-doc. A vignette is also available on the website for detailed illustration of various usage of the package.

NN-GLS Overview

In a simple linear regression scenario, Gauss-Markov's Theorem states that when the data exhibits a correlation structure, generalized least squares (GLS) provides greater efficiency than ordinary least squares (OLS). For vanilla neural networks, it is typically implicitly assumed that the observations Y_i are not correlated, and mean squared error is used as the loss function for regression tasks:

$$Y = m(X) + \epsilon.$$

NNGLS considers the spatial model

$$Y = m(X) + \epsilon(s)$$

where m is a non-linear function to be estimated using a neural network, and the error process $\epsilon(s)$ was same as before, i.e., a Gaussian process for the spatial dependence and, possibly, a noise component. Given the observations Y, the covariates X, and the locations s as the input, NN-GLS estimates the mean function $m()$ and predicts Y at new locations. The ultimate goal of NN-GLS is to help understanding the complex relationship between observations and covariates and provide accurate geospatial prediction.

Let Σ be a model for the spatial error ϵ, then the well-known theory (Gauss-Markov theorem) of OLS and GLS from the statistical literature motivates the introduction of a GLS-style loss in neural networks:

$$L\big(m(\cdot)\big) = \frac{1}{n}\big(Y - m(X)\big)^\top \Sigma^{-1}\big(Y - m(X)\big).$$

However, minimizing this GLS-style loss in practice presents several challenges:

1. The covariance matrix Σ is based on a parametric covariance function, and the parameters are typically unknown.
2. Even if Σ is well-estimated, inverting Σ becomes computationally infeasible as the sample size n grows large.

3. Since the GLS loss is not additive across observations, mini-batching—an essential technique in modern deep neural networks—cannot be applied directly.

NN-GLS addresses these issues by introducing the Nearest Neighbor Gaussian Process (NNGP) to approximate Σ^{-1}, and it naturally equates to a specialized Graph Neural Network (GNN). In brief, NNGP is a nearest-neighbor-based approximation to a full Gaussian Process with a specific covariance structure (Datta 2022). On a finite sample, NNGP sparsify the covariance matrix, thus simplifying both likelihood computation and the GLS-style loss, addressing the three issues mentioned above.

Specifically, for the GLS loss function:

$$L\big(\hat{m}(\cdot)\big) = \frac{1}{n}\big(Y - \hat{m}(X)\big)^{\top}\Sigma^{-1}\big(Y - \hat{m}(X)\big) = \frac{1}{n}\big(Y^* - \hat{m}^*(X)\big)^{\top}\big(Y^* - \hat{m}^*(X)\big) = \frac{1}{n}\sum_{i=1}^{n}(Y_i^* - \hat{m}^*(X_i))^2,$$

where $Y^* = Q^{1/2}Y$ and $\hat{m}^*(X) = Q^{1/2}\hat{m}(X)$. The GLS loss returns to an additive form, allowing for mini-batching instead of full-batch training. For likelihood-based parameter estimation, `geospaNN` uses the `BRISC` R package as an efficient solution (Saha and Datta 2018).

In NN-GLS, we assume that the parameters of the covariance matrix is unknown. The spatial parameters θ and the mean function $\hat{m}$ (weights and biases of the neural network used to model m) are estimated iteratively, and training proceeds until the validation loss converges. Once estimation is complete, nearest-neighbor-based kriging is used to generate spatial predictions and prediction intervals at new locations. The whole procedure embeds the three core features of `geospaNN`, 1. Estimate the non-linear mean function by $\hat{m}$. 2. Estimate the spatial parameters by $\hat{\theta}$. 3. Predict the outcome at new locations by $\hat{Y}$.

NNGP and Other Features

In addition to estimation and prediction for the non-linear spatial mixed model using NN-GLS, `geospaNN` offers a standalone efficient implementation of the spatial linear mixed model using NNGP. Given an $n \times n$ covariance matrix Σ and a k-neighbor list (defaulting to nearest neighbors), our implementation guarantees $O(n)$ computational complexity for any approximate matrix products involving $\Sigma^{1/2}$, $\Sigma^{-1/2}$, and Σ^{-1}. The option of customizing the k-neighbor list is provided in case specially designed neighbor set is desired. NNGP is fundamental to several key scalable features in `geospaNN`, including spatial data simulation and kriging.

For simulation, `geospaNN` allow users to customize the spatial parameters and mean functions to generate Y, X, and s from the spatial non-linear mixed model. Users are allowed to customize the spatial coordinates to simulate under different context. This feature can be flexibly used to simulate a Gaussian process as a general spatial random effect by specifying a zero mean function. For kriging, instead of using the full observed set, only the nearest neighbors of each new spot in the observed set are used to compute the kriging weights. It has been shown in Zhan and Datta (2024) that the nearest neighbor kriging provide accurate prediction interval under various settings.

`geospaNN` offers user-friendly modules for essential machine learning tasks, including neural network (NN) architecture design, training log management, and result visualization. In addition to tracking validation loss, the training log also visualizes the evolution of spatial parameters, providing deeper insight into model convergence. For high-dimensional result interpretation, `geospaNN` incorporates partial dependence plots (PDPs) to illustrate the marginal effect of individual covariates on the outcome. When compared across multiple models, PDPs offer invaluable insights into the complex relationships between covariates and the predicted outcome.

As a special case of the spatial non-linear mixed model, SPLMM can also be solved by `geospaNN`. The implementation here is equivalent to the R-packages BRISC and should be an optimal choice for the python users if efficient SPLMM solution is wanted for large geospatial dataset.

All these functions are included explicitly in the package and can be called independently.

Discussion

The `geospaNN` package offers an efficient implementation of the geospatial neural networks, specifically, the NN-GLS approach proposed in Zhan and Datta (2024). NN-GLS embeds a neural network with the spatial mixed model, accounts for spatial correlation by replacing the original loss function with a GLS-style version. `geospaNN` is capable of performing various statistical tasks, including non-linear mean-function estimation, covariance parameter estimation, spatial prediction with uncertainty quantification. The NNGP approximation is fundamental to the scalable implementation of `geospaNN`. Moreover, due to the sparsity of the NNGP approximation, `geospaNN` can be seamlessly integrated into the framework of Graph Neural Networks (GNNs), opening up new possibilities for a wide range of advanced neural architectures. A promising future direction for `geospaNN` is to evolve into a general framework for geospatially-informed deep learning, where graph-based message-passing (convolution) can occur multiple times, with weights determined by spatial processes to maintain statistical interpretability. We are also planning to explore the extension of `geospaNN` towards other data types and distributions in the future.

Acknowledgements

This work is supported by National Institute of Environmental Health Sciences grant R01ES033739. The authors report there are no competing interests to declare.

References

Datta, Abhirup. 2022. "Nearest-Neighbor Sparse Cholesky Matrices in Spatial Statistics." *Wiley Interdisciplinary Reviews: Computational Statistics* 14 (5): e1574.

Datta, Abhirup, Sudipto Banerjee, Andrew O Finley, and Alan E Gelfand. 2016. "On Nearest-Neighbor Gaussian Process Models for Massive Spatial Data." *Wiley Interdisciplinary Reviews: Computational Statistics* 8 (5): 162–71.

Heaton, Matthew J, Andrew Millane, and Jake S Rhodes. 2024. "Adjusting for Spatial Correlation in Machine and Deep Learning." *arXiv Preprint arXiv:2410.04312*.

Saha, Arkajyoti, and Abhirup Datta. 2018. "BRISC: Bootstrap for Rapid Inference on Spatial Covariances." *Stat* 7 (1): e184.

Zhan, Wentao, and Abhirup Datta. 2024. "Neural Networks for Geospatial Data." *Journal of the American Statistical Association*, no. just-accepted: 1–21.