GeospaNN: An python package for geospatial neural networks

13 December 2024

Summary

In geographical science, datasets with spatial information are prevalent. In geostatistics, the spatial linear mixed model (SPLMM)

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

has long been the standard approach to account for the fixed effects between observations y and covariates x, as well as the spatial effects $\epsilon(s)$ across spatial locations s. Typically, the spatial dependency embedded in $\epsilon(s)$ is modeled using a Gaussian Process, which provides a parsimonious and efficient solution.

Recently, machine learning techniques have been incorporated into geostatistics to address increasingly complex interactions among variables, extending the SPLMM to non-linear scenarios by replacing $X\beta$ with $m(X)$. However, few methods were originally designed for dependent data structures. Even those modified to account for dependencies still face scalability limitations due to the restrictive covariance structures. To address these concerns, we introduce `geospaNN`, which resolves these issues through NN-GLS, a novel adaptation of Neural Networks (NN) proposed in Zhan and Datta (2024). `geospaNN` simultaneously performs mean function estimation and prediction (including prediction intervals). For computational efficiency, `geospaNN` leverages the Nearest Neighbor Gaussian Process (NNGP) approximation (Datta et al. 2016). It is also the first Python package to implement NNGP for scalable covariance matrix computation, benefiting other geospatial computation tools.

Statement of Need

`GeospaNN` is a Python package for geospatial analysis that uses NN-GLS, a novel extension 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 a geographically-informed Graph Neural Network (GNN). It can be embedded within the PyG (PyTorch Geometric) framework, which is designed for efficient 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 data with dependency structures. `geospaNN` provides user-friendly wrappers for data simulation, preprocessing, and model training, which significantly simplify the analytical pipeline.

The implementation of the NNGP approximation within `geospaNN` is of independent importance, enabling scalable covariance matrix inversion and enhancing relevant Python-based applications.

The goal of `geospaNN` is to provide a lightweight, efficient, and user-friendly machine learning tool for geospatial analysis. According to simulations, the package can handle datasets with up to half a million observations in under an hour on a standard personal laptop. 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, `geospaNN` is poised to play a significant role at the interface of machine learning and spatial statistics, serving as a foundation for both scientific and methodological explorations.

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 to illustrate typical 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 assumed that the observations Y_i are independent, and mean squared error is used as the loss function for regression tasks:

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

The OLS vs. GLS example motivates the introduction of a GLS-style loss:

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

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

1. The covariance matrix Σ is based on a parametric covariance function, which is 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). Mathematically, given a covariance matrix Σ, its inverse can be approximated as:

$$\Sigma^{-1} = Q = Q^{\top/2}Q^{1/2},$$

where $Q^{1/2}$ is a lower triangular sparse matrix. The j-th element in the i-th row of $Q^{1/2}$ is non-zero if and only if j is in the i-th k-nearest neighborhood, where k is a pre-specified neighborhood size. This approximation simplifies both likelihood computation and the GLS-style loss, addressing the issues mentioned above (issue 1, 2, and 3).

Specifically, for the GLS loss function:

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

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 covariance structure is unknown. The spatial parameters θ and the mean function $\hat{m}$ are estimated iteratively, and training proceeds until the validation loss converges. Once estimation is complete, nearest-neighbor-based kriging is used to generate predictions and confidence intervals at new locations.

NNGP and Other Features

In addition to estimation and prediction, `geospaNN` efficiently implements the NNGP approximation. 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}. NNGP is fundamental to several key scalable features in `geospaNN`, including spatial data simulation and kriging.

Discussion

The GeospaNN package offers an efficient implementation of the NN-GLS approach proposed in Zhan and Datta (2024). It accounts for spatial correlation by replacing the original loss function with a GLS-style version. GeospaNN is capable of performing various statistical tasks, including mean-function estimation, parameter estimation, prediction, and 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 machine learning 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 the spatial process 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.