geospaNN: A Python package for geospatial neural networks

Summary

Geostatistical models are widely used to analyse datasets with spatial information encountered frequently in the geosciences (climate, ecology, forestry, environmental sciences, and other research fields). In geostatistics, the spatial linear mixed effects 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.

Recently, machine learning techniques like Neural Networks (NNs) have been increasingly incorporated into geostatistics to address complex and non-linear interactions among variables. In this paper, we introduce `geospaNN`, a Python software for analysis of geospatial data. `geospaNN` implements a novel adaptation of NN and simultaneously performs non-linear mean function estimation using a NN and spatial predictions (including prediction intervals) using Gaussian processes. For computational efficiency, `geospaNN` leverages the widely popular Nearest Neighbor Gaussian Process (NNGP) approximation Finley et al. (2019). It is also, to our knowledge, the first Python package to implement scalable covariance matrix computations in geostatistical models.

Statement of Need

`geospaNN` is a Python package for geospatial analysis that uses NN-GLS (Zhan and Datta 2024), a novel and scalable class of NNs explicitly designed to account for spatial correlation in the data. The package is effectively represented as a geographically-informed Graph Neural Network (GNN) by using NNGP covariance matrices. It is embedded within the PyG 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 geostatistics, but can also be applied more generally to estimation and prediction tasks involving other data with dependency structures, like time-series. `geospaNN` provides lightweight, user-friendly wrappers for data simulation, preprocessing, and model training, which significantly simplify the analytical pipeline. A 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.

The implementation of NNGP models within `geospaNN` is of independent importance. NNGP enables scalable covariance matrix inversions, which features extensively in geospatial models. There exist two widely used R-packages `spNNGP` (Finley et al. 2019) and `BRISC` (Saha and Datta 2018) 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.

State of the field

In Python, to integrate geospatial data with deep learning, several specialized tools have been developed. Notably, `TorchGeo` (Stewart et al. 2022) extends `PyTorch` (Paszke et al. 2019) for tasks such as land cover classification, object detection, and geospatial segmentation. Independently, the R package `geodl` (Maxwell et al. 2024) was recently introduced for analyzing geospatial and spatiotemporal data. However, these tools primarily supports raster and vector data, such as satellite imagery, which limits their general applications.

GNNs are the most common approaches for data with irregular geometry. It efficiently processes graph-structured data by learning graph-level representations through message passing and aggregation. For GNNs implementation, the PyTorch Geometric (PyG) library, provides a highly customizable framework for defining graph convolutional layers (Fey and Lenssen 2019). GNNs have been widely applied in geospatial analysis, including crop yield prediction (Fan et al. 2022), and traffic flow modeling (Wang et al. 2020). However, despite their growing adoption, there is still no systematic analytical software tailored for broad use within the statistical community.

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. A vignette is also available for detailed illustration of the package.

NN-GLS Overview

NNGLS considers a simple model for spatial data:

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

where $Y(s)$ and $X(s)$ are respectively the outcome and covariates observed at location s, m is a non-linear function relating $X(s)$ to $Y(s)$, to be estimated using a NN. The key distinction from the standard non-linear regression setup is that here the errors $\epsilon(s)$ is a dependent process that models spatial correlation.

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 NNs:

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

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 used in implementation of modern NNs—cannot be applied directly.

NN-GLS addresses these issues by introducing the NNGP to approximate Σ^{-1}, and it naturally equates to a specialized GNN. In brief, NNGP is a nearest-neighbor-based approximation to a full Gaussian Process with a specific covariance structure (Datta 2022). NNGP sparsifies 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:

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

where $Y^* = Q^{1/2}Y$ and $m^*(X) = Q^{1/2}m(X)$ can be obtained easily using aggregation over the nearest-neighbot directed acyclic graph specifying the NNGP approximation. The GLS loss returns to an additive form, allowing for mini-batching instead of full-batch training. In NN-GLS, the spatial parameters θ and the weights and biases parameters of the NN 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 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 spatial mixed models using NN-GLS, `geospaNN` offers a suite of additional features that support a wide range of geospatial analyses. `geospaNN` provides simulation module allowing users to customize the spatial parameters and mean functions to generate Y, X, and s. Users are allowed to customize the spatial coordinates to simulate under different context. `geospaNN` implements nearest neighbor kriging, an alternate to full kriging, which has been shown in Zhan and Datta (2024) to guarantee accurate prediction interval under various settings. For essential machine learning tasks, `geospaNN` offers modules including NN architecture design, training log report, and result visualization. `geospaNN` also implements SPLMM solution as a special case of NN-GLS. It should be an optimal choice for the Python users if efficient SPLMM solution is wanted for large geospatial datasets.

All these functions are included explicitly in the package and can be called independently (see vignette).

Discussion

The `geospaNN` package offers an efficient implementation of NN-GLS approach proposed in Zhan and Datta (2024). NN-GLS embeds NN with the spatial mixed model, accounting 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. Due to the sparsity of the NNGP approximation, `geospaNN` is seamlessly integrated into the framework of 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.

Fan, Joshua, Junwen Bai, Zhiyun Li, Ariel Ortiz-Bobea, and Carla P Gomes. 2022. "A GNN-RNN Approach for Harnessing Geospatial and Temporal Information: Application to Crop Yield Prediction." In *Proceedings of the AAAI Conference on Artificial Intelligence*, 36:11873–81. 11.

Fey, Matthias, and Jan E. Lenssen. 2019. "Fast Graph Representation Learning with PyTorch Geometric." *arXiv Preprint arXiv:1903.02428*.

Finley, Andrew O, Abhirup Datta, Bruce D Cook, Douglas C Morton, Hans E Andersen, and Sudipto Banerjee. 2019. "Efficient Algorithms for Bayesian Nearest Neighbor Gaussian Processes." *Journal of Computational and Graphical Statistics* 28 (2): 401–14.

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

Maxwell, Aaron E, Sarah Farhadpour, Srinjoy Das, and Yalin Yang. 2024. "Geodl: An r Package for Geospatial Deep Learning Semantic Segmentation Using Torch and Terra." *PloS One* 19 (12): e0315127.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. 2019. "Pytorch: An Imperative Style, High-Performance Deep Learning Library." *Advances in Neural Information Processing Systems* 32.

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

Stewart, Adam J., Caleb Robinson, Isaac A. Corley, Anthony Ortiz, Juan M. Lavista Ferres, and Arindam Banerjee. 2022. "TorchGeo: Deep Learning with Geospatial Data." In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 1–12. SIGSPATIAL '22. Seattle, Washington: Association for Computing Machinery. https://doi.org/10.1145/3557915.3560953.

Wang, Xiaoyang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. 2020. "Traffic Flow Prediction via Spatial Temporal Graph Neural Network." In *Proceedings of the Web Conference 2020*, 1082–92.

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