Spatialize v1.0: A Python/C++ Library for Ensemble Spatial Interpolation

Response to reviewers

Dear reviewer,

We sincerely appreciate the time and effort that you have dedicated to providing valuable feedback on our manuscript. We are truly grateful for your insightful comments on our paper. We find them valuable and constructive. We have provided a point-by-point response to your comments and concerns (in blue). Additionally, certain figures have been adjusted, and we have prepared a new manuscript that incorporates all the changes.

Referee #2

The manuscript "Spatialize v1.0: A Python/C++ Library for Ensemble Spatial Interpolation" introduces a python package "spatialize". The methodology is based on a previous publication (Egaña et al., 2021). The motivation of "spatialize" is to provide geostatistical tools to non-experts that lack the experience of spatial analysis, i.e., regarding spatial interpolation. The implemented ESI approach replaces the expert knowledge of a modeler with an ensemble based estimation and grid search for hyperparameters. In my view, the manuscript lacks the necessary clarity in its comparison analysis. Comparisons are only carried out through a visual assessment of results and precision maps. Typical statistics like RMSE, MAE and alike are missing. Furthermore, to underline the added value of simplifying the application for non-experts lacks a code based comparison with existing python implementations for spatial interpolation. It would also be interesting to see in this manuscript how robust the approach is, i..e how well miss-specifications can be compensated. Targeting at non-experts, a clear road map with guidance and caveats would also be beneficial.

This comment has been addressed by incorporating additional information. The revised text has incorporated tables with MAE and RMSE metrics in order to provide a more explicit numerical performance evaluation.

This *added value of simplicity* is not dependent on the code implementation; rather, it is related to the preliminary work conducted by kriging practitioners to perform spatial modelling, a requirement prior to implementing kriging. This process typically involves the use of supplementary tools and the application of expert knowledge to determine one or more variogram models, which must be manually specified when using kriging.

Libraries such as PyKrige provide support for the use of the grid search utility of scikit-learn in order to determine the optimal variogram model, although only basic variogram models are incorporated. More complex spatial structures require the use of advanced tools for variogram modelling.

When using ESI with kriging as a local interpolator, while it is true that a variogram model must be specified, the robustness of the kriging implementation becomes less important given the ensemble scheme. There are two key reasons for this. (a) In situations involving complex spatial structures that may require a complex variogram, a simple variogram will likely suffice on a local scale once the space has been partitioned; and (b) the whole point of ensemble models is that less robust individual models can be used, since we are combining the results of many "weak" models.

We understand that adding a misspecification could demonstrate the robustness of our ensemble modelling approach. This is a reasonable observation from the perspective of traditional modelling approaches. However, as mentioned previously, since we are combining the results of many 'weak' models, this misspecification will directly affect the outcome. For example, if you add an incorrect nugget or range when using the kriging interpolator, the grid search will not be able to find optimal results.

We also appreciate the suggestion to enhance accessibility for non-expert users. First, we have added "The Spatialize Library" section, which provides an introduction to the library and its main features. Users can consult the documentation for more detailed information on specific parameters and advanced functionality. Second, we have restructured the usage examples into case studies that follow a standard procedure for implementing ESI. These case studies demonstrate both the core workflow and optional features or variants, providing readers with a practical roadmap for applying the library to their own analyses.

Additionally, we will soon publish a series of complementary tutorials and examples, along with complete documentation in Read the Docs format, to further support novice users.

Specific comments:

Classical approaches are not limited to gridded data, neither kriging nor IDW (lines 328/329)

Thank you for this important clarification. You are correct that libraries like PyKrige, SciPy, and GSTools do offer interpolation for non-gridded data. We have revised the manuscript to accurately reflect this capability.

The key advantages of Spatialize lie in its flexibility and scope. Specifically, Spatialize supports interpolation up to 5 dimensions (not available in PyKrige or SciPy), provides multiple user-friendly interpolation methods with accessible interfaces (unlike GSTools, which focuses exclusively on kriging with user-specified parameters), and offers comprehensive tools, including parameter search functions for both gridded and non-gridded implementations.

We have updated Table 1 to clearly present these distinctions and provide a more accurate comparison across libraries.

Figures with grid search results could benefit from indication which parameters are currently investigated; the jig-saw pattern might e.g., be due to different variogram types

Thank you for this suggestion. We have revised all grid search implementations to clearly specify the parameter configurations being employed in each case.

How is the sill obtained? The code snippets only list range and nugget as parameters

When not explicitly specified in the grid search or ESI implementation, the sill parameter defaults to a value of 1.

Does/can the grid search also optimize the data splits, i..e tree configurations?

The partitions themselves are generated randomly and cannot be directly controlled. However, the grid search can optimize several aspects of the partitioning process, including the number of partitions and their coarseness. Additionally, users can select the partitioning algorithm (Mondrian or Voronoi) and set a seed for reproducibility.

The 3D case only shows possibilities, but lacks any explanation or discussion appropriate for a manuscript (in contrast to, e.g., a manual)

We appreciate this feedback and have removed the 3D example. The primary objective of our case studies is to demonstrate the practical usage of the library's tools with sufficient detail and analysis. Since the syntax remains consistent regardless of dimensionality, we have focused on two 2-dimensional examples that allow for clearer visualization and more thorough discussion of the results and methodology.

Comparisions lacks a number based comparisions MSE/RMSE/MAE and alike

Thank you for this suggestion. We have incorporated tables with MAE, RMSE, and MSE metrics to provide quantitative performance evaluation. For the synthetic data case study, these metrics are calculated across the entire estimation grid. For the real-world example, we use both leave-one-out cross-validation and k-fold cross-validation.

In case of the simple mean aggregation and IDW with p=1, is there an actual benefit of ESI? To my understanding, and under the assumption than on average all tree induced partitions would have the same sum of distances of its members to x, The ESI approach would just be IDW with more points.*

Even with p=1 and mean aggregation, ESI differs fundamentally from plain IDW due to its use of local interpolation, which considers only the samples within the vicinity of the target location.

In plain IDW, all sample points contribute with weights inversely proportional to distance, meaning distant points have a small but non-zero influence. In contrast, ESI creates a different local neighborhood in each partition by restricting estimation to only the samples within the same partition cell as the target location. When aggregating across partitions, ESI computes the mean of ratios rather than a single ratio of sums. Consequently, samples closer to the target appear in the same cell more frequently, gaining higher effective weight, while distant samples are excluded from most partitions entirely. This creates a probabilistic soft neighborhood that emphasizes local spatial structures, which is advantageous when studying spatially heterogeneous phenomena. In contrast, traditional IDW captures both global and local structures simultaneously.

Technical corrections:

Package name is typeset in different format, recommendation to set it always as fixed width font.

Thank you for this observation. We have revised the manuscript to consistently typeset all instances of the package name in fixed-width font using a verbatim environment.

Typo: line 179 "...ion 9rep..."

This issue has been solved (line **179**).