Journal of Geographical Systems (2025) 27:523–553
https://doi.org/10.1007/s10109-025-00485-0

ORIGINAL ARTICLE

Supervised spatial metric learning with applications to spatial clustering and spatial model prediction

Xinyue Zhang[1] · **Hong Gu**[1] · **Andrew Irwin**[1] · **Toby Kenney**[1]

Received: 6 March 2025 / Accepted: 14 November 2025 / Published online: 9 December 2025
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2025

Abstract
Spatial patterns and relationships are crucial for statistical modeling and inference across various fields. This study develops a novel approach using supervised Random Forest to compute similarity scores between locations, effectively capturing spatial dependencies of a response variable. The approach begins by enriching location coordinates, enabling Random Forest to split space into irregular shaped subspaces. The similarity score is then derived from the proportion of trees in which two locations fall in the same node for the same values of other predictors. From the resulting similarity matrix, eigen-scores and cluster labels are extracted and integrated into predictive models such as GWR, XGBoost, Random Forest, GAM, spatially and non-spatially varying coefficient (S&NVC) models and Spatial Durbin Model (SDM). Simulations and two real data analyses indicate that the similarity matrix can both capture more spatial information leading to meaningful clustering results and significantly enhances the predictive performance of various models.

Keywords Spatial correlation · Similarity matrix · Random forest · Spatial clustering

JEL Classification c31 · c14

1 Introduction

Spatial data analysis plays a crucial role in various fields such as geography, ecology, epidemiology, geostatistics, and urban planning Bailey et al. (1994). Information about spatial correlation and patterns is often represented by spatial clusters Shekhar et al. (2011), thus spatial clustering methods that can identify groups or clusters

✉ Xinyue Zhang
xn770547@dal.ca

[1] Department of Mathematics and Statistics, Dalhousie University, Halifax, Canada

which exhibit a high degree of spatial correlation within a data set have been a main technique in spatial data analysis Halkidi et al. (2001). Spatial metric or spatial clustering allows us to discover useful spatial patterns that may not be evident through conventional statistical analysis. Quantifying the spatial autocorrelation and spatial dependence can further strengthen statistical modeling and inference Overmars et al. (2003).

Traditional spatial clustering methods group spatial data points mainly based on their proximity or similarity computed according to the spatial coordinates Xu and Wunsch (2005). The commonly used proximity and similarity measures include Euclidean distance, Manhattan distance, cosine measure, and correlation-based measures Shirkhorshidi et al. (2015). In addition to the distance or similarity based clustering algorithms which rely on predefined distance measures, another commonly used clustering algorithm is the density-based spatial clustering algorithm that identifies clusters based on both spatial proximity and attribute similarity Liu et al. (2012). These existing algorithms for spatial clustering are typically unsupervised algorithms, i.e., there is not a clear interpretation or target for the spatial clusters.

Supervised and semi-supervised clustering methods incorporate external information such as labels or constraints to guide the clustering process to find clusters that are more aligned with the known class structure, for example, constraint-based clustering (Basu et al. 2008), supervised taxonomies (STAXAC Amalaman et al. (2017)) and supervised extensions of k-means and hierarchical clustering (Yadav et al. 2019; Nguyen and De Baets 2019) use labels, constraints, or representative points for clustering. Semi-supervised approaches, such as Cai et al. (2023) also use partial labels or auxiliary data to guide clustering. However, these methods are not designed for spatial clustering and do not capture spatial dependence. Existing supervised spatial clustering aims to find spatially contiguous regions that are homogeneous with respect to the target variable Y. Several algorithms (AssunÇão et al. 2006; Openshaw 1977; Guo 2008) have been developed by partitioning a spatial landscape into contiguous regions while optimizing an objective function based on the target variable (e.g., minimizing within-cluster variance of Y). Literature for supervised spatial metric learning is scarce.

In this paper, we develop a method for supervised spatial metric learning which can be in turn applied for spatial clustering and improving spatial predictive models; more specifically, we develop a new spatial similarity measure based on a Random Forest model for a specific response variable. Our similarity measure explores the spatial correlation of the response variable over the proximity in geographic coordinates conditional on other observed predictors. In other words, the similarity measure absorbs unexplained or unobserved information that is predictive of the response variable, with higher similarity values corresponding to greater homogeneity of the predictive model of Y over other observed predictors, instead of the homogeneity of the value of Y. Clustering analysis based on our similarity measure typically results in homogeneous models within the cluster.

Random Forest based similarity measures have also been studied in the unsupervised random forest framework Afanador et al. (2016), where synthetic data are generated and a random forest is trained to classify original versus synthetic samples, with the resulting proximity matrix used for clustering. Our approach differs in

that we do not generate artificial data and we incorporate the response variable and construct pseudo-observations to evaluate conditional spatial associations that are directly relevant for prediction.

We demonstrate the applicability of our algorithms in two very different fields of study: real estate prices and marine ecology. Accurately predicting house prices is crucial for business and economics Goodman and Ittner (1992). Most statistical models for house price prediction focus on the house and neighborhood attributes and some economic indicators Fan et al. (2018). However, house prices are profoundly influenced by location, which includes information related to the characteristics of the surrounding neighborhood and spatial relationships that are often difficult to measure Bourassa et al. (2007). Specifically, proximity to amenities such as schools, parks, shopping centers, and transportation hubs, as well as the quality of nearby schools and the crime rate in the neighborhood influence the house price significantly. Additionally, spatial relationships such as the access to waterfronts, or major landmarks can further impact property values. Generally, government or developers classify the neighborhoods based on the geographic proximity, amenities and infrastructure, property types and architectural styles. These data are generally useful but not sufficient to describe the importance of location in house price prediction models. There is additional information that is difficult to measure about location for house price. Our approach measures this additional information at a location conditional on the available predictor variables. This measure can naturally lead to spatial clusters such that geographically close locations are grouped together according to their latent correlations to the response variable; it can also be further incorporated into house price prediction models. We evaluate our method on real house price data, and show that the clusters based on our similarity measure are reasonable and can provide useful information for property appraisal firms. We demonstrate the utility of our method that incorporates new neighbourhood information through increased accuracy in house price prediction compared to predictions made with the model using the neighborhood information provided by the property appraisal company.

Similarly, in oceanography, spatial clustering is important because physical, chemical, and biological processes acting locally create spatial structure which then influences the temporal evolution of the biological community in the ocean. The ocean is a vast and complex system with diverse physical, chemical, and biological properties that vary across space Crowder and Norse (2008). Spatial sampling of the ocean is generally sparse, with many data collected at point locations or along transects during research cruises. Spatial clustering is used to compute area-weighted averages of these data by identifying relatively homogeneous regions of the ocean. This clustering is traditionally done using expert knowledge Longhurst (2010) or with unsupervised learning Oliver and Irwin (2008); Sonnewald et al. (2020). Delineating ecologically or biogeochemically distinct regions within the ocean using physical, chemical, and biological variables has the potential to yield insights into the spatial organization of marine ecosystems Reygondeau et al. (2013). Our spatial clustering method will generate a new meaningful spatial decomposition of the ocean using supervised learning with chlorophyll concentration as the response variable.

The rest of this paper is organized as follows. We describe our similarity measure and how to apply it for spatial clustering and improving spatial model prediction

in Section 2. We evaluate our proposed method in simulation studies in Section 3. We analyze two large real data sets in Section 4. Our conclusions are presented in Section 5.

2 Methods

The main purpose of our proposed method is to derive and investigate a measurement of the latent association for observations at different locations conditional on the observed features. The measurement will take the form of a similarity measure between locations, based on a machine-learning model for predicting the response variable. We use random forest for this model, because the decision trees that comprise a random forest are clusters based on similar values of the response variable. Once we have extracted a similarity matrix between locations, we can use it to fit improved predictive models in several different ways - we can derive a clustering of locations, or we can use multidimensional scaling methods to obtain transformed spatial co-ordinates that are better for predicting the response than the original spatial co-ordinates. Other methods, such as Geographically-Weighted Regression (GWR) or the Spatial Durbin Model (SDM) work directly with a similarity matrix to build models of the response from the predictors.

In contrast to traditional spatial clustering approaches, we aim to gain deeper insights into the underlying spatial relationships and dependencies for the response variable at different locations that may not be included among the predictors. For example, in the context of house prices, concealed spatial relationships could encompass factors such as neighborhood greenery or proximity to amenities, which may significantly influence property values but are not explicitly included as predictors in the model.

Given a set of predictors X and a set of location variables S, our model is $Y = f(S, X) + \epsilon$. The homogeneity of the model on two different locations S_1 and S_2 can be measured by the difference between $f(S_1, X)$ and $f(S_2, X)$ for all X or over the distribution of X, or equivalently $E(|f(S_1, X) - f(S_2, X)|)$. We use the random forest (RF) method as a flexible non-parametric supervised learning model that predicts the response variable Y from a set of location variables S and other observed predictor variables X. Random forest is an ideal method because tree based methods naturally form clusters, i.e., each tree in the fitted random forest model separates the training data points into different nodes with the observations falling in the same node being given the same predicted value by the tree. Random forest can be thought of as a method to split the predictor variable space into many small pieces. In order to find the association between locations, we use the fitted random forest model $\hat{f}(S, X)$ to predict for any two different locations, S_1 and S_2, given the same value for other predictors X. This can be envisaged as, for example in the real estate case, predicting prices for exactly the same house in any location. The values predicted by the random forest for any two locations will provide information due only to location, which we can use to calculate a similarity score between the two locations.

Generally, location information is provided by Cartesian coordinates or longitude and latitude values. With this simple location information, random forest can only separate the location spaces into small rectangles when the coordinate values are used in the tree splits. This will restrict the shapes of location subspaces. A more flexible approach is needed for the detection of irregular-shaped clusters. We will develop a set of enriched or redundant coordinates for locations to be included in the model input. Since the random forest method is quite resistant to overfitting, adding these redundant dimensions is not a big concern if there are a sufficient number of observations. Our method is detailed below for locations in two or three dimensional spaces and on a spatial manifold such as the surface of a sphere; an extension to higher dimensions or other types of space, e.g. spatiotemporal space, is straightforward.

2.1 Coordinate enrichment

Prior to fitting a random forest model, we generate many redundant dimensions of location coordinates for each observation. In a Cartesian system, each location is given by one set of coordinates, e.g., $(\tilde{s}_1, \tilde{s}_2)$ in two dimensional space and $(\tilde{s}_1, \tilde{s}_2, \tilde{s}_3)$ in three dimensional space. In a random forest model, when a location variable is used to split a tree node, this corresponds to a split aligned with the axes. As an example, in two-dimensional space, this will result in all horizontal or vertical cuts, restricting the shapes of location clusters. To capture more complex spatial relationships, we need to allow the shapes of location clusters to be flexible and irregular. To achieve a more flexible piece-wise linear boundary, we create variables oriented in many different directions, and allow the trees to split based on these variables.

For a two-dimensional space, instead of using only coordinates $(\tilde{s}_1, \tilde{s}_2)$, we rotate the axis to cover many directions. More specifically, suppose we allow M dimensional coordinates for a two-dimensional space, we can arrange the coordinates evenly so that the angles between the neighbouring axes are π/M. The choice of M depends on how smooth the clustering boundary needs to be and how many observations are in the training data. The enriched coordinates for location S are then $(s_1, \cdots, s_M)$, where $s_i = \tilde{s}_1 \cos\left(\frac{(i-1)\pi}{M}\right) + \tilde{s}_2 \sin\left(\frac{(i-1)\pi}{M}\right), (i = 1, 2, \cdots, M)$.

Creating an evenly spaced coordinate system for three or higher dimensional space is not trivial for an arbitrary number of enriched coordinates. Indeed in three dimensions, regularly directed axes are only possible for a few special cases with $M \in \{4, 6, 8, 12, 20\}$. Randomly generated coordinate axes are sufficient for our needs. We generate M random points uniformly distributed on a sphere and use these points to form basis directions. Then we take the dot product between the original three-dimensional coordinates $(\tilde{s}_1, \tilde{s}_2, \tilde{s}_3)$ and each of the randomly generated unit vectors to get the projection of each location on the new coordinate directions.

2.2 RF_Sim: a similarity measure between locations

Our method contains four main steps for calculating the similarity between any two locations:

1. Enrich the two-dimensional or three-dimensional coordinates into M-dimensional coordinate features $S = (s_1, \cdots, s_M)$.
2. Fit a random forest model $\hat{f}(S, X)$ using both the newly generated coordinate features S and the other observed features X.
3. Randomly select P observations from the training data identifying P sets of feature values of X (denoted by $X_1, \cdots, X_P$). For each location S_i, combine the location features S_i with each set of the selected P features X_j to form a set of P pseudo-observations: (S_i, X_j) $(j = 1, \cdots, P)$. Generate predictions $\hat{f}(S_i, X_j)$ for these pseudo-observations.
4. For any two locations k and l, count the number n_{kl}, of pairs (j, T), where T is a tree in the random forest, and $j \in \{1, \ldots, P\}$, such that the pair of pseudo-observations (S_k, X_j) and (S_l, X_j) fall in the same terminal node of the tree T. Define a similarity score as $\mathrm{sim}(S_k, S_l) = \frac{n_{kl}}{PN}$, where N is the total number of trees in the random forest model.

The similarity measure is an estimate of the expectation over the distribution of X of the conditional probability that two locations fall in the same node in the random forest model given that their X values are the same. Each tree in the random forest has the potential to separate the same training data into different nodes, corresponding to a clustering result for the locations according to the homogeneity of the response variable. The similarity measure is obtained by averaging over many trees. By conditioning on the same X value we can calculate the latent influence on the response variable of the location alone. By using the proportion of times that two locations fall in the same node instead of their model predicted values, we obtain a measure with geographically contiguous regions for the clusters.

This measure can be generalized in multiple ways. If geographically contiguous regions for the clusters are not desired or homogeneity of locations is thought to be beyond the geographical regions, we can generalize the probability that two locations fall in the same node to the random forest predicted values, and use the expected difference between the predicted values conditional on the same X value for two locations to define a dissimilarity measure. When only predicted values are used from a model, the random forest can be replaced with another flexible non-parametric model.

Once we obtain a similarity or dissimilarity measure, we can apply a number of off-the-shelf clustering methods to get the cluster analysis results or use a multidimensional scaling method to visualize the locations' influence to the response variable. There are also multiple ways that the location association derived from the similarity matrix can be used to further improve the predictive models. For example, we can further incorporate the similarity matrix in spatial or spatiotemporal models as a spatial kernel to improve the model performance. We describe several ways to utilize the similarity measure to improve a predictive model in the next section.

2.3 Predictive modelling using the similarity measure

A natural way to use the similarity matrix is as a weight matrix in the Geographically Weighted Regression (GWR) model (Fotheringham et al. 2002). GWR is a local

regression technique that allows the regression coefficients to be fitted differently for each data point by using different weights. Typically the spatial weights kernel is a Gaussian kernel based on Euclidean distance with a fixed or adaptive bandwidth chosen by leave-one-out cross-validation. By using our similarity matrix as the spatial weights kernel, it is possible to capture more nuanced relationships between data points than the mere geographical proximity provided by Euclidean distance. This allows the model to incorporate more complex spatial relations. We also include a cut-off threshold below which similarity weights are set to zero. The reason for this is that it is possible for some of the trees in the RF to not include any location variables, meaning that all locations are in the same node for these trees, thus resulting in a non-zero similarity for all pairs of locations. Setting a threshold excludes distant points from the local regression model. This threshold is chosen by cross-validation, similar to the bandwidth selection in standard GWR.

For other predictive models, there are two ways to directly incorporate the information from the similarity matrix as predictors. One is based on location clustering (discrete predictors) and the other is based on eigenvectors of the similarity matrix (continuous predictors). Since the similarity measure is built on RF models for predicting the response variable, clustering of locations based on the similarity measure means the location effects on the response are more similar within a cluster. A multidimensional scaling method can project the data on the space spanned by the eigenvectors of the similarity matrix, which can be interpreted as a nonlinear transformation of the data such that the proximity of the projected data means similar effects of locations on the response. Thus the scores of the data in the space spanned by the major eigenvectors (eigen-scores) can be used as input for a predictive model.

For prediction models incorporating location-based clustering, we first use a clustering procedure (e.g. hierarchical clustering with average linkage Murtagh and Contreras (2012)) on the training data based on the similarity matrix. This clustering step assigns each training observation to a cluster. We then build a prediction model on the training data (using e.g., XGBoost Chen and Guestrin (2016) in our examples) using both the original features X and additional clustering indicators (cluster labels).

For the eigenvector-based methods, we build a predictive model using scores of training data on the eigenvectors of the training data similarity matrix (denoted as $\mathrm{Sim}_{\mathrm{train}}$) and the other X features. In more detail, based on the eigen-decomposition, $\mathrm{Sim}_{\mathrm{train}} = E\Lambda E^T$, selecting the first K dimensions in the eigen-space, the K dimensional scores are the first K columns of $E\Lambda$.

In order to make predictions for new data points, it is necessary to calculate the similarity between new data points and the training data points. Once the RF has been fitted on the training data, the same trees can be used to determine the similarity between a new data point and each training data point using Steps 1, 3 and 4 from the RF_Sim procedure in Section 2.2. For the GWR models, these similarities can then be used as weights for the new data point we are trying to predict.

For cluster-based methods, we assign the new data point to the closest cluster from the training data to get its clustering label. For eigenvector-based methods, the K-dimensional scores on the eigenvectors E are the first K columns of $\mathrm{Sim}_{\mathrm{new}}E$, where $\mathrm{Sim}_{\mathrm{new}}$ is the similarity matrix between all new data point locations and all training data point locations.

It is noted that using a similarity measure which is derived from the training data as additional predictors to fit a predictive model on the training data again means using the training data twice, which leads to a risk of overfitting. In a data rich scenario, this can be avoided by splitting the training data to use part of the data to fit the similarity measure and use the other part to fit a predictive model. This overfitting concern is relatively low for GWR since the local regression model is fairly robust to misspecified weights.

3 Simulation

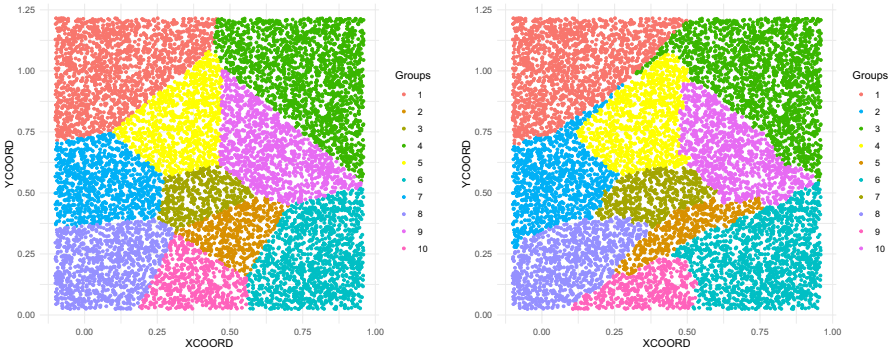
There are two purposes for the simulation: the first is to test the sensitivity of the similarity measure calculated by the algorithm RF_Sim to the hyperparameter selection. We quantify the effect according to both the clustering results and predictive model accuracy based on XGBoost model prediction on the test data. The second purpose is to evaluate the usefulness of the similarity measure in improving the model predictive accuracy. We use two simulation designs for this second purpose. The first is based on a spatially-varying coefficient linear model where we use hierarchical clustering with average linkage as clustering method and XGBoost, random forest and GWR as final predictive models to compare results; the second is based on the Spatial Durbin Model (SDM) with a global spatial dependence pattern where we aim to assess whether RF_Sim can recover the global spatial structure. We calculate the similarity measure and fit the predictive models using the same training data in all simulations.

To demonstrate the clustering effect of our method, we begin with a visualization of the hierarchical clustering results on one simulated training data where 10 true clusters are simulated using nearest neighbor rules based on 10 initial centers; then a different linear model is simulated for each cluster (details of the simulation are given in Section 3.1.1). Figure 1 shows that our method recovers cluster boundaries well.

Figure 2 shows the distribution of the similarity scores for pairs of points within and between the true clusters, using the same horizontal axis scale. It is clear that within-cluster similarity scores are significantly higher than between-cluster similarity scores and the majority of between-cluster similarity scores are very close to zero. Therefore, the similarity score matrix strongly reflects the true clustering patterns in the data.

3.1 Hyperparameter selection

We conduct a comprehensive simulation study to evaluate the effects of four key hyperparameters: P (the number of pseudo observations), N (the number of trees in the RF fitting), M (the number of enriched coordinates) and the minimum leaf size (L) in the RF model fitting. The number of pseudo observations (P) is the sample size for estimating the expectation of the similarity over the distribution of X variables, thus larger P should always give better results on average. This is similar for N: more trees in a RF, on average give better results. Even if claims from some literature that excessively large N might cause overfitting of RF models are to be believed, this is refer-



(a) 10 true clusters (b) 10 predicted clusters

Fig. 1 Example clustering of simulated data with 5000 data points: **a** true clusters **b** predicted clusters with $n = 10$

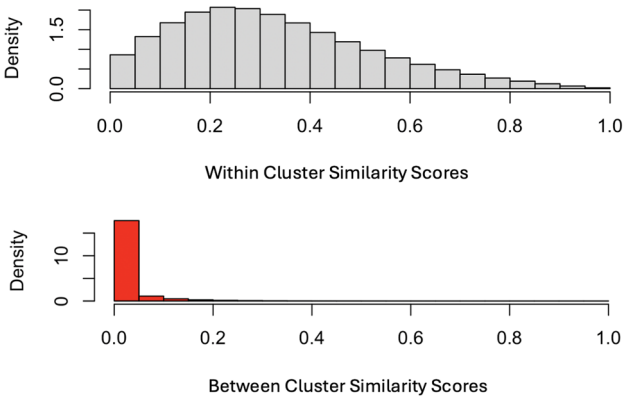


Fig. 2 Histogram of similarity scores

ring to prediction purposes, whereas in RF_Sim, we estimate the expectation of an indicator variable. However, larger values of P and N, while theoretically producing better results, mean higher computational costs. Therefore we aim to find a threshold beyond which increasing the value will not significantly improve performance. Regarding the parameter M, smaller values of M may result in a rough boundary for the spatial clusters, while larger M may introduce irrelevant features that can negatively affect the RF model performance. Thus, our goal is to find an appropriate range that balances the smoothness of the clustering boundary and RF model performance. In our implementation using the R package `randomForest`, depth is indirectly controlled by the minimum leaf size (L). Therefore, we also conduct simulation studies to select an appropriate value for L.

3.1.1 Simulation design

We simulate data with locations on a plane (2-dim) with a linear model for the response variable, and data with locations on the surface of a unit ball (3-dim) with a nonlinear model for the response variable. For 2-dim locations, data points are simulated from a uniform distribution over the unit square. We first randomly choose 10 points as 10 cluster centers and then each point is assigned to its nearest cluster center. For each data point, 8 predictor variables $X = (X_1, \cdots, X_8)^T$ are simulated from independent standard normal distributions. Then for each cluster, a set of regression coefficients $(\beta_0, \beta_1, \cdots, \beta_8)^T$ are simulated i.i.d. from standard normal distributions. A subset of one to three randomly chosen coefficients, are then replaced by zero in each set of coefficient vectors $\beta = (\beta_1, \cdots, \beta_8)^T$. Finally, a response variable is simulated using $Y = \beta_0 + X^T\beta + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$.

For 3-dim locations, we simulate uniformly generated points on a unit sphere, by sampling from a 3-dim independent normal distribution, then rescaling the vector to unit length. We first generate 10 uniformly distributed random points on the unit sphere as the cluster centers. Then, we simulate the data points uniformly over the sphere. Each point is assigned to its nearest cluster center. Similar to the 2-dim locations, eight predictor variables are simulated from $\mathcal{N}(0, I)$. Regression coefficients β are also simulated from $\mathcal{N}(0, I)$ for each cluster. To enhance the distinctiveness of each data cluster, we randomly set two elements of β to zero for each data cluster. The response variable is simulated using the nonlinear equation $Y = \beta_0 + \beta_1 X_1^2 + \beta_2\beta_3 X_2 X_3 + \beta_4\beta_5 X_4 \log(X_5^2) + \beta_6 \sin(X_6) + \beta_7\beta_8 X_7 X_8 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$.

For each experiment, we simulate 5000 points for the training data and 10000 points for test data. Results are summarized over 100 replicates for each configuration.

3.1.2 Simulation results

Comparisons are based on both test data clustering results and predictive mean squared errors on test data. For clustering results, the Adjusted Rand Index (ARI) serves as an indicator of the clustering performance. ARI is a measure of similarity between two clusterings, with values ranging from −1 to 1. A high positive ARI score indicates a strong agreement between the two clusterings; negative ARI indicates less agreement than expected by chance Rand (1971). We calculate the test ARI between the hierarchical clustering results based on RF_Sim for 10 clusters, using average linkage, and the true clusters, for each set of hyperparameter values. We use a one-sided paired t-test for the null hypothesis that ARI values are the same for a pair of hyperparameter values based on 100 replicate data sets.

For predictive models, we fit an XGBoost model on the training data with either eigen-scores or clustering labels derived from RF_Sim as additional predictors. We record the predictive mean squared error on the test data for each replicate, denoted as MSE_eig and MSE_label respectively. A one-sided paired t-test is again employed to determine whether the test errors are significantly different for a pair of hyperparameter values, based on 100 replicate data sets.

In the evaluation of the hyperparameter performance, we begin by selecting P from the options $\{10, 50, 100, 200, 400\}$, while keeping fixed values for $N = 400$ and $M = 180$ for 2-dim simulations and $M = 1000$ for the 3-dim simulations. We also fix $L = 5$ at its default value. Subsequently, we fix an optimal choice for P, and proceed to select N from the set $\{50, 100, 200, 300, 400\}$ while maintaining $L = 5$ and $M = 180$ and 1000 for the 2-dim and 3-dim simulations respectively. Then we focus on M, keeping P and N fixed at their optimal values and $L = 5$, with M chosen from $\{9, 18, 36, 90, 180\}$ for 2-dim locations and from $\{50, 100, 200, 500, 1000\}$ for 3-dim locations. Instead of conducting all possible pairwise comparisons among the evaluated hyperparameter values, we compare each candidate value with the largest value for the same hyperparameter. The null hypothesis for each test is that there is no difference in performance between two candidate hyperparameter values with the alternative hypothesis being that larger hyperparameter value results in better performance.

The p-values for 2-dim and 3-dim simulations are presented in Tables 1 and 2 respectively. Table 1 shows that even $P = 10$ and $N = 50$ are sufficient for our 2-dim simulation, and $M = 18$ is not significantly different from $M = 180$. We note that there is a p-value slightly less than 0.05 for the $M_1 = 90$, $M_2 = 180$ clustering case, but given that results for smaller M_1 are not significant, and results for $M_1 = 90$ are not significant for other measures, we ascribe this result to chance, rather than a significant difference. From Table 2, we can choose $P = 200$, $N = 100$, and $M = 100$ for 3-dim simulations. Both Tables 1 and 2 show that the results of RF_Sim are generally robust for a wide range of values of P, N and M.

Finally we fix P, N and M at their chosen values and select the minimum leaf size (L) from $L \in \{1, 3, 5, 10, 20, 30, 50, 70, 90, 110\}$. Figures 3 and 4 show the performance for 2-dim and 3-dim simulations respectively. The curves in the figures dis-

Table 1 One-sided paired t-test p-values for comparing test data ARI, MSE_eig and MSE_label under different hyperparameter values for 100 2-dim simulations

Selecting P		$N = 400, M = 180$		
P_1	P_2	ARI	MSE_eig	MSE_label
10	400	0.22521	0.46341	0.38085
50	400	0.21394	0.35751	0.41349
100	400	0.74684	0.29277	0.95732
200	400	0.33566	0.42831	0.28718
Selecting N		$P = 50, M = 180$		
N_1	N_2	ARI	MSE_eig	MSE_label
50	400	0.16129	0.11065	0.11950
100	400	0.22577	0.46720	0.11126
200	400	0.62561	0.18005	0.11154
300	400	0.89096	0.47884	0.78502
Selecting M		$P = 50, N = 200$		
M_1	M_2	ARI	MSE_eig	MSE_label
9	180	0.00071	0.86306	0.00141
18	180	0.12826	0.99996	0.35618
36	180	0.44120	0.99115	0.80315
90	180	0.19776	0.95770	0.04501

Table 2 One-sided paired t-test p-values for comparing test data ARI, MSE_eig and MSE_label under different hyperparameter values for 100 3-dim simulations

Selecting P		$N = 400, M = 1000$		
P_1	P_2	ARI	MSE_eig	MSE_label
10	400	0.02761	0.58647	0.47797
50	400	0.31986	0.85122	0.62014
100	400	0.02985	0.58597	0.70188
200	400	0.29681	0.34446	0.44946
Selecting N		$P = 200, M = 1000$		
N_1	N_2	ARI	MSE_eig	MSE_label
50	400	0.00725	0.93328	0.20146
100	400	0.2232	0.90222	0.60026
200	400	0.17428	0.85896	0.19489
300	400	0.12371	0.56574	0.67462
Selecting M		$P = 200, N = 100$		
M_1	M_2	ARI	MSE_eig	MSE_label
50	1000	0.01617	0.49095	0.01032
100	1000	0.30593	0.18936	0.62845
200	1000	0.87603	0.36757	0.62839
500	1000	0.31248	0.18328	0.45805

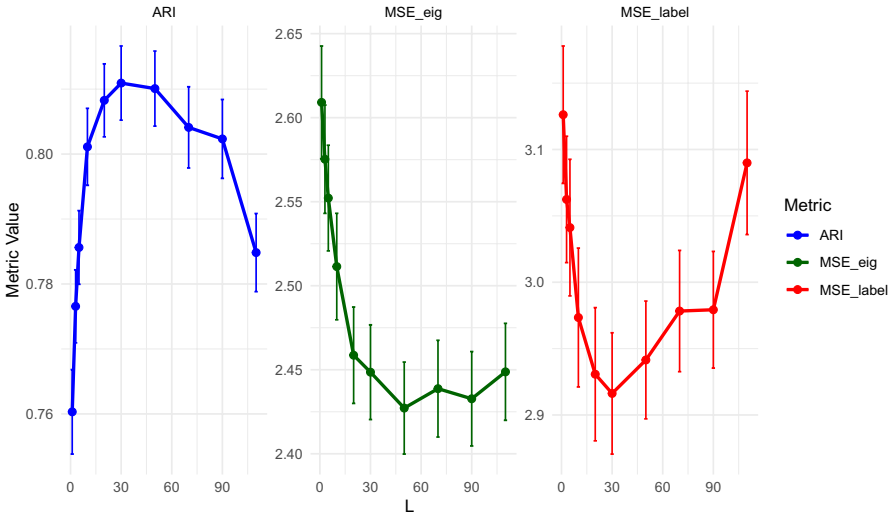


Fig. 3 Effect of RF minimum leaf size (L) on clustering accuracy and predictive performance for 2-dim simulations with $P = 10, N = 50, M = 18$

play the mean performance with error bars indicating one standard error. We see that L from 20 to 50 consistently give higher ARI and lower mean squared errors.

Further simulations are also conducted to assess the robustness of RF_Sim to the minimum leaf size (L) based on simulation design in Section 3.2.1 with (1) varied

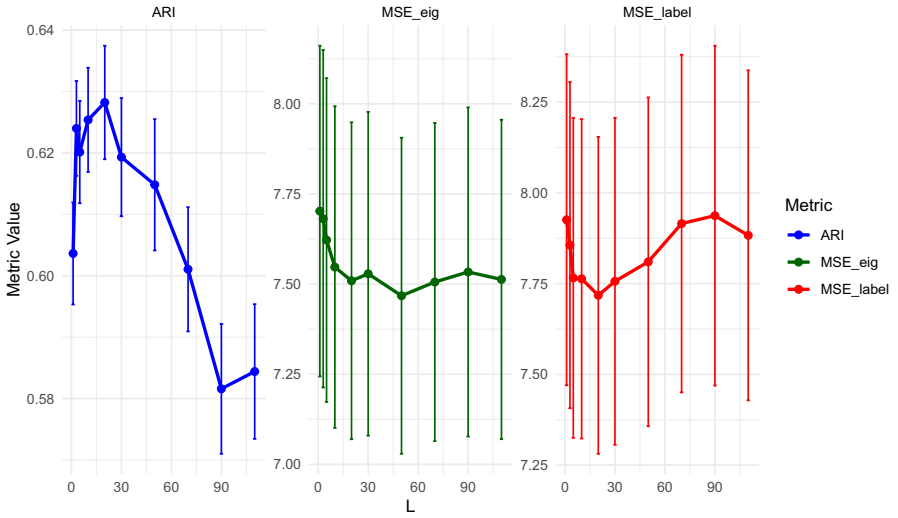


Fig. 4 Effect of RF minimum leaf size (L) on clustering accuracy and predictive performance for 3-dim simulations with $P = 100$, $N = 100$, $M = 100$

training data size $n_{\text{train}} \in \{1000, 5000, 20000\}$ and fixed signal strength $\sigma^2_{\beta_0} = 1$ and $\sigma^2_{\beta} = 1$; (2) varied intercept signal $\sigma^2_{\beta_0} \in \{0.5, 1, 2\}$ and fixed $n_{\text{train}} = 5000$ and $\sigma^2_{\beta} = 1$; (3) varied slope signal $\sigma^2_{\beta} \in \{0.5, 1, 2\}$ and fixed $n_{\text{train}} = 5000$ and $\sigma^2_{\beta_0} = 1$. For each setting, we tune L over $\{1, 3, 5, 10, 20, 30, 50, 70, 90, 110\}$ and evaluate performance using ARI, MSE_eig, and MSE_label on test data with fixed hyperparameters as $P = 100$, $N = 200$, $M = 18$.

Figures 5, 6, and 7 show the effect of L for three complementary settings over 20 replicate data sets. In all settings, $L \in \{20, 30, 50\}$ provides the best clustering accuracy and predictive performance. Therefore, we set $L = 30$ for all subsequent analyses.

3.2 Comparisons for clustering analysis and spatial model prediction over a varying coefficient linear model design

3.2.1 Simulation design

We use the 2-dim simulation design from Section 3.1.1 with 10 spatial clusters of data and a different linear regression model within each cluster to generate the response Y. Three different training data sizes (1000, 5000, and 20000) are used with test data size fixed as 10000. We vary the signal strengths by setting the variance of β_0 as 0.5, 1, or 2 and for each fixed value for variance of β_0, setting the variance of each element of β as 0.5, 1, or 2. Each of the $3 \times 3 \times 3 = 27$ scenarios is repeated 20 times (using the same set of locations and X variables for all 9 scenarios with given n_{train}). We fix the hyperparameters as $P = 100$, $N = 200$, $M = 18$ and $L = 30$ when applying the RF_Sim procedure.

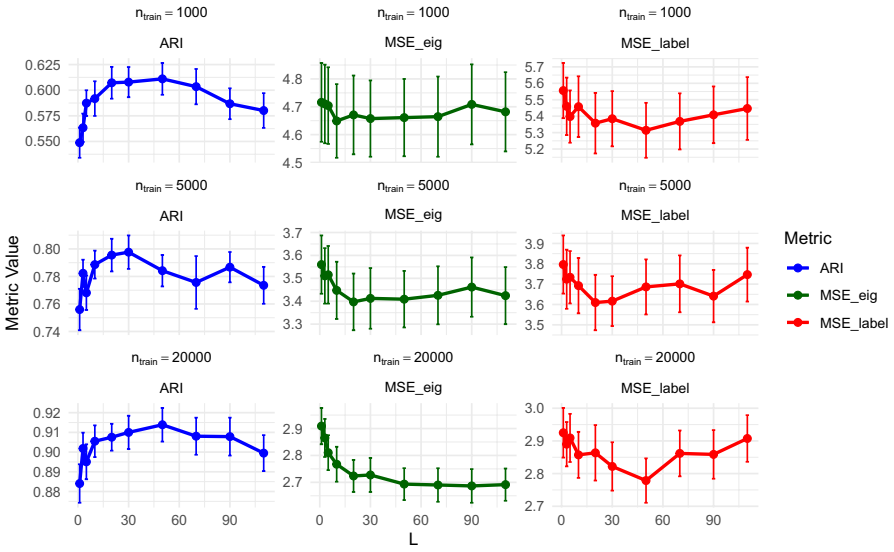


Fig. 5 Effect of minimum leaf size (L) for varied training data size (n_{train})

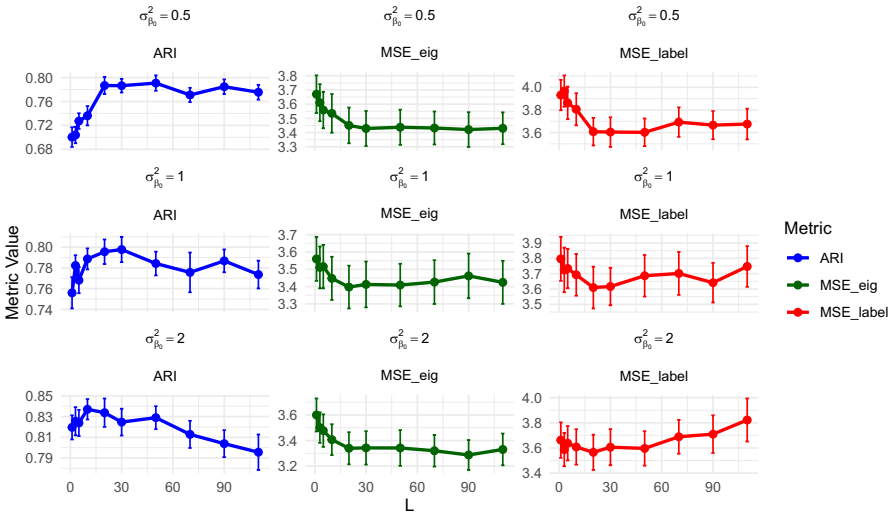


Fig. 6 Effect of minimum leaf size (L) over varied signal strength for β_0 ($\sigma^2_{\beta_0}$)

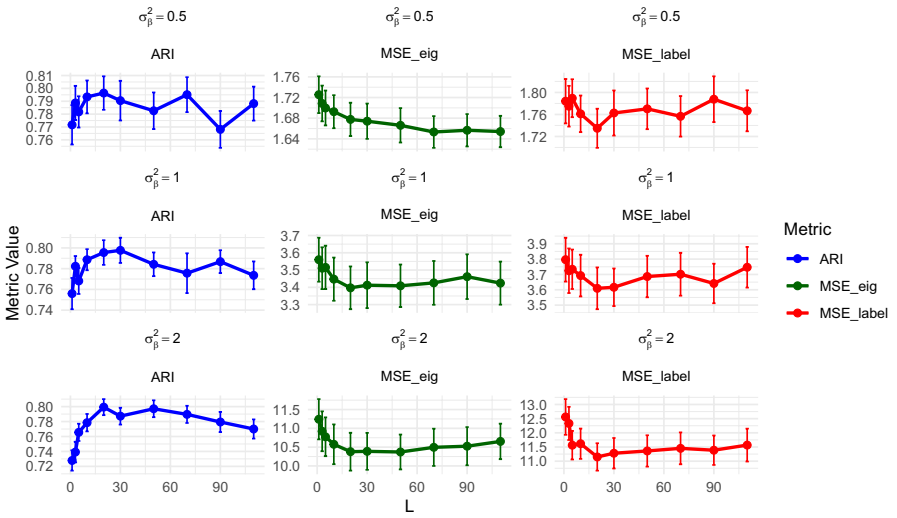


Fig. 7 Effect of minimum leaf size (L) over varied signal strength for β (σ^2_β)

3.2.2 Clustering analysis results

We compare the clustering results obtained from hierarchical clustering with average linkage based on our similarity matrix (Hclust_RFsim), hierarchical clustering using the Euclidean distance matrix (Hclust_ED), hierarchical clustering using a weighted combination of spatial coordinates, the predictors and the response variable (Hclust_SX), and K-means clustering. Both Hclust_ED and K-means clustering are only based on the location coordinates. Hclust_SX is based on the Euclidean distance calculated by the combined spatial coordinates, the predictors and the response variable, with each spatial coordinate given weight 9, so that the total weight of the spatial coordinates was twice that of the other variables. The K-means clustering method is chosen since the data are simulated under the K-means clustering design, so it is anticipated that K-means clustering should perform reasonably well for the simulated data. The comparison with Hclust_ED and Hclust_SX helps to highlight the improvement of RF_sim over the Euclidean distance. For all clustering methods, we set the number of clusters as 10 and compute the Adjusted Rand Index (ARI) of the clustering results compared to the truth on the test data.

Table 3 shows the means of the test data ARIs of the three clustering methods over 20 replicates for 27 scenarios. The test ARIs of Hclust_ED are approximately 0.45, 0.46, and 0.43 for training sizes of 1,000, 5,000, and 20,000, respectively. As we use the same 20 sets of location data for all scenarios with a given size, the results for Hclust_ED and K-means are the same for all scenarios of a given size (with results for K-means fluctuating due to randomness in the algorithm). From the table, we can see that the ARIs from Hclust_RFsim increase from about 0.6 to 0.9 as the sample size of training data increases 20-fold, while the ARIs from the K-means, Hclust_ED and Hclust_SX clustering methods change little and are notably lower. Higher signal strengths also result in more accurate estimates of the similarity matrices and bet-

Table 3 Test ARI comparisons between *K*-means, Hclust_RFsim, and Hclust_SX clustering for 2-dim simulations; "n_{train}" is the number of observations in the training data, "$\sigma^2_{\beta_0}$" is the variance of the intercept, and "σ^2_{β}" is the variance of the regression coefficients in simulated functions

n_{train}	$\sigma^2_{\beta_0}$	σ^2_{β}	Hclust_RFsim	Hclust_SX	*K*-means
1000	0.5	0.5	0.5559	0.4718	0.4564
	0.5	1.0	0.5833	0.4603	0.4574
	0.5	2.0	0.5997	0.4754	0.4476
	1.0	0.5	0.6171	0.4926	0.4498
	1.0	1.0	0.6165	0.4703	0.4503
	1.0	2.0	0.6024	0.4785	0.4514
	2.0	0.5	0.7078	0.5299	0.4514
	2.0	1.0	0.6803	0.4929	0.4557
	2.0	2.0	0.6295	0.4820	0.4477
5000	0.5	0.5	0.7665	0.4909	0.4705
	0.5	1.0	0.7994	0.5010	0.4644
	0.5	2.0	0.7961	0.4905	0.4625
	1.0	0.5	0.7811	0.5080	0.4647
	1.0	1.0	0.7935	0.4934	0.4697
	1.0	2.0	0.8075	0.4919	0.4682
	2.0	0.5	0.8333	0.5336	0.4611
	2.0	1.0	0.8193	0.5264	0.4616
	2.0	2.0	0.8160	0.5010	0.4641
20000	0.5	0.5	0.8756	0.5044	0.4580
	0.5	1.0	0.8993	0.5118	0.4621
	0.5	2.0	0.9160	0.5002	0.4568
	1.0	0.5	0.8927	0.5285	0.4655
	1.0	1.0	0.9022	0.5265	0.4630
	1.0	2.0	0.9233	0.5080	0.4562
	2.0	0.5	0.9109	0.5647	0.4635
	2.0	1.0	0.9023	0.5390	0.4527
	2.0	2.0	0.9038	0.5217	0.4701

ter results by Hclust_RFsim. Compared with distance-based clustering, our method uses the other available information in the data and it can provide better clustering performance. Moreover, Hclust_SX shows slightly higher ARIs than *K*-means and Hclust_ED. However, its improvement with increasing training size is still limited compared with Hclust_RFsim, which means our similarity matrix can effectively incorporate relationships between predictor and response variables and improve clustering performance.

3.2.3 Spatial predictive models

Our simulation is designed as a spatially-varying coefficient linear model. We compare the predictive accuracy for the following three types of models: GWR, random forests and XGBoost for different types of input variables under different signal strength and different sample sizes for the training data.

Three types of GWR model are evaluated, including a standard GWR model using a Gaussian kernel based on Euclidean distance, with bandwidth determined by cross-validation (referred to as "standard GWR" in Figure 8) and implemented by the `R`

package `spgwr` Fotheringham et al. (2009); a GWR model based on our similarity matrix without cut-off, i.e. using all training data to fit each model (termed as "GWR"); and a GWR model based on the RF_Sim derived similarity matrix with the optimal cut-off value determined by cross-validation ("GWR_CV"). For random forest, we fit three random forest models combining different location input variables with the original X variables: "RF_loc" only uses the 2-dim Cartesian coordinates as location variables. "RF_enriched" is fitted by combining the X variables with the $M = 18$ enriched location variables. "RF_eigen" is fitted by combining the X variables with the first several eigen-scores of the RF_Sim-derived similarity matrix. The number of eigen-scores included in RF_eigen is selected to retain 90% of the variance in the similarity matrix.

Similarly for XGBoost, "XGB_loc" fits XGBoost on the X variables combined with the 2-dim Cartesian coordinates location variables. "XGB_cluster" fits XGBoost on the X variables combined with cluster labels for 10 predicted clusters derived from Hclust_RFsim. "XGB_eigen" is fitted on the X variables combined with the eigen-scores of the RF_Sim derived similarity matrix (using the same eigen-scores as for the RF_eigen results). "XGB_Moran" is fitted on the X variables combined with the

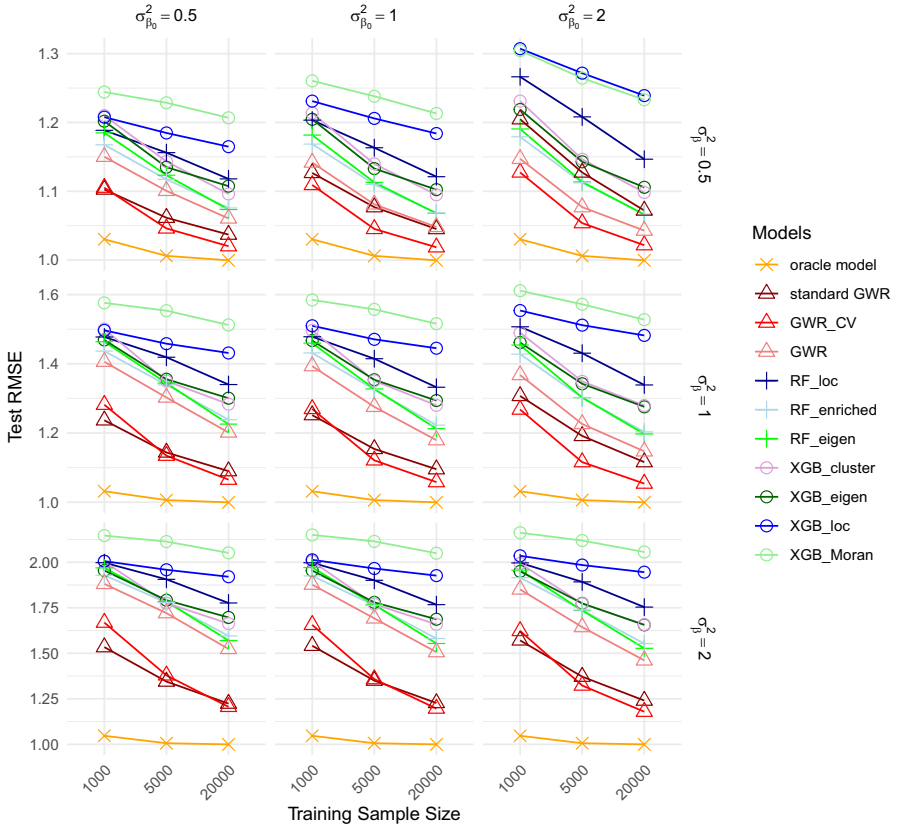


Fig. 8 Model performance comparisons on spatially-varying coefficient linear model simulation

Moran's eigenvectors Dray et al. (2006) that explain positive spatial dependence. Moran's eigenvectors are the eigenvectors associated with the spatial weight matrix in Moran's I statistic, which is a measure of spatial autocorrelation. Moran's I is a popular tool for comparison in spatial analysis, especially when dealing with spatial autocorrelation and exploring the spatial structure of data.

As a baseline comparison, we also included an oracle model which fits linear regression models for each true data cluster in the training data and predicts based on the corresponding true clusters of the test data. The average test RMSEs are summarized in Fig. 8.

From Fig. 8, we can see that when the sample size is large or signal strength is strong, the RMSE of the oracle model is almost equal to the irreducible error. All models perform better with increasing sample size in the training data.

Among the three types of models, GWR models consistently perform better than random forest and XGBoost. There may be two reasons for this result. One is that GWR fits linear regression models which are the true underlying model format. The other reason is that GWR seems less prone to overfit. Among all methods, the GWR_CV, which uses our similarity matrix as weights and a selected cut-off value determined by cross-validation, gives the best performance, except in certain settings with $n_{\text{train}} = 1000$ where standard GWR is slightly better. With limited training data, the RF-based similarity may be noisier, which increases variance. As training size increases, these similarity scores tend to stabilize and GWR_CV consistently outperforms standard GWR. The superior performance of GWR_CV compared to GWR lies in the fact that our similarity measure often gives small but non-zero values to points from different clusters, and using a hard threshold to set those to zero focuses the model on the relevant data points. Also, for GWR, downweighting a true data point to zero is often less harmful than including an incorrect data point.

Among the random forest models, the performances of RF_enriched and RF_eigen are very similar, and both are much better than RF_loc, meaning that the 2-dim Cartesian coordinates are not sufficient to capture the spatial heterogeneity. Among the XGBoost models, XGB_cluster and XGB_eigen perform similarly, with XGB_eigen performing slightly better. Both of these methods are based on our similarity matrix and both these methods largely outperform XGB_loc and XGB_Moran. Indeed the performance of XGB_Moran is even worse than RF_loc or XGB_loc. This shows that naively defined distance kernels can't capture the irregularly shaped clusters in the data and the model heterogeneity. Such kernels could possibly negatively influence the model accuracy, leading to even worse results than only providing the location information through the Cartesian coordinates to a flexible machine learning model.

3.3 Spatial durbin model simulation

The Spatial Durbin Model (SDM) Mur and Angulo (2006) extends the Spatial Autoregressive Model (SAR) by including spatial lags of the covariates and a spatially lagged response:

$$y = \rho W y + X\beta + WX\theta + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2 I).$$

where W is a row-standardized spatial weight matrix, ρ captures spatial autoregression in the response variable, and θ represents the coefficients on the spatially lagged covariates WX.

SDM is widely used in spatial econometrics because it effectively captures both direct effects and indirect spatial spillover effects. Moreover, as a nested model encompassing Spatial lag (SAR) (Kelejian and Prucha 1998), Spatial error (SEM) (Anselin 1988), and Spatially lagged X (SLX) (Angrist and Pischke 2009) models, SDM can provide unbiased coefficient estimates when estimated appropriately with a correctly specified W.

Compared with the previous local linear simulation design in Section 3.2.1, SDM displays a global spatial dependence pattern. Therefore, simulation based on SDM can be used to assess whether our RF_Sim can recover such global spatial structure.

3.3.1 Simulation design

We generate data from the SDM model with 8 predictors in the X matrix and error variance all simulated from independent standard normal distributions. The intercept is fixed at $\beta_0 = 0.5$ and $\beta = (0.5, -0.4, 0.6, 0.4, 0.3, -0.5, 0.2, 0.25)^T$. We vary the spatial autoregression parameter $\rho \in \{0.2, 0.5, 0.8\}$ and draw the lagged covariate coefficients as $\theta \sim \mathcal{N}\left(0, \delta_\theta^2 I\right)$ with $\delta_\theta^2 \in \{0.5, 1, 2\}$. Each scenario is replicated 20 times. For each replicate, we sample 15000 locations from a uniform distribution over the unit square, then generate Y from the SDM using a fixed weight matrix W_{full} which is the k-nearest-neighbors (KNN) matrix ($k = 10$) defined on the full set of 15000 locations. Each row assigns weight $1/k$ to the k closest neighbors and 0 to all others. We then randomly divide the data into 5000 training points and 10000 test points.

3.3.2 Evaluation

We fit three SDMs under alternative spatial weights W. SDM_knn uses k-nearest-neighbour matrix built from the training coordinates, where k is chosen through cross-validation among candidate values between 10 and 200. This model serves as the conventional geographic baseline model. Note that this model is misspecified, as the generating model is based on a weight matrix that includes test data values and locations that are not available in the training data. SDM_sim directly uses our similarity matrix from RF_sim as W. SDM_sim_cv applies hard thresholding to our RF_sim similarity matrix, only retaining similarity scores greater than a fixed threshold determined by cross-validation and setting all smaller values to zero.

Table 4 shows test MSE means and standard errors of MSEs for different ρ and δ_θ^2. From the table, we can see that SDM_sim_cv outperforms SDM_sim in all scenarios (though in many cases the difference is not significant). SDM_knn outperforms both SDM_sim methods when ρ is small, but performs worse than SDM_sim_cv when ρ is large. As in the GWR simulations, using a hard threshold to set small similarities to zero improves the model fitting, for the same reasons. These results show that our

Table 4 Test MSE comparisons for three SDMs over 20 replicates

		Test_MSE_mean			Test_MSE_SE		
ρ	δ_θ^2	SDM_knn	SDM_sim	SDM_sim_cv	SDM_knn	SDM_sim	SDM_sim_cv
0.2	0.5	1.08726	1.12140	1.11600	0.01263	0.01458	0.01388
0.2	1	1.28739	1.39317	1.36582	0.04267	0.0465	0.04088
0.2	2	1.88265	2.14909	1.95496	0.1593	0.11935	0.08024
0.5	0.5	1.21751	1.25794	1.23544	0.03205	0.02487	0.02131
0.5	1	1.52539	1.66136	1.55177	0.092	0.06905	0.04884
0.5	2	2.37905	2.71012	2.10644	0.30718	0.16278	0.06982
0.8	0.5	1.8118	1.84741	1.56769	0.13868	0.06273	0.02996
0.8	1	2.45775	2.62746	1.82549	0.36022	0.13674	0.04904
0.8	2	4.10717	4.39407	2.64724	1.09053	0.27970	0.14220

similarity matrix can capture a more reliable and robust representation of global spatial dependence. Especially when ρ and δ_θ^2 are larger, meaning stronger spatial structures, SDM_sim_cv achieves substantial reductions in MSE compared to SDM_knn.

4 Real data analysis

Two real data sets are used to demonstrate the effectiveness of our Hclust_RFsim in identifying location clustering patterns and improving predictive accuracy for various spatial predictive models.

4.1 House price data

Property Valuation Services Corporation (PVSC) is responsible for the yearly evaluation of all properties in Nova Scotia (NS) for the purpose of property tax. Traditionally, Nova Scotia was manually divided into homogeneous regions, and a linear regression model was fitted on each region. We apply our Hclust_RFsim method to derive clusters of house locations and compare them with PVSC's neighbourhood divisions. To demonstrate the effectiveness of our similarity matrix, we use the cluster labels and eigen-scores as additional predictors to predict house prices using various machine-learning models.

Our data include all 36,756 residential single-house sales in NS between 2013 and 2017. We use the log-transformed sales price as our response variable. Available predictors include sales history, land and building details, location and distance information, and Statistics Canada Survey data. In our study, we use 435 house features identified by PVSC as important for valuation. These features describe the specific characteristics or attributes of a house, such as living area (in square feet), effective age, and the number of bedrooms. We also have location coordinates (longitude and latitude) for each house. The original data also include several derived variables related to geographic locations, such as distances to school, hospital, highway, and major nodes. We remove all these other location-related variables to show that the spatial information derived from our method is sufficient. We convert the time variable into year and month variables for our analysis. A single random split is per-

formed here. The whole data set is randomly split into a training set (90%) and a test set (10%).

Based on our simulation results, we choose $M = 18$, $P = 100$, $N = 200$, and $L = 30$ in RF_Sim for this application. Since the sample size of this house data is much larger than our simulation experiment, we increase the number of pseudo observations (P) to make the results more reliable. Figure 9 shows the cluster results of Hclust_RFsim with the number of neighbourhoods selected as 150. Figure 10 shows zoomed-in versions of this plot around the four major cities in NS. These clusters are fairly similar to the manually selected clusters from PVSC and agree with the experience of the market. It is difficult to get these cluster patterns based on Euclidean distances between houses. We see a clear pattern to separate coastal and inland areas of the province into separate neighborhoods.

In order to further verify the effectiveness of the clustering of houses, we fit an XGBoost model to predict log-transformed house prices from house features and location variables using different numbers of neighbourhoods estimated by Hclust_

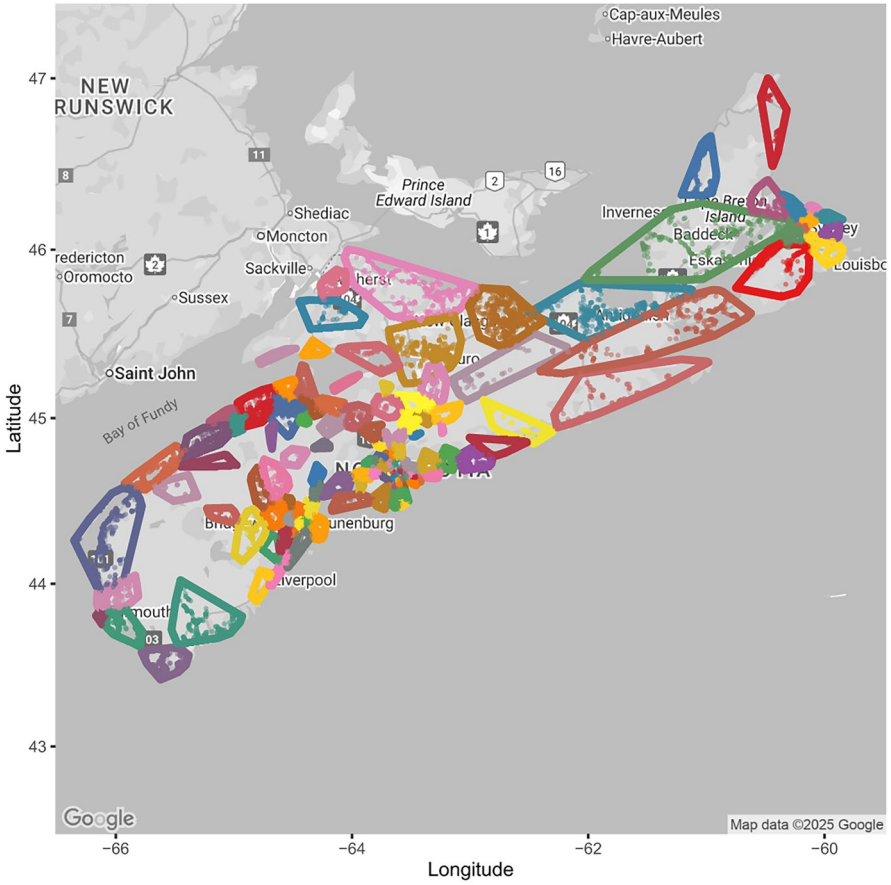


Fig. 9 Clustering results of Hclust_RFsim with 150 clusters based on house sales price between 2013 and 2017 in Nova Scotia with details for four major cities given in Fig. 10

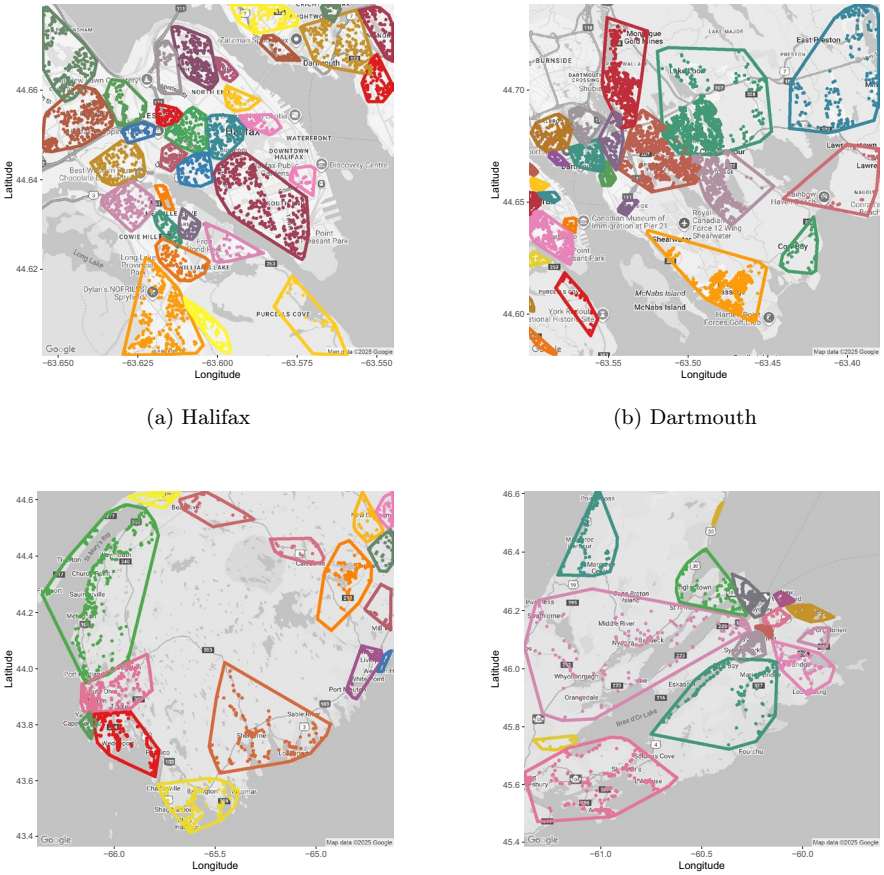


Fig. 10 Clustering results of Hclust_RFsim with 150 clusters based on house sales price between 2013 and 2017 in Nova Scotia for four major cities in NS

RFsim or using different numbers of eigen-scores derived from our similarity matrix. (In a former unpublished study with research teams from PVSC, extensive predictive model comparisons among XGBoost, GAM, neural networks and different types of local linear models with or without shrinkage on the parameters have been conducted on this data with XGBoost prediction performing best, thus we choose XGBoost over alternative methods here.) The left-hand side of Table 5 shows the test prediction RMSE using house features only, house features plus basic house locations, and house features plus various cluster labels including the manually curated 366 neighbourhoods provided by PVSC. As seen in the table, the results are fairly stable with similar RMSEs for various numbers of neighbourhoods, all around 15.9%. The best results are obtained using 150–250 neighbourhoods from Hclust_RFsim, and are comparable or slightly better than the results using 366 manually curated neighbourhoods from PVSC.

Table 5 Prediction of logarithm of House Price: accuracy comparisons by XGBoost with neighbourhood labels or eigen-scores as additional predictors on NS house price data

No. of NBHD Labels	Test RMSE	No. of Eigen-scores	Test RMSE
X only	0.23059	5	0.15119
X+location	0.16472	10	0.14807
366 PVSC	0.15971	20	0.14588
100	0.16082	30	0.14549
150	0.15792	40	0.14639
200	0.15901	50	0.14634
250	0.15873	100	0.14629
300	0.16077	200	0.14716
350	0.16131	300	0.14973
400	0.1614	400	0.15121

The right-hand side of Table 5 shows the test RMSEs of XGBoost using the house features and the eigen-scores of our similarity matrix corresponding to the largest eigenvalues. We see that the prediction RMSE is improved to 14.55% which is the best predictive accuracy for this data based on the former comparisons among different machine-learning models. This suggests that the neighbourhoods are slightly more continuous. We also see that the model is more sparse: instead of 150 neighbourhood variables we only need 30 eigen-score variables to get accurate predictions. Compared with models that exclude location information or use only latitude and longitude, models using spatial features derived from RF_sim achieve substantially better performance.

Given the performance of the eigen-scores in the predictive models, it is interesting to interpret the results by looking at the eigenvector loading matrix. Figure 11 color-codes the first six eigenvectors of the similarity matrix over the house location maps. These eigenvectors, also referred to as Empirical Orthogonal Functions (EOFs), represent the principal modes of variation in the spatial relationships between the houses. It can be clearly seen that these loadings show contrasting weights between different cities or regions such as Halifax, Cape Breton and Yarmouth etc. with some other local areas. Thus we have demonstrated that the RF_Sim-derived similarity matrix can not only be used to get better prediction results and simplify the model fitting by reducing the number of predictors. It can also provide good interpretations of the spatial patterns in the data.

4.1.1 Further comparisons with other spatial models

Unlike the local linear models we used in our simulation, the relationship between the house price and house features is highly nonlinear and heterogeneous. Although XGBoost achieves the best prediction accuracy, it is not easy to interpret how house price changes with different house features. It is natural to think that, for example, price per square foot living area changes with the location and the house features itself. In addition to the GWR models, another important class of spatial models that can fit heterogeneous nonlinear models is spatially and non-spatially varying coefficient (S&NVC) models (Murakami and Griffith 2023, 2015). The S&NVC model allows the covariate coefficients to vary depending on the spatial locations and the

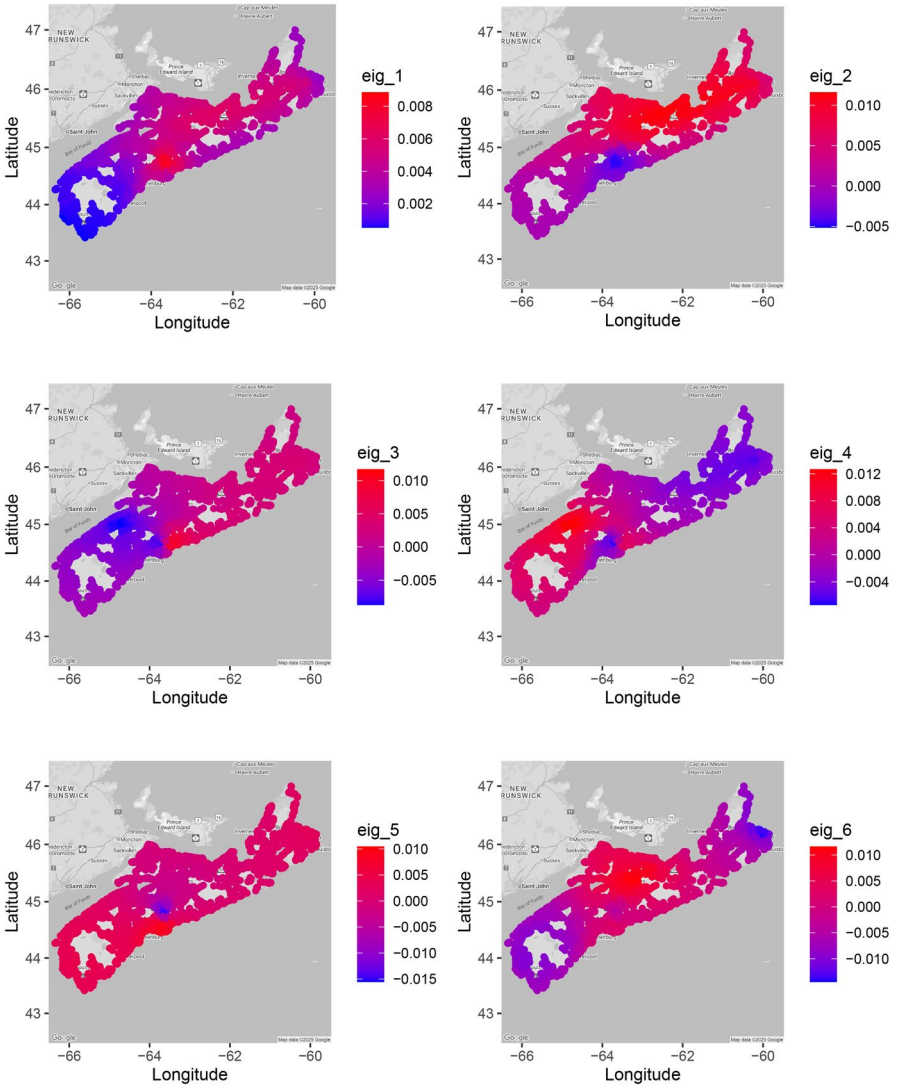


Fig. 11 Eigenvector plots of NS house sales data

covariate value itself, which can capture highly nonlinear relationships between the predictors and the outcome variable. The S&NVC model is defined as

$$y_i = \beta_0 + \sum_{k=1}^{K} x_{i,k}\beta_{i,k} + f_{MC}(s_i) + \epsilon_i, \quad \beta_{i,k} = b_k + f_{MC,k}(s_i) + f_k(x_{i,k}), \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

where $f_{MC}(s_i) = E\gamma$ and $f_{MC,k}(s_i) = E\gamma_k$, E is the matrix of Moran's eigenvectors with γ and all γ_k being random effects. Both $f_{MC}(s_i)$ and $f_{MC,k}(s_i)$ are eigenvector spatial filters derived from the Moran's Coefficient-based spatial random

process. The purpose of these is to efficiently eliminate residual spatial dependence. These processes are optimized to represent spatial dependencies based on Moran's I statistic. $f_k(x_{i,k})$ represents a smooth function of x_k, capturing the non-spatial and nonlinear influence of that particular covariate on the response variable, which is often modeled using splines (Murakami and Griffith 2023). For comparison, we fit the Moran's Coefficient-based S&NVC model (termed S&NVC_MC) and we also substitute the eigen-scores of our similarity matrix for Moran eigenvectors to re-fit the S&NVC model (termed S&NVC_ES).

Fitting the S&NVC model is very time consuming and it is not possible to include all 435 house features in the S&NVC model fitting. Since we only want to demonstrate the utility of the eigen-scores based on the RF_Sim-derived similarity matrix, we select the 10 most important house feature variables based on XGBoost. To ensure a balance between capturing fundamental spatial relationships and maintaining computational efficiency, we use the first 25 eigen-scores of the similarity matrix. The `R` package `spmoran` is used to fit S&NVC models.

For comparison, we also fit the following Generalized Additive Model (GAM) using the `mgcv` package, based on 10 house feature variables, 25 eigen-scores of our similarity matrix and their interaction terms:

$$y_i = \beta_0 + \sum_{k=1}^{K} f_k(x_{i,k}) + \sum_{l=1}^{L} g_l(E_{i,l}) + \sum_{k=1}^{K}\sum_{l=1}^{L} h_{kl}(x_{i,k}E_{i,l}) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

where $x_{i,1}, ..., x_{i,K}$ are the K house features for the ith house, and $E_{i,1}, ..., E_{i,L}$ are the first L eigen-scores from RF_Sim. f, g and h are smooth functions fitted using penalized B-splines.

We compare the standard GWR, and the GWR_CV models (as the best locally linear model from the simulation section, based on the RF_Sim derived similarity matrix), the S&NVC models using Moran's eigenvectors (S&NVC_MC) and our eigen-scores (S&NVC_ES), the GAM model as defined above, and RF and XGBoost both fitted on the 10 house features and 25 eigen-scores, in terms of the prediction error on the test data. Table 6 shows the performances of these spatial models. We see that GWR_CV outperforms the standard GWR, and S&NVC_ES achieves a lower RMSE than S&NVC_MC, indicating that the eigen-scores of our similarity matrix can capture spatial relationships more effectively. In addition, the GAM model utilizing eigen-scores as additional predictors gives good performance although slightly worse than XGBoost, indicating the highly non-

Table 6 More spatial model comparisons on house price data

	Train RMSE	Test RMSE
Standard GWR	0.21294	0.25353
GWR_CV	0.20854	0.23434
S&NVC_MC	0.22175	0.25778
S&NVC_ES	0.21019	0.21777
GAM	0.18288	0.19599
RF	0.20994	0.23601
XGBoost	0.17794	0.19366

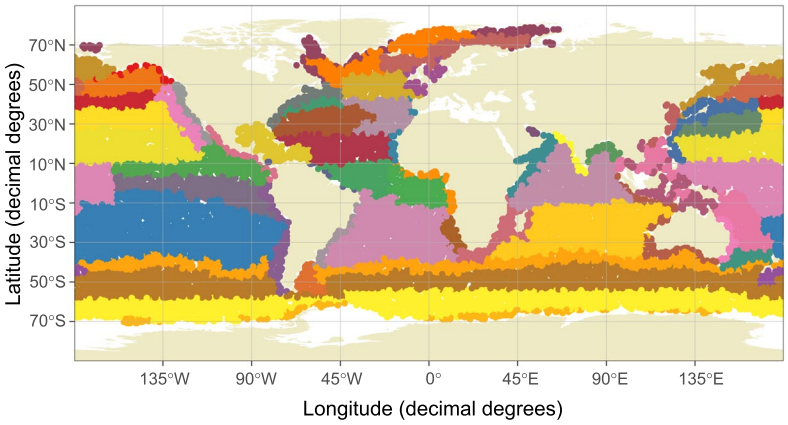
linear nature of the problem, and as in the former unpublished study, we find that XGBoost is still able to achieve best prediction for this data.

4.2 Ocean data

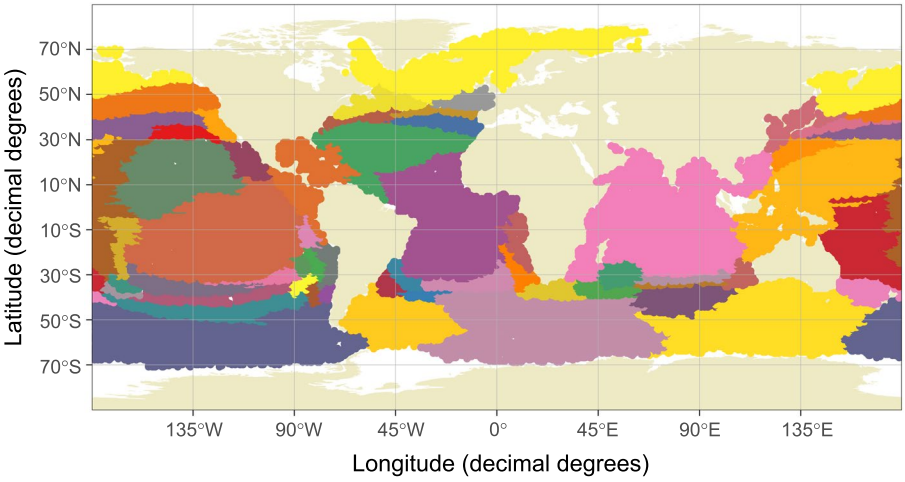
Our second data set contains data from Earth's oceans from both *in situ* and satellite remote-sensed observations (Fredriksson 2024). The measurements are recorded on a regular grid of surface ocean locations, with resolution of half a degree in both latitude and longitude. We use the chlorophyll concentration as the response variable and the other seven variables (sea surface temperature, photosynthetically available radiation, mixed layer depth, bathymetry, wind speed, eddy kinetic energy, euphotic depth) as predictors (X variables). We log-transform the chlorophyll concentration, resulting in a distribution close to normal for the response variable. Data are monthly climatological averages. This data set includes a total of 1,380,384 observations over 12 months covering 133,546 distinct locations across the globe. For our study, we randomly select 30,000 distinct locations as our training data and used the rest as our test data. For each month, we choose 100 pseudo observations. In total, $M = 1000$, $N = 200$, $P = 1200$ and $L = 30$ are selected for this application. Because of the size of the data, we did not investigate the effects of other values of M, N, P and L, with the exception of $N = 600$, which was compared in terms of predictive RMSE.

We compare our clustering results with Longhurst Province clustering, which divides the world's oceans into 57 ecologically distinct regions based on patterns of surface water properties and biological productivity (Longhurst 2010). Due to some missing data, we have a total of 55 Longhurst Province regions in our data. Figure 12 shows the clustering plots for the 55 predicted clusters based on RF_Sim and the 55 Longhurst Province regions. For the Longhurst Province clusters, there are some small regions located on the coast. In contrast, our method has fewer and less pronounced coastal clusters. Many of our predicted clusters are in the form of horizontal bands, like the Longhurst Province clusters. In particular, there was a clear pattern of hierarchical clustering at 30 to 50 degrees south latitude in our clustering, which is consistent with the Longhurst Province clustering.

Table 7 compares the accuracy of XGBoost to predict log chlorophyll concentration using month and the other seven covariates only and with different types of location variables as additional predictors: the left side is for X plus cluster labels for various numbers of clusters from Hclust_RFsim or from Longhurst Provinces; the right side is for X plus eigen-scores. X+location is the model fitted on the X variables and the latitude and longitude coordinates. From the left table, we can see that the model with 55 Hclust_RFsim clusters, which has the same number of clusters as Longhurst Provinces, performs better than the model with Longhurst Provinces. When the number of clusters is increased to 100, the prediction is even better. From the right table, we see that the latitude and longitude significantly improve the prediction of log-chlorophyll concentration both compared with a model with only the ocean measurements and with a model using cluster information without the coordinates. However, the models using our eigen-scores without the latitude and longi-



(a) Spatial clustering using Longhurst Province



(b) Spatial clustering using 55 clusters based on RF_sim

Fig. 12 Comparison of spatial clustering results on the ocean data

tude coordinates make much better predictions. We see that increasing from 25 to 50 eigen-scores improves the prediction slightly.

Furthermore, Table 7 also examines the effect of the number of trees used in RF_Sim. Recall that our similarity score is effectively an average over the trees of the random forest, so with more trees, we should obtain more stable results. We see that there is a slight improvement when we increase the number of trees to 600.

The improvement in predictive accuracy by adding either the cluster labels or eigen-scores derived from our similarity matrix demonstrates the applicability of the similarity matrix. It is useful to examine and interpret the eigenvectors of the simi-

Table 7 Prediction of Log Chlorophyll Concentration in Ocean Data: accuracy comparisons by XGBoost with predictors using X variables only or X variables with different types of location variables as additional predictors: Left side is for X plus cluster labels; right side is for X plus eigen-scores

	No.of clusters	Test RMSE	No.of eigen-scores	Test RMSE
No.of trees in RF_Sim	55 Longhurst	0.17077	X only	0.24171
			X+location	0.11140
200	55 Hclust_RFsim	0.16032	25 eigen-scores	0.11041
	100 Hclust_RFsim	0.15809	50 eigen-scores	0.10764
600	55 Hclust_RFsim	0.14837	25 eigen-scores	0.10888
	100 Hclust_RFsim	0.14816	50 eigen-scores	0.10619

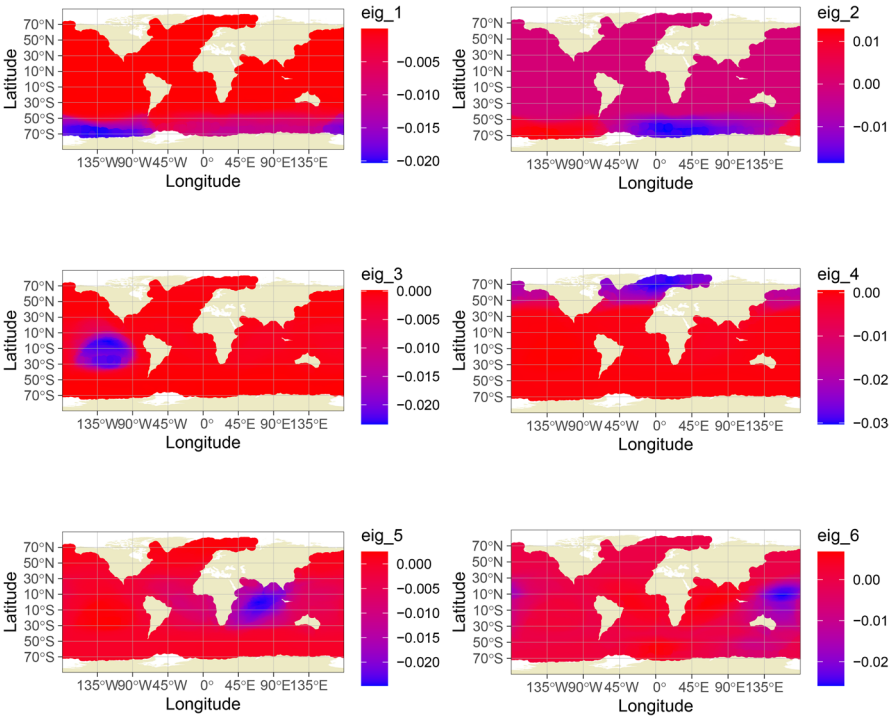


Fig. 13 Eigenvector plots for ocean data

larity matrix. Figure 13 shows the first six eigenvectors of our similarity matrix. We see that the largest eigenvectors are mostly concentrated on a single region, or else two regions with opposite signs. These regions correspond to clusters in the clustering, but instead of sharp edges, the eigenvectors vary continuously. The regions for each eigenvector make geographical sense. For example, the fourth eigenvector is concentrated in the Arctic Ocean. The fifth eigenvector is concentrated in the Indian ocean. The first, second and sixth eigenvectors are focused on various regions in the Southern Ocean. These findings are also reflected in the clustering plot, which confirms the consistency between the clusters and the spatial patterns captured by the eigenvectors.

5 Conclusions

In this paper, we proposed a new method that calculates similarity scores between different locations based on a supervised random forest model. Our similarity scores are obtained by computing the proportion of times a pair of locations are predicted to be in the same terminal nodes of the random forest trees conditional on the other predictors.

Through simulations, we demonstrated that the similarity measure obtained is robust for a large range of hyperparameter values. From the simulation results, we also observed that our similarity matrix captures more accurate information on spatial patterns and has better performance than other spatial dependence measurements such as the Euclidean distance based on latitude and longitude only. The clustering analysis results showed good agreements with the underlying true clusters. Unlike distance-based clustering methods, which fail to incorporate the relation between the spatial structure and the covariates, or model-based clustering methods where results are sensitive to the assumed model; the clustering results of our method have a clear interpretation, namely the spatial regions in which the relation between covariates and response is homogeneous. We demonstrated two real-data applications of our spatial clustering method: finding local neighbourhoods for real estate prices and finding ocean regions with similar patterns of chlorophyll concentration.

We also demonstrated three different ways that our similarity matrix can be used to improve spatial model prediction. The first is as a weighting for GWR and SDM models. The second is to apply a clustering method to our similarity score and use the cluster labels as additional predictors. The third is to include the eigen-scores from the similarity matrix as additional predictors. We showed through simulations and real data examples that all three methods improve the predictive accuracy for various spatial predictive models. In most cases, the eigen-scores yield more improvement than the clusters. We compared our clusters to previous manually annotated clustering labels, and found that models using our clusters gave better prediction accuracy.

All code used for the analyses and simulations in this paper is available at https://github.com/XinyueZ0406/Spatial-clustering-pattern-discovery-through-supervised-learning. The repository includes implementations of the RF_sim construction procedure, scripts for simulation studies, and reproducible workflows for the ocean chlorophyll dataset.

Acknowledgements The authors gratefully acknowledge the support of Mitacs. The computation support was provided by Compute Canada www.computecanada.ca. The source code for the models presented in this paper is publicly available at https://github.com/XinyueZ0406/Spatial-clustering-pattern-discovery-through-supervised-learning.

Funding Mitacs: https://doi.org/10.13039/501100004489, Compute Canada: https://doi.org/10.13039/100013020.

References

Afanador NL, Smolinska A, Tran TN, Blanchet L (2016) Unsupervised random forest: a tutorial with case studies. J Chemo 30(5):232–241

Amalaman PK, Eick CF, Wang C (2017) Supervised taxonomies—algorithms and applications. IEEE Trans Knowl Data Eng 29(9):2040–2052

Angrist JD, Pischke J-S (2009) Mostly harmless econometrics: an empiricist's companion. Princeton University Press

Anselin L (1988) Spatial econometrics: methods and models (vol. 4). Springer Science & Business Media

AssunÇão RM, Neves MC, Câmara G, Freitas CDC (2006) Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. Int J Geogr Inf Sci 20(7):797–811

Bailey TC, Fotheringham S, Rogerson P (1994). A review of statistical spatial analysis in geographical information systems. Spatial Analysis and GIS, 13–44,

Basu S, Davidson I, Wagstaff K (2008) Constrained clustering: Advances in algorithms, theory, and applications. Chapman and Hall/CRC

Bourassa SC, Cantoni E, Hoesli M (2007) Spatial dependence, housing submarkets, and house price prediction. J Real Estate Financ Econ 35:143–160

Cai J, Hao J, Yang H, Zhao X, Yang Y (2023) A review on semi-supervised clustering. Inf Sci 632:164–200

Chen T, Guestrin C (2016). Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining pp. 785–794

Crowder L, Norse E (2008) Essential ecological insights for marine ecosystembased management and marine spatial planning. Mar Policy 32(5):772–778

Dray S, Legendre P, Peres-Neto PR (2006) Spatial modelling: a comprehensive framework for principal coordinate analysis of neighbour matrices (pcnm). Ecol Model 196(3–4):483–493

Fan C, Cui Z, Zhong X. (2018). House prices prediction with machine learning algorithms. In: Proceedings of the 2018 10th international conference on machine learning and computing pp. 6–10

Fotheringham AS, Brunsdon C, Charlton M (2002) Geographically weighted regression: The analysis of spatially varying relationships. John Wiley & Sons

Fotheringham AS, Brunsdon C, Charlton ME (2009) Geographically weighted regression. Sage Handbook Spat Anal 1:243–254

Fredriksson, B. (2024). Ocean clustering repository. https://github.com/brorfred/ocean clustering. (Accessed: 19 Nov 2024)

Goodman JL Jr, Ittner JB (1992) The accuracy of home owners' estimates of house value. J Hous Econ 2(4):339–357

Guo D (2008) Regionalization with dynamically constrained agglomerative clustering and partitioning (redcap). Int J Geogr Inf Sci 22(7):801–823

Halkidi M, Batistakis Y, Vazirgiannis M (2001) On clustering validation techniques. J Int Inform Syst 17:107–145

Kelejian HH, Prucha IR (1998) A generalized spatial two-stage least squares procedure for estimating a spatial autoregressive model with autoregressive disturbances. J Real Estate Financ Econom 17(1):99–121

Liu Q, Deng M, Shi Y, Wang J (2012) A density-based spatial clustering algorithm considering both spatial proximity and attribute similarity. Comput Geosci 46:296–309

Longhurst AR (2010) Ecological geography of the sea. Elsevier

Mur J, Angulo A (2006) The spatial durbin model and the common factor tests. Spat Econ Anal 1(2):207–226

Murakami D, Griffith DA (2015) Random effects specifications in eigenvector spatial filtering: a simulation study. J Geogr Syst 17:311–331

Murakami D, Griffith DA (2023) Balancing spatial and non-spatial variation in varying coefficient modeling: a remedy for spurious correlation. Geogr Anal 55(1):31–55

Murtagh F, Contreras P (2012) Algorithms for hierarchical clustering: an overview. Wiley Interdiscip Rev Data Min Knowl Discov 2(1):86–97

Nguyen B, De Baets B (2019) Kernel-based distance metric learning for supervised k-means clustering. IEEE Trans Neural Netw Learn Syst 30(10):3084–3095

Oliver MJ, Irwin AJ (2008) Objective global ocean biogeographic provinces. Geophys Res Lett. https://doi.org/10.1029/2008GL034238

Openshaw S (1977). A geographical solution to scale and aggregation problems in region-building, partitioning and spatial modelling. Transactions of the Institute of British Geographers, pp. 459–472,
Overmars KP, De Koning GHJ, Veldkamp A (2003) Spatial autocorrelation in multi-scale land use models. Ecol Model 164(2–3):257–270
Rand WM (1971) Objective criteria for the evaluation of clustering methods. J Am Stat Assoc 66(336):846–850
Reygondeau G, Longhurst A, Martinez E, Beaugrand G, Antoine D, Maury O (2013) Dynamic biogeochemical provinces in the global ocean. Glob Biogeochem Cycles 27(4):1046–1058
Shekhar S, Evans MR, Kang JM, Mohan P (2011) Identifying patterns in spatial information: a survey of methods. Wiley Interdiscip Rev Data Min Knowl Discov 1(3):193–214
Shirkhorshidi AS, Aghabozorgi S, Wah TY (2015) A comparison study on similarity and dissimilarity measures in clustering continuous data. PLoS One 10(12):e0144059
Sonnewald M, Dutkiewicz S, Hill C, Forget G (2020) Elucidating ecological complexity: unsupervised learning determines global marine eco-provinces. Sci Adv 6(22):eaay4740
Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678
Yadav N, Kobren A, Monath N, Mccallum A (2019). Supervised hierarchical clustering with exponential linkage. International Conference on Machine Learning (pp. 6973–6983)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.