



Spatial machine learning: new opportunities for regional science

Katarzyna Kopczewska¹

Received: 30 May 2021 / Accepted: 8 December 2021 / Published online: 24 December 2021
© The Author(s) 2021

Abstract

This paper is a methodological guide to using machine learning in the spatial context. It provides an overview of the existing spatial toolbox proposed in the literature: **unsupervised learning, which deals with clustering of spatial data, and supervised learning, which displaces classical spatial econometrics**. It shows the potential of using this developing methodology, as well as its pitfalls. It catalogues and comments on the usage of spatial clustering methods (for locations and values, both separately and jointly) for **mapping, bootstrapping, cross-validation, GWR modelling and density indicators**. It provides details of spatial machine learning models, which are combined with spatial data integration, modelling, model fine-tuning and predictions to deal with **spatial autocorrelation and big data**. The paper delineates “already available” and “forthcoming” methods and gives inspiration for transplanting modern quantitative methods from other thematic areas to research in regional science.

JEL Classification C31 · R10 · C49

1 Introduction

Since its growth on the 1980s, machine learning (ML) has attracted the attention of many disciplines which are based on quantitative methods. Machine learning uses automated algorithms to discover patterns from data and enable high-quality forecasts, although the relationships between input data have not been widely studied. This is contrary to classic statistics and econometrics, which are designed to use formal equations, make inferences and test hypotheses to conclude on population having a sample, while forecasts are of secondary importance. ML often works similar to a black-box testing approach, not as an explicitly defined, commonly used statistical and econometric model. **ML has three primary purposes: clustering of data into unknown a priori groups, classification of data into known groups based**

✉ Katarzyna Kopczewska
kkopczewska@wne.uw.edu.pl

¹ Faculty of Economic Sciences, University of Warsaw, Warsaw, Poland

on a trained model and prediction. According to Google Ngrams, its current applications are found approximately ten times more frequently in text than those of econometrics, but still around seven times less frequently than those of statistics. In many research areas (such as epidemiology, geology, ecology and climate), it has become a standard, but this is yet to occur in the field of regional science. Spatial methods need spatial data. Recent assessment has found that around 80% of all data can have a geographic dimension, and much of these data can be geo-referenced (VoPham et al. 2018). Spatial information can stem from conventional sources such as regional databases in statistical offices, grid datasets and geo-located points. One can also easily gather data from Open Street Map and Google Maps (in the form of background maps, points of interest (POI), roads and traffic, for example), as well as from geo-referenced images (such as satellite photographs (Rolf et al. 2021), night light photographs and drone photographs), geo-tagged social media posts on Twitter and climate sensors. This type of data requires powerful computational methods due to its complexity, diversity and volume.

Machine learning (ML) is the ability of a machine to improve its performance based on previous results. It is a part of artificial intelligence and can be divided into unsupervised learning, supervised learning and semi-supervised learning, depending on the algorithms used. Deep learning, being a subfield of machine learning, uses neural networks for training the models. Big data is linked to data mining and knowledge discovery on large datasets by using machine learning techniques.¹ Machine learning algorithms can be implemented on ready-to-use data to obtain simple, self-standing machine learning forecast or used with workflow and data processing or even apply it to artificial intelligence, where decisions are made by algorithms.² The hermetic nature of the scientific communities may give the impression that ML methods are largely inaccessible to a wider audience. However, ML has a great potential in non-big data analysis in terms of its methods being used as supplements to spatial statistics and econometrics. The goal of this paper is to present a methodological overview of machine learning in the spatial context. Firstly, it outlines the nature of the information ML gives us, and concludes if ML is substitutive or complementary to the traditional methods. Secondly, it presents two ways in which ML has been incorporated into spatial studies—by using typical ML on spatial data and by developing new ML methods dedicated to spatial data only. Thirdly, it aims to promote the application of ML to regional science. The paper concentrates solely on the following selected ML methods: unsupervised learning, which is closer to traditional statistics and encompasses clustering, and supervised learning, which is closer to econometrics and encompasses classification and regression.³ A

¹ Artificial intelligence (AI) is often defined as a “moving target” with regards to technological challenges; its main feature is to make decisions. AI wider definition is about performing tasks commonly associated with intelligent beings, such as reasoning, discovering meaning, generalizing, or learning from past experience. Early examples of AI include computers playing chess; current example would be an autonomous car.

² The popularity of Artificial Intelligence (AI) results in its overuse; e.g. VoPham et al. (2018) named a standard predictive model of environmental exposure (for PM_{2.5} air pollution) geospatial AI (geoAI).

³ In review of ML in the spatial context, Du et al. (2020) limit machine learning to regression models only, which is not true, and they fail to mention clustering tasks.

general overview of these methods is presented in “Appendix 1” and their R implementation in “Appendix 3”.

2 Statistical applications of machine learning in regional science

Unsupervised learning is the collection of machine learning methods that are equivalent to statistics. Like data mining, it does not study relationships or causality, but instead looks for unknown but meaningful data patterns. Unsupervised learning covers mainly clustering, dimension reduction and association rules. In spatial data analysis, of course, the core area of interest is geographical location. The methodological question is how to address this unique attribute of spatial data. The separation between observations is measured with distance. It can be an intuitive, shortest (Euclidean) distance from one point to another point on the plane but can also be a multi-dimensional distance between quantitative and qualitative variables. This is why machine learning, in addition to Euclidean distance, also uses Manhattan, Minkowski, Gower, Mahalanobis, Hamming, cophenetic and cosine distances (see “Appendix 1”).

One should remember that the remarkable progress observed in recent years related to ML has caused the methodological standards to change—new developments have replaced previous innovations, and some solutions have transpired to be a dead end. The discussion below presents an overview of these diverse methods, including their development trajectories and their usefulness in spatial analysis.⁴

2.1 Clustering of points in space

Geo-located points, independently of having features assigned, are characterised by the longitude and latitude (x, y) projected coordinates. Based on this information, one can group observations into spatial clusters, which will be spatially continuous and covering all analysed points. In the case of a small- or medium-sized sample n , one can use the *k-means* algorithm, mostly with Euclidean distance metrics. It works well for limited values of n , as it requires the computation of resource-consuming $n \times n$ mutual distance matrix and solves the problem as an optimisation model.⁵ Centroids of *k-means* clusters are artificial points (potentially not existing in a sample), located in order to minimise distances between points within a cluster. With larger datasets, one applies the CLARA (*clustering large applications*) algorithm, which is the big data equivalent of PAM (*partitioning around medoids*). Both methods also apply distance metrics (such as Euclidean) but work iteratively in search of the best real representative point (*medoid*) for each cluster. In CLARA, the restrictive issue of the $n \times n$ distance matrix is solved by sample shrinking when sampling; PAM suffers from the same limitations as

⁴ Increasingly one can find in the literature comparisons of different spatial clustering methods, e.g. Jégou et al. (2019) in an empirical example, and Yuan et al. (2020) in looking for outliers.

⁵ The $n \times n$ distance matrix can be simplified using the Fastmap and modified Fastmap algorithm.

the *k-means* algorithm in this regard. Quality of clustering is typically tested with silhouette or gap statistics (see “Appendix 1”). This mechanism can be applied to delineating catchment areas (e.g. for schools, post offices and supermarkets) or to divide the market for sales representatives—in both instances, the challenge is to organise individual points around centres, with possible consideration of capacity and/or fixed location of the centre. Aside from statistical grouping, clustering has huge potential for forecasting. A calibrated clustering model enables the automatic assignment of new points to established clusters. The prediction mechanism works on the basis of the *k-nearest neighbours* algorithm.

In a portfolio of clustering methods based on a dissimilarity matrix (being equivalent to a matrix of distances between points), one can assign hierarchical grouping. For n observations, the results are presented in a dendrogram, showing continuous division from 1 to n clusters. It is based on the *k-nearest neighbours* (*knn*) concept and can be applied to clustering points or values. The hierarchical clustering algorithm works iteratively, starting from the state in which each observation is its own cluster. In the next steps, the two most similar clusters are combined into one until point is reached when a single cluster is created. The final result is the assignment of points to clusters, as is also the case with *k-means*, PAM and CLARA.

Clustering with the *k-means* algorithm has the significant advantages of ease of interpretation, a high degree of flexibility and computational efficiency; however, its main disadvantage lies in the need to specify a priori the number of k clusters. If it does not result from analytical assumptions (e.g. known number of schools to define catchment areas), it can be optimised by checking partitioning quality measures for different k values, or it can follow density. Brimicombe (2007) proposed a dual approach to cluster discovery, which is to find density clusters (“hot spots”) using, for example, GAM or kernel density and use these as initial points in *k-means* clustering. This automates the selection of k and speeds up the computations by setting starting centroids.

In other applications, *k-means* helps to build irregular, non-overlapping spatial clusters so that spatially stratified sampling can be run from those clusters (e.g. Russ and Brenning 2010; Schratz et al. 2019). This solves the problem of inconsistency in bootstrapping (Chernick and LaBudde 2014; Kraamwinkel et al. 2018) and addresses the issue of autocorrelation in cross-validation (as discussed later in the text). *K-means* irregular partitioning can also be applied to the block bootstrap (Hall et al. 1995; Liu and Singh 1992). Sampling blocks of data from spatially pre-defined subsamples allows for drawing independent blocks of data but lowers the computational efficiency.

2.2 Clustering of features regardless of location

Features measured in regions (or territorial units) can also be clustered to form possible homogenous clusters, which are later mapped. A very interesting example of a spatial study with hierarchical clustering presented in a dendrogram analyses fire distribution in Sardinia. It evidences phenological metrics as well as spatio-temporal dynamics of the vegetated land surface (normalised difference vegetation index

[NVDI] from satellite photographs) (Bajocco et al. 2015) of each territorial unit. Hierarchical clustering groups the territorial units into similarly covered areas. For each cluster group, the fire frequency is determined in order to assess the natural conditions that increase and decrease fire-proneness.⁶

Non-spatial k -means clustering may also help in the detection of urban sprawl. Liu et al. (2008) proposed a-spatial partitioning of local spatial entropy H calculated for a gridded population. Local spatial entropy is expressed as $H = \sum_i p_i \ln(p_i)$, where p_i is the relative population in the analysed cell and eight neighbouring grid cells and $\sum_{i=1}^9 p_i = 1$. Clustering of entropy, when mapped, may delineate areas with high and low local density.

Clustering assignments may reveal uncertainty, which can be addressed. Hengl et al. (2017) mapped soil nutrients in Africa, by selecting a number of clusters through running hierarchical clustering for parameterised Gaussian mixture models and optimising the Bayesian information criterion. Clustering itself is run on Aitchison compositions of data, which helps to avoid highly skewed variable space. They use fuzzy k -means, which may classify observations into a few clusters with some probabilities. This uncertainty of multi-cluster assignment can be mapped using the scaled Shannon entropy index (SSEI). In the Hengl et al. (2017) study, the SSEI reflected the density of sample points and extrapolation effects.

2.3 Clustering of locations and values simultaneously

The clustering of locations and values in the individual procedures presented above can be linked. In the literature, some examples of spatially restricted clustering can be found. All of them deal with the issue of integrating spatial and non-spatial aspects. In general, they take two approaches: order of clustering—spatial issues first and then data (spatial-data-dominated generalisation) or the opposite (non-spatial-data-dominated generalisation) or evaluating a trade-off by mixing or weighting dissimilarity matrices of data and space. As Lu et al. (1993) show, the order of spatial and non-spatial clustering matters for the result.

Historically, the oldest application is **SKATER** (*Spatial “K”luster Analysis by Tree Edge Removal*) introduced by Assunção et al. (2006), extended as **REDCAP** (*Regionalisation with dynamically constrained agglomerative clustering and partitioning*) by Guo (2008) and recently improved as **SKATER-CON** (Aydin et al. 2018). It is based on pruning the trees. For each region, it formulates a list of contiguous neighbours, and for each neighbour, it calculates the cost, that is, the total distance between all variables attached to areas. For each region, an algorithm chooses the two closest neighbours (in terms of data) and finally groups areas into the most coherent spatially continuous clusters. SKATER can be used in dynamic data analysis for robust regionalisation—as in drought analysis in Pakistan (Jamro et al. 2019). It is also used to group GWR coefficients (see below).

⁶ Clusters are not always derived using a partitioning procedure. An example of detecting spatial clusters is a study on local obesity in Switzerland. Joost et al. (2019) mapped the local Getis-Ord G_i^* statistics for body mass index (BMI) and sugar-sweetened beverages intake frequency (SSB-IF), drawing conclusions “optically” from visualisation about spatial agglomeration of high and low values of G_i .

Among the latest solutions is **ClustGeo (Chavent et al. 2018)** which examines the potential clustering of data and locations by studying the inertia of parallel hierarchical grouping of space and values. It derives two inertia functions (for space and values) depending on division. A compromise, when both inertia functions cross, sets the proportion of both groupings expressed by mixing parameter α . It weights both dissimilarity matrices,⁷ D_0 for values and D_1 for locations, in order to increase the clusters' spatial coherence. ClustGeo (CG) developed by Chavent et al. (2018) was extended as **Bootstrap ClustGeo (BCG)** by Distefano et al. (2020). The bootstrapping procedure generates many CG partitions. Spatial and non-spatial attributes are combined with the Hamming distance based on dissimilarity measures (silhouette, Dunn, etc.) and are used in CG to obtain final partitioning, which minimises the inertia within clusters. **The BCG approach outperforms CG**, as proved by dissimilarity measures. However, the algorithms are very demanding due to the dissimilarity matrix, which limits their application in the case of big data.

Clustering of locations and values jointly is also possible with k -means. It was applied to seismic analysis of the Aegean region (Weatherill and Burton 2009), for which not only the location of earthquakes but also their magnitude is essential. Proposed k -means clustering of locations refers to the magnitude in a quality criterion—the k -means optimisation requires minimising the total sum of squares within clusters, which means subtracting the individual values from the cluster average within each cluster. This cluster average was replaced by a magnitude-weighted average, which shifts the centroid of a cluster towards the stronger earthquakes.

Spatially oriented k -means clustering appears not only in regional science but also in biostatistics. In mass spectrometry brain analysis, the imaging is based on pixels, in which one observes spectra—being technically equivalent to time series. Alexandrov and Kobarg (2011) proposed the idea of spatially-aware k -means clustering. As with every k -means approach, it is based on a dissimilarity (distance) matrix between pixels. To compare the distance between pixels, a composite distance between their spectra is determined. Instead of directly comparing two spectra (one from each pixel), the method compares two weighted spectra, each averaging the neighbouring spectra in radius r , similar to the spatial lag concept. Even if k -means clustering itself has no spatial component, the distances used in clustering include neighbourhood structure.

2.4 Clustering of regression coefficients

Clustering procedures are more frequently applied to values than to geo-located points. In regional science, a popular approach is to cluster *beta* coefficients using geographically weighted regression (GWR). GWR operates as multiple local regressions on point data, which estimate small models on neighbouring observations. This generates individual coefficients for each observation and variable, making those values challenging to summarise in a traditional manner. Mapping of the clustered regression coefficients enables an efficient overview to be had. As many studies

⁷ In the traditional a-spatial approach, clusters for observations are created based on a set of attributes assigned to these observations, while their diversity is reflected in the dissimilarity matrix D_0 .

show (e.g. Lee et al. 2017), clusters are predominantly continuous over space, even if computations do not include explicitly locational information.

These output data—clustered GWR coefficients—can be used in a few ways in further analysis. Firstly, they can be used in profiling the locations assigned to different clusters—a study by Chi et al. (2013) uses *k*-means clusters to present an obesity map. Secondly, one can model spatial drift (Müller et al. 2013), which addresses heterogeneity and autocorrelation. In the global spatial econometric model, which typically controls autocorrelation, one includes dummies for each cluster assignment, reflecting spatial heterogeneity. Müller et al. (2013) applied this approach to modelling public transportation services. Thirdly, one can model spatio-temporal stability (Kopczewska and Cwiakowski 2021). For each period, GWR coefficients are estimated and clustered separately. Next, they are rasterised, and for each raster cell, the median or mode values of the cluster ID are calculated. Finally, the Rand index and/or Jaccard similarity index is applied to test the temporal similarity of the median/mode cluster ID in each cell. This approach, which originally has been applied to housing valuation, can test spatio-temporal stability of clusters in any context. Fourthly, one can try to generalise clusters based on inter-temporal data. Soltani et al. (2021) applied GTWR (geographically and temporally weighted regression) to obtain single-period local coefficients and used the SKATER algorithm, which clusters both locations and values, to delineate submarkets. Helbich et al. (2013) derived MGWR (mixed GWR), which keeps coefficients with non-significant variation constant for inter-temporal housing data. For fully spatial coverage, they kriged coefficients, reduced dimensions with PCA and clustered with SKATER, which allowed for the derivation of robust submarket division.

It is not only GWR coefficients that can be clustered. In general, clustering requires multiple values to be grouped. This occurs in bootstrapped regression. The majority of the literature runs bootstrapped OLS (ordinary least squares) models with a single explanatory variable only, enabling a simple summary of beta in one-dimensional distribution. However, for more than one explanatory variable, derivation of “central” coefficient values requires multi-dimensional analysis, which has not been presented in the literature until now. A solution to this problem is a PAM algorithm in the one-cluster study. As it searches for the in-sample “best representative”, it finds the best model, which is most central with regard to all its beta coefficients. This approach was presented in Kopczewska (2020, 2021) in bootstrapped spatial regression to solve big data limitations.

2.5 Clustering based on density

The above-discussed clustering procedure has three main features: (a) an algorithm used a distance matrix; (b) all points or regions were classified to one of the clusters; and (c) user assumed a priori a number of clusters. Density-based clustering differs in all those aspects. Its goal is to detect hot spots, defined as a localised excess of some incidence rate and understood as locally different density (e.g. dense and sparse areas). The implication of the hot spot approach is an automatic partitioning mechanism that assigns observations to clusters and leaves others as noise.

One of the most commonly used solutions is the DBSCAN algorithm (*density-based spatial clustering of applications with noise*) (Ester et al. 1996), which detects the local density of a point pattern. In simple terms, it screens the surroundings of each point iteratively by checking whether the minimum number of points is located in a specified radius. If yes, points are classified as the core; if not, points are classified as border points when the given point belongs to the core point radius or as noise if the point is located outside the radius of the core point. This algorithm works mostly in 2D (on the plane) or 3D (in the sphere); broader applications are rare but are slowly appearing (as 6D DBSCAN) (Czerniawski et al. 2018). What is essential is that it does not use a mutual $n \times n$ distance matrix, thus automatically increasing its efficiency in big data applications. It also does not assume any parametric distributions, cluster shapes or number of clusters and is resistant to weak connections and outliers. DBSCAN was extended in different directions, e.g. as C-DBSCAN (density-based clustering with constraints) (Ruiz et al. 2007), which controls for “Must-Link” and “Cannot-Link”, ST-DBSCAN (spatio-temporal DBSCAN) (Birant and Kut 2007), K-DBSCAN (Debnath et al. 2015) and OPTICS (Ankerst et al. 1999) for different density levels and HDBSCAN (hierarchical DBSCAN) (Campello et al. 2013) which finds epsilon automatically (Wang et al. 2019a, b). Joshi et al. (2013) have run multi-dimensional DBSCAN for polygons, in which the spatial ε -neighbourhood (points in a radius of ε) is substituted with a spatio-temporal neighbourhood. Khan et al. (2014) and Galán (2019) have reviewed the latest advances in DBSCAN and their applications.

DBSCAN has many applications. Pavlis et al. (2018) used DBSCAN to estimate the retail spatial extent. To address local variability, they used individual radii in subsets derived from a distance-constrained k -nearest neighbour adjacency list. Cai et al. (2020) estimate tropical cyclone risk with ST-DBSCAN. It can be used in astronomy, e.g. to test the spatial distribution of Taurus stars (Joncour et al. 2018), where the DBSCAN parameters were set based on correlation function and knn . It can be applied to the classification of objects from imaging with an airborne LIDAR technique (Wang et al. 2019a, b), WLAN indoor positioning accuracy (Wang et al. 2019a, b) and traffic collision risk in maritime transportation (Liu et al. 2020a, b). DBSCAN may also work with text data and computer codes. Mustakim et al. (2019) ran DBSCAN on the cosine distance obtained for text representation (frequency-inverse document frequency and vector space model) and checked partitioning quality with the silhouette. Reis and Costa (2015) clustered computer codes; they used tree edit distance (as Levenshtein distance) for strings to compare trees, which constituted the input data for DBSCAN. Their analysis clustered codes in terms of execution time, which helps in the pro-ecological selection of equivalent, but quicker codes.

Before the introduction of DBSCAN, there were a few other methods for scanning statistics, constructed based on a moving circle—GAM (*geographical analysis machine*), BNS (*Besag–Newell statistic*) and Kulldorff’s spatial scan statistics. GAM (Openshaw et al. 1987) works on point data within a rectangle and divides an area into grid cells, and for each grid, it plots a ring of the radius (radii) r specified by the user. It counts cases (e.g. disease) within a circle and makes a comparison of that number with the expected number of points from Poisson distribution

(e.g. population) or other phenomena cases. The significant circle is the output. BNS (Besag and Newell 1991) works similarly to GAM but with a pre-defined cluster size k . This means that each ring expands to reach k cases inside and is then compared with the underlying distribution. Spatial scan statistics (Kulldorff 1997) compares within the moving ring the probability of being the case given populations at risk inside and outside the ring. The ring is adaptive (up to a given percentage of total cases). However, nowadays, only Kulldorff's measure is still applied widely in epidemiological studies, while GAM and BNS have largely been forgotten. A notable progressive method stemming from GAM is a scan test for spatial group-wise heteroscedasticity in cross-sectional models (Chasco et al. 2018).

After DBSCAN,⁸ there arose a group of methods based on the Voronoi/Dirichlet tessellation (Estivill-Castro and Lee 2002; Lui et al. 2008), called Autoclust. In the Voronoi diagram, for each point, the mean and standard deviation of the tile's edges are calculated. In dense clusters, all edges are short; in the case of border points, the variance of edges increases, as one edge is significantly longer than the other. Analysis of edges and border points delineates the borders of dense clusters. The biggest advantage is that parameters (the number of clusters) are self-establishing, which is not the case with k -means or DBSCAN. This approach was also forgotten and did not become a part of machine learning due there being a lack of solutions for predictions. Recently, proposals of 3D implementations (Kim and Cho 2019) have been put forward, suggesting a revival of this method.

2.6 Overview of ML spatial clustering

The above-discussed methods differ in their approaches, but their goal is similar. In any case, one may ask the question: to which cluster does a given spatial point belong? Depending on input data, the answer may be: (i) a cluster of spatially close points; (ii) a cluster of feature-similar observations; (iii) a cluster of points that are both spatially close and have similar features; (iv) a cluster of similar regression coefficients; or (v) a cluster of densely located points. Spatial locations can be addressed directly with geo-coordinates, but can also be addressed as one of the clustered features, as a restriction in the pairing of points, as weight in optimisation, as background in running the GWR regression, or as local density (Fig. 1).

This methodological summary can have applications for many regional science problems. It can facilitate the locating of clusters of features and can map them in a smart way, so as to ascertain whether or not geographic segmentation exists and whether points (customers) are clustered. It can be applied to the analysis of (co) location patterns emerging from the values, in order to determine where our customers are, where else they visit, where to locate the business, and who the best neighbours will be. Finally, it can help to reduce multi-dimensional data.

Machine learning combines older, more established statistical concepts with new challenges. Current methodological research efforts are focused on better

⁸ After DBSCAN, a group of grid-based clustering algorithms were introduced, which are less popular. A spatial solution STatistical INformation Grid-based clustering method (STING) was proposed by Wang et al. (1997).

forecasting, improving computational efficiency (especially with big data), and finding more sophisticated approaches, such as for spatial techniques. Even if this summary aims to provide a comprehensive description of spatial clustering designs, there are still more methods to be found in the literature. One of these is cluster correspondence analysis for multiple point locations, to address the occurrence of the same event in many places (Lu and Thill 2003).

3 Econometric applications of machine learning to spatial data

Machine learning approaches to the dependency between variables are demonstrated by another class of models, which differ from traditional econometrics in following ways: (a) even if the input data (x and y) seem similar, the structure of the model itself is much less transparent; (b) as the machine learning modelling searches numerically for the best model, the forecasts are mostly much better than in classical theory- and user-feeling-driven approaches; and (c) due to data selection via boosting, sampling, bootstrapping, etc., the machine learning model can work with much bigger datasets.

There are two general groups of ML models: (a) typical regressions, which link the levels of features of variables x and y , and (b) classifiers, which detect feature levels x in observed classes y . The fact of both features x and classes y being known in supervised machine learning is in contrast to the unsupervised learning approach, which clusters data without a priori knowledge of which observation is in which group. Many spatial classification problems are as follows: from an image (e.g. pixels of a satellite photograph) features of the land are extracted (e.g. vegetation index, water index, land coverage) and geographical information added (e.g. location coordinates). Additionally, one knows the real classification (e.g. type of crops), which is to be later forecasted with the model. A common application is to teach an algorithm to determine the desired image elements by linking information from the photograph with the real class, where an image pixel is an individual observation. Subsequently, the model can detect those elements in new photographs to predict the class. This is widely applied in agriculture to distinguish between different crops, landscape and land uses (Pena and Brenning 2015). It also works in geological mapping (e.g. Cracknell and Reading 2014). Possible new applications are regional socio-economic development indicators based on night-light data or land use satellite images (e.g. Cecchini et al. 2021).

The most common machine learning classifier models are: naive Bayes (NB), k-nearest neighbours (kNN), random forests (RF), support vector machines (SVM), artificial neural networks (ANN), XGBoost (XGB) or Cubist. (Details of methods are outlined in “Appendix 1”). Recent years have also seen the introduction of so-called ensemble methods, which are combinations of the aforementioned classifiers. There are many studies on which methods perform the best (very often, it is random forest), or which ones are equivalent to classical approaches. Table 1 presents the latest studies which use the ML toolbox.

Machine learning models are not only more accurate than traditional methods, but might also be much faster. Sawada (2019) reports that applying machine learning

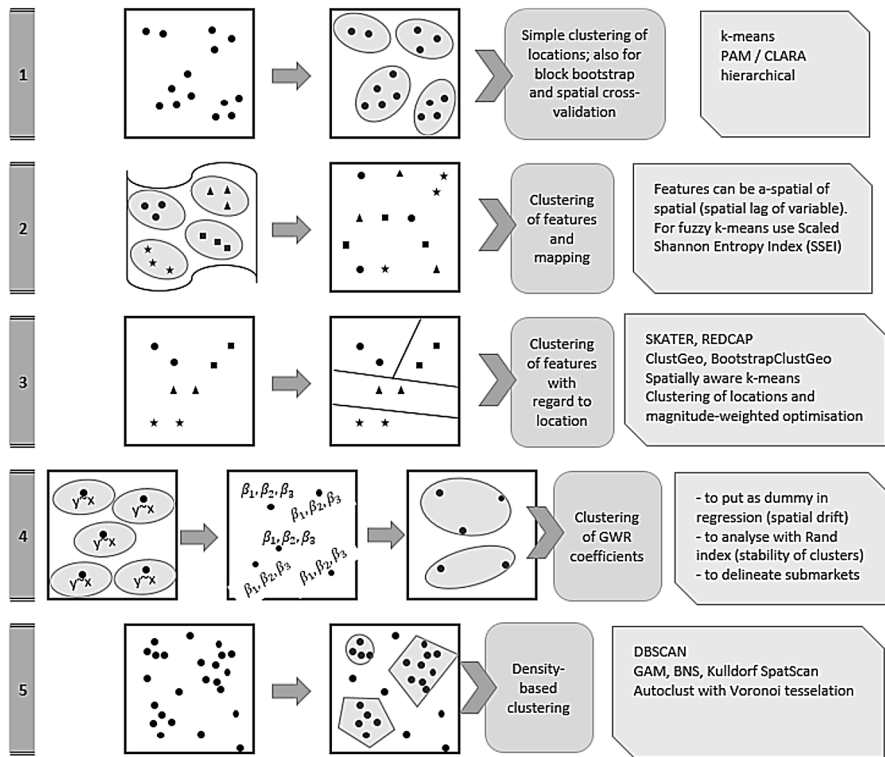


Fig. 1 Unsupervised spatial machine learning models. Source: Own concept

and the Markov chain Monte Carlo approach to a land surface model decreases computation time by 50,000 times.

It is generally agreed that most machine learning methods in spatial applications do not consider relative location and neighbourhood features and that they analyse pixels regardless of their surroundings. However, many authors have proposed various measures to address the spatial dimension, which are presented below.

3.1 Simple regression models to answer spatial questions

The most basic application of ML is to run a classification or regression model on data that is spatial in nature. Examples published in recent years apply to spatial data just as they do to any other kind of data—one understands that data are geo-projected and were gained in specific locations, but no spatial information is included. There are many examples. Appelhans et al. (2015) explained temperatures on Kilimanjaro using elevation, hill slope, aspect, sky-view factor and vegetation

Table 1 Usage of machine learning models in spatial applications (*Source*: Own study)

Type of model	Examples of usage	Thematic area	Remarks
Naïve Bayes	Park and Bae (2015)	Housing valuation	Model worked not the best, as C4.5 and AdaBoost. Much better was RIPPER
	Cracknell and Reading (2014)	Lithology classification	Model worked not the best, random forest was better
<i>k</i> -Nearest neighbours	Cracknell and Reading (2014)	Lithology classification	Model worked not the best, random forest was better
Random forest	Cracknell and Reading (2014)	Lithology classification	Model worked the best
	Meyer et al. (2019)	Land cover	Focus on selection of spatial variables and spatial CV, no other models in study
	Behrens et al. (2018)	Soil	Focus on Euclidean distance fields, model worked well. Model was compared with bagged multivariate adaptive regression splines (MARS), which also worked well
	Ahn et al. (2020)	Soil	Focus on coordinates, distances and PCA-reduced distances as covariates, model worked well
	Appelhans et al. (2015)	Temperature	Model performed well
	Liu et al. (2020a, 2020b)	Poverty	Model performed well, better than regression tree
	Hengl et al. (2018)	Soil	Focus on buffer distance, model performed well
	Goetz et al. (2015)	Landslide susceptibility	Model worked well the same as bootstrap aggregated classification trees (bundling) with penalised discriminant analysis (BPLDA)
	Li et al. (2011)	Seabed mud	Focus on mixture with kriging, model performed well
	Xu and Li (2020)	Housing valuation	Focus on stacking ensemble model, model performed well
	Hengl et al. (2017)	Soil	Model with many spatial covariates, non-spatial CV, problems of high spatial clustering of sample points; model predicts individual data which are later clustered for composite prediction, model worked well
	Pourghasemi et al. (2020)	Gully erosion	Random forest with many spatial covariates performed better than LASSO, generalised linear model (GLM), stepwise generalised linear model (SGLM), elastic net (ENET), partial least square (PLS), ridge regression, support vector machine (SVM), classification and regression trees (CART), bagged CART. No spatial cross-validation applied

Table 1 (continued)

Type of model	Examples of usage	Thematic area	Remarks
Support vector machines	Behrens et al. (2018)	Soil	Focus on radial basis function support vector machines (SVM) and on Euclidean distance fields, model performed poorly
	Goetz et al. (2015)	Landslide susceptibility	Model worked well
	Li et al. (2011)	Seabed mud	Focus on mixture with kriging, model performed not the best
	Du et al. (2020)	Land use	Strategic comparison of ML models, model performed well
	Cracknell and Reading (2014)	Lithology classification	Model worked not the best, random forest was better
Neural network	Behrens et al. (2018)	Soil	Focus on Euclidean distance fields, model averaged neural network performed poorly
	Appelhans et al. (2015)	Temperature	Model-averaged neural network performed well
	Nicolis et al. (2020)	Seismic rate	Using deep neural network—long short-term memory (LSTM) and convolutional neural networks (CNN), model worked well
	Masolele et al. (2021)	Land use	Using deep neural network in spatio-temporal application, models worked well, spatial or temporal structures can dominate depending on dataset
XGBoost	Appelhans et al. (2015)	Temperature	Focus on stochastic gradient boosting, model performed well
	Hengl et al. (2017)	Soil	Model with many spatial covariates, non-spatial CV, problems of high spatial clustering of sample points; model predicts individual data which are later clustered for composite prediction, model worked well
	Xu and Li (2020)	Housing valuation	Focus on stacking ensemble model, using adaptive boosting, gradient boosting decision tree, light gradient boosting machine and extreme gradient boosting, models performed well
Cubist	Behrens et al. (2018)	Soil	Focus on Euclidean distance fields, model worked well
	Appelhans et al. (2015)	Temperature	Cubist combined with residual kriging performed well

index data—they used machine learning models in a regression, with the only spatial issue being spatial interpolation with kriging.⁹ Similarly, Liu et al. (2020a, b) ran non-spatial regression and a random forest model on socio-economic and environmental variables to explain poverty in Yunyang, China, using data from 348 villages. The only computational spatial component was the Moran test of residuals, which showed no evidence of spatial autocorrelation. The study was effective because it merged different sources of geo-projected data: surface data for elevation, slope, land cover types and natural disasters (with spatial resolution of 30 m or 1:2000); point data, such as access to town, markets, hospitals, bank, schools, or industry, taken from POI (point-of-interest) or road density networks (on a scale of 1:120,000); and polygonal data for the labour force from a statistical office. Rodríguez-Pérez et al. (2020) modelled lightning-triggered fires in geo-located grid cells in Spain. They used RF, a generalised additive model (GAM) and spatial models to show instances of lightning-triggered fires appearing in a given grid-cell were attributable to observable features in that location, such as vegetation type and structure, terrain, climate and lightning characteristics. Also, an applied example of statistical learning in a book by Lovelace et al. (2019) uses a generalised linear model on rastered data of landslides (e.g. slope, elevation) with point data of interest. The spatial location and autocorrelation are included in spatial cross-validation.

Another interesting example is the mapping of rural workers' health conditions and exposure to severe disease (Gerassis et al. 2020) using a ML approach. The study is based on geo-located medical interviews which provided health data—both hard medical data and the person's general health condition. Using a ML Bayesian network (BN), the authors discovered which variables were connected with the patient's condition when they were flagged as ill. In the next step, with binary logistic regression run on individual cases and thresholds from the BN, model classification was obtained, and predictions of high disease risk for a person could be made. Spatial methods appear only for interpolation of illness cases observed, which is a separate model—Gerassis et al. (2020) used the point-to-area Poisson kriging model, which deals with spatial count data, unequal territories and diverse population composition. The spatial challenge was in the different granulation of data: point data in the study sample and polygonal data as a basis of prediction.

3.2 Spatial cross-validation

Current implementations of machine learning in the spatial context are often restricted to spatial k -fold cross-validation (CV) only, which can solve the issue of non-independence. This works by dividing points into irregular k clusters (by using k -means, for example) and selecting one cluster as an out-of-sample cross-validation part. Due to spatial autocorrelation between training and testing observations, simple spatial data sampling gives biased and over-optimistic predictions. However, spatial CV increases prediction error (Liu 2020). Lovelace et al. (2019) show that a spatially cross-validated model gives a lower AUROC (*area under the*

⁹ Kriging, which is often a part of ML modelling, is also the best imputation method in the case of missing data (Griffith & Liao, 2020).

receiver operator characteristic curve), as it is not biased by spatial autocorrelation. The same applies to models that tune hyper-parameters (e.g. SVM) using sampling (Schratz et al. 2019). In the case of spatio-temporal data, one should account for spatial and temporal autocorrelation when doing CV (Meyer et al. 2018). Spatial cross-validation is becoming a standard (e.g. Goetz et al. 2015; Meyer et al. 2019), but some studies still neglect this effect and do not address the autocorrelation problem (Park and Bae 2015; Xu and Li 2020).

3.3 Image recognition in spatial classification tasks

One of the typical applications of ML is image recognition in spatial classification tasks. A good example is supervised lithology classification, i.e. geological mapping (Cracknell and Reading 2014). As input (X), data from airborne geophysics and multispectral satellites are used, while as output (Y) for a given territory, the known lithology classification is used, shown as polygons on the image for each class. The xy coordinates of the pixels of those images are also known. In the modelling process, an algorithm is produced which discovers the lithology classification from airborne geophysics and multispectral satellites. Three kinds of models are run on pixel data: (i) $X \rightarrow Y$, (ii) $xy \text{ coords} \rightarrow Y$ and (iii) X and $xy \text{ coords} \rightarrow Y$, using the aforementioned NB, kNN, RF, SVM and ANN algorithms. In fact, this is an image processing phase, in which software is taught to understand what is in the picture, and each pixel is classified according to lithology. The goodness of fit and prediction differ between models. ML produces the model, which will generate a lithology classification when fed with new satellite and airborne data. A similar study was conducted by Chen et al. (2017), who used the following 11 conditioning factors to predict landslide data: elevation, slope degree, slope aspect, profile and plan curvatures, topographic wetness index, distance to roads, distance to rivers, normalised difference vegetation index, land use, land cover and lithology. They used maximum entropy, neural networks, SVM and their ensembles.

A very different approach is involved when dealing with dynamic spatial data. Nicolis et al. (2020) modelled earthquakes in Chile. Their dataset of seismic events spanned a period of 17 years, with 86,000 geo-located cases occurring over 6,575 days. For each day that an earthquake was recorded, they created a grid-based image ($1^\circ \times 1^\circ$) of the territory; grid intensity was estimated by an ETAS (*epidemic-type aftershock sequences*) model. Using this, they applied deep learning methods, such as long short-term memory (LSTM) and convolutional neural networks (CNN) for spatial earthquake predictions—predicting the maximum intensity and the probability that this maximum will be in a given grid cell.

Images as predictors in spatial models are not always informative. Fourcade et al. (2018) proved that images that are meaningless for spatial process such as paintings or faces can predict environmental phenomena well. This finding formed the basis of deeper studies (Behrens and Rossel, 2020) which reached two major conclusions. Firstly, spurious correlations without causality raise the danger of meaningless but efficient predictors, which can be mitigated by using domain-relevant and structurally related data. Secondly, by comparing the variograms of regressors, it was

recommended to use covariates with the same or a narrower range of spatial dependence than the dependent variable. Meyer et al. (2019) have similarly concluded that highly correlated covariates result in over-fitted models, which replicate data well and fail in spatial predictions.

3.4 Mixtures of GWR and machine learning models

An example of development and adaptation of traditional methods is the transformation of geographically weighted regression (GWR) into a machine learning solution. The process behind GWR lies in applying small local regressions to neighbouring points for each observation instead of one global estimation. Additional factors to consider are: (i) the radius and shape of the “moving geometry” (e.g. circle, ellipse), which indicates which points to include in a given local regression; (ii) its flexibility—fixed kernel for a fixed radius and adaptive kernel for a changing radius to react to various densities of the point data; and (iii) the weighting scheme—whether observations included in local regressions have the same weight when distance-decaying from the core point. These features of GWR can be applied to any machine learning model. Li (2019) mixed GWR with neural networks, XGB and RF to improve wind speed predictions in China by more effectively capturing local variability. It gave a 12–16% improvement in R^2 and a decrease in RMSE (root mean square error). Quiñones et al. (2021) applied GWR concept with RF in analysing diabetes prevalence and showed that it detects well the spatial heterogeneity.

According to Fotheringham et al. (2017), traditional GWR should be replaced by multiscale geographically weighted regression (MGWR). In MGWR, the user decides on bandwidth not only with regard to location/local density but allows for optimisation of covariate-specific bandwidth. The performance of MGWR surpasses that of simple GWR. In both approaches, the problem of bias when “borrowing” data from territories with a different local process is negligible (Yu et al. 2020).¹⁰

3.5 Spatial variables in machine learning models

It has become very popular to replace geo-statistical models with machine learning solutions in order to model and interpolate spatial point patterns. In fact, the current literature makes comparisons between geo-statistical models (such as regression kriging and geographically weighted regression), between prediction models (ordinary kriging and indicator kriging, for example) and between multiscale methods (such as ConMap and ConStat). It also compares contextual spatial modelling with ML models.

Over the last decade, researchers have been looking for the best model for spatial interpolation. The most straightforward approach, introduced in early studies (as Li et al. 2011), simply monitors the efficiency of non-spatial ML models in spatial tasks. Mostly, they have combined RF or SVM with ordinary kriging or inverse

¹⁰ Geographical and Temporal Weighted Regression (GTWR) is also used, to address time series (Fotheringham et al. 2015).

distance squared. Random forest was often proven to be the most accurate method, which increased its popularity in further studies. This approach is still used. For example, Sergeev et al. (2019) predicted the spatial distribution of heavy metals in soil in Russia by applying a hybrid approach: they simulated a general nonlinear trend using an artificial neural network (ANN) (by applying the generalised regression neural network and multilayer perceptron) and fine-tuned the residuals with the classical geo-statistical model (residuals kriging).¹¹

Later solutions have aimed to include spatial components among covariates, namely coordinates or distances between other points. Hengl et al. (2018) have promoted the use of a buffer distance among covariates of random forest. The buffer distance is calculated between each point of the territory and observed points. It can be a distance to a given point or a distance to low, medium or high values. Hengl et al. (2018) give a number of empirical examples to show that this solution is equivalent to regression kriging but is more flexible in terms of specification and allows for better predictions. Buffer distance is used to address spatial autocorrelation between observations and works better than the inclusion of geographical coordinates. Another example of this is in Ahn et al. (2020), who used the random forest model with spatial information to predict zinc concentration, having only its geo-location to work with. They considered PCA reduction of dimensions in distance vectors and used kriging for expanding predictions on new locations. They highlighted a trade-off between the inclusion of coordinates (which give lower model precision and do not allow for the controlling of spatial autocorrelation, but which do not overload computational efficiency) and the inclusion of the distance matrix (which works in the opposite manner). They showed that the best solution is to use PCA-reduced distance vectors, which limit the complexity and improved estimation performance. An alternative is to add spatial lag and/or eigenvector spatial filtering (ESF), which can deal with most autocorrelation issues (Liu 2020). The proposals of Ahn et al. (2020) and Liu (2020) may lead to an increased number of applications of random forest for spatial data, as it works for predicting 2D continuous variables with and without covariates, as well as binominal and categorical variables, and can effectively address extreme values and spatio-temporal and multivariate problems (Hengl et al. 2018). In general, random forest, compared with geo-statistical models, requires fewer spatial assumptions and performs better with big data.

An alternative approach to including spatial components is using Euclidean distance fields (EDF), which address non-stationarity and spatial autocorrelation and improve predictions (e.g. in soil studies) (Behrens et al. 2018). These are features of analysed territory generated in GIS. Typically, for each point of territory seven EDF covariates are derived: X and Y coordinates, the distances to the corners of a rectangle around the sample set and the distance to the centre location of the sample set. They prove that as long as spatial regressors have a narrower range of spatial dependence than the dependent variable, they improve the model.

¹¹ They also use many prediction quality measures such as correlation, R^2 , RMSE, Willmott's index of agreement and a ratio of performance to interquartile distance (RPIQ) between the prediction and raw test data.

The selection of spatial variables to the model is still ambiguous. In many papers, all collected variables are included, with trust that ML methods by their nature will eliminate the redundant ones. Some studies propose running standard a-spatial algorithms as BORUTA (Amiri et al. 2019) to indicate which variables should stay in the model. There are also proposals for the removal of correlated covariates and regularisation to cope with multi-collinearity (Farrell et al. 2019); these actions will not significantly impact the results—random forest showed the best performance on raw data; however, spatial autocorrelation was not addressed. There are also some controversies. Meyer et al. (2019) assessed the inclusion of spatial covariates using quality measures such as kappa or RMSE. They claim that longitude, latitude, elevation and the Euclidean distances (also as EDF) can be unimportant or even counterproductive in spatial modelling, and they recommend that those regressors be eliminated from models. They highlighted two other aspects: Firstly, contrary to the popular narrative, they do not approve of the high fit of ML models, treating them as over-optimistic and misleading; secondly, they claim that in the course of visual inspection, one observes artificial linear predictions resulting from the inclusion of longitude and latitude, and that their elimination helps in making predictions real.

3.6 Overview of spatial ML regression and classification models

The above-described modelling patterns can be summarised in a general framework, which consists of four stages: data integration, data modelling, model fine-tuning and prediction (Fig. 2). All of them include spatial components.

1. **Data integration** The central focus of many current spatial machine learning studies is in the integration spatial data in different formats. As a standard, one uses geo-located points (for observation location, point-of-interest, etc.), irregular polygons (for statistical data), regular polygons such as grids or rasters (for summed or averaged data within that cell), lines (such as rivers or roads) and images (such as satellite photographs, spectral data, digital elevation models, vegetation and green leaf indices, etc.). There is a diverse range of forms of individual observation: point, polygon, grid or pixel. Depending on the researcher's choice of data target granulation, the dataset integration process may be only technical or may involve more or less advanced statistical methods. For classification purposes, the researchers may add the classes of objects manually.
2. **Modelling** Machine learning methods differ from econometric¹² algorithms when obtaining a mutual relationship between the dependent (y) and explanatory (x) data. Regression models are used to explain usually continuous variables, while classification models are used for categorical variables. ML models for spatial

¹² Due to its inherited spatial weights matrix, spatial econometrics, deals with neighbourhood, tracks the spillover and importance of relative location, and technically improves the quality of estimation by reducing bias and improving consistency. By adding distance variables one controls for distance-decay patterns and spatial interactions. Dummies for specific location (such as the Central Business District, on the border, at the seaside, in the main city) measure the effect of absolute location and special spatial features.

data have mostly neglected the issue of spatial autocorrelation between observations. The latest studies, however, have aimed to address this issue by using spatial variables among covariates. These can be geo-coordinates, distance to a given point (e.g. core), mutual distances between observations, PCA-reduced mutual distances between variables, buffer distance, spatial lag of the variable and eigenvector or Euclidean distance fields. Addressing the spatial autocorrelation issue not only enables the training data to be successfully reproduced well but also allows predictions to be made in new locations beyond the dataset (Meyer et al. 2019). GWR-like local machine learning regression bridges the gap between spatial and ML modelling. This stage results in sets of global or local regression coefficients or thresholds of decision trees.

3. *Model fine-tuning* The common approach is to test and improve model estimation with k -fold cross-validation. For a long time, many scientists reported excellent performance among ML models when testing them on fully randomly sampled observations. The current literature suggests that not addressing autocorrelation falsely improves the quality of the model, and they recommend spatial cross-validation to overcome this—it takes as folds the k -means spatially continuous clusters of data. The other option is classical testing of spatial autocorrelation of model residuals (e.g. Moran's I) and re-estimation of whether the spatial pattern is found.
4. *Prediction* The majority of ML studies are oriented towards predictions based on the model. In regression tasks, they often use one of the kriging variants, which expands results from observations on all possible points within the analysed territory. In classification tasks primarily based on pixel data, the calibrated models are fed with a new image that enables running prediction for all input pixels.

The general overview from the literature is that visible progress has been made in the development of spatial machine learning modelling. Over the last decade, the following approaches have been developed:

1. Classic ML + non-spatial variables + random cross-validation,
2. Classic ML + spatial all variables + random cross-validation,
3. Classic ML + spatial all variables + spatial cross-validation,
4. Classic ML + spatial selected variables + spatial cross-validation,
5. Spatial ML + spatial selected variables + spatial cross-validation.

The current standard of modelling is expressed by approach “(4) Classic ML + spatial selected variables + spatial cross-validation”. Models estimated with approaches (1), (2) or (3) may not be fully reliable, due to the autocorrelation issues discussed above. The progress and innovations in the development

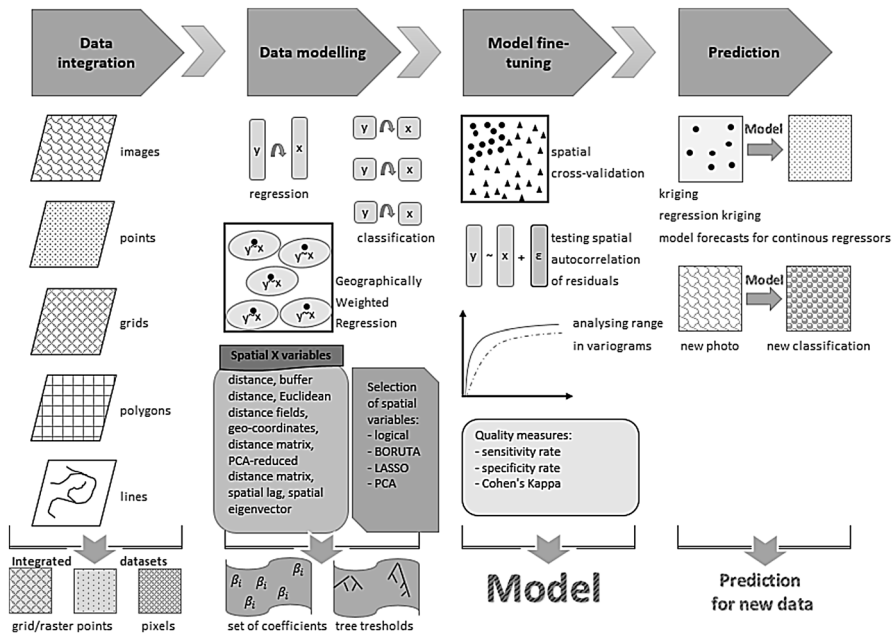


Fig. 2 Spatial machine learning modelling. Source: Own concept

of approach (5) are mostly concerned with the formulation of ML methods to incorporate spatial components into the algorithms.

It is clear from many studies that unaddressed spatial autocorrelation generates problems, such as overoptimistic fit of models, omitted information and/or biased (suboptimal) prediction. Thus, an up-to-date toolbox dealing with spatial autocorrelation should be used in all ML models in order to ensure methodological appropriateness. One can mention here methods such as (i) adding spatial variables as covariates; (ii) GWR-like local ML regressions; (iii) using spatial cross-validation; (iv) testing for spatial autocorrelation in model residuals; v) running spatial models on grids or pixels with a spatial weights matrix W ; and (vi) running spatial predictions with kriging. To sum up, the spatial dimension and spatial autocorrelation can be addressed at each stage of the modelling process, and combinations of these solutions seem to improve the quality of models. ML algorithms are often more efficient than classical spatial econometric models, which renders them more effective in dealing with big spatial data.

4 Perspectives of spatial machine learning

The methodological solutions presented above open up new pathways for advanced research using spatial and geo-located data.

Firstly, these methods enable more efficient computation in the case of big data and inclusion of new sources of information. Switching from regional data to a finer

degree of data granulation—such as individual points or pixels of the image—brings about a significant increase in the magnitude of datasets. This higher level of granulation is especially troublesome for a classical spatial econometrics based on an $n \times n$ spatial weights matrix W or $n \times n$ distance matrix. As indicated by Arbia et al. (2019), the maximum size of the dataset for computation with personal computers is around 70,000, while even with 30,000 observations, the creation of W is already challenging (Kopczewska 2021). ML models, which are free of W , are automatically quicker, but the issue of autocorrelation, currently treated as critical, is addressed in another way. New sources of data, such as lightmaps of terrain (Night Earth, Europe At Night, NASA, etc.) or day photographs of landscape (Google Maps, Street View, etc.), bring new insights and information and are useful due to the robustness of their big-data analytics (see “Appendix 3”). Spatial data handling (e.g. processing remote sensing image classification or spectral–spatial classification, executed with supervised learning algorithms, ensemble and deep learning) is especially helpful in big data tasks (Du et al. 2020).

Secondly, the methods present a way to address spatial heterogeneity and isotropy. Classical spatial econometrics was focused on spatial autocorrelation and mostly neglected other problems. Local regressions, combined with global ones, help in capturing unstable spatial patterns. The overview of methods shows that integration of classical statistics and econometrics with machine learning enables more tools to be added to the modelling toolbox than with a single approach.

Thirdly, the methods open up possibilities for spatio-temporal modelling and for studies of the similarities between different layers: spatial, multi-dimensional and spatio-temporal, among others. The dynamics connected to location can be addressed in more ways than just the classical panel model. One of the approaches is to run similar to PCA method EOF (empirical orthogonal functions) decomposition (Amato et al. 2020).

Fourthly, these methods allow for better forecasting due to inherited boosting and bootstrapping in ML algorithms. ML results are also more flexible for spatial expansion into new points. Ensemble methods, popular in ML, are enabling researchers to make the best prediction. A shift from spatial econometrics towards spatial ML also represents a move from explanation to forecasting. The predictive power of classical spatial models was rather limited (Goulard et al. 2017), mostly due to simultaneity in spatial lag models. The second problem was that out-of-sample data were not included in W and therefore impossible to cover with the forecast. New solutions such as ML spatial prediction can be fine-tuned in line with spatial econometric predictions based on bootstrapping models (Kopczewska 2021).

Fifthly, these methods foster the development of new innovations, such as indicators based on vegetation or light indices. The methods presented also introduce 3D solutions to certain areas of spatial studies, such as social topography (with 3D spatial inequalities) (Aharon-Gutman et al. 2018; Aharon-Gutman and Burg 2019), 3D building information models (Zhou et al. 2019) or urban compactness growth (Koziatek and Dragićević 2019). There are urban studies that rely on information from Google Street View, such as those that count cars, pedestrians, bikers, etc., to predict traffic (Goel et al. 2018), or those recording indicators of urban disorder (such as cigarette butts, trash, empty bottles, graffiti abandoned cars and houses) to

predict neighbourhood degeneration (Marco et al. 2017) or studies that count green vegetation indices in order to predict degrees of safety (Li et al. 2015).

All of the above demonstrates that spatial modelling built on econometrics, statistics and machine learning is the most effective approach. It has wide-ranging applications in such areas as epidemiology, health, crime, the safety of the surrounding area, location of customers, business, real estate valuation, socio-economic development and environmental impact, among many others.

In addition to all of this, the ML approach can still provide the answer to common questions, which have been asked over recent years in quantitative regional studies. On the one hand, these studies are designed to examine invisible policies and their impact on observable phenomena—by studying policy flows, core-periphery patterns and their persistence, urban sprawl patterns, diffusion and spillover from the core to the periphery, cohesion and convergence mechanisms, institutional rent, effects of administrative division, the role of infrastructure and the effects of agglomeration. On the other hand, these studies may be of an opposite nature, that is, analysing visible spatial patterns to draw conclusions about unobservable phenomena, such as studying clusters, tangible flows such as trade or migrations, similarity and dissimilarity of locations, spatio-temporal trends, spatial regularities in labour markets, GDP and its growth, education, location and movements of customers, and business development, location and co-location. In those studies, questions on spatial accessibility, spatial concentration and agglomeration, spatial separation, spatial interactions and spatial range have mostly been answered.

Progress in science over the past decades has involved the interdisciplinary transfers of knowledge and methods. Regional science is yet to experience such a transfer. The general findings presented in recent papers would suggest that it has already begun (with first literature reviews on ML for spatial data by Nikparvar and Thill (2021)), but the regional science field still awaits mass interest from researchers.

5 Conclusions

This paper shows that, even if universal in terms of algorithms used, machine learning (ML) solutions are very specific in field applications. ML design in regional science presented above differs from designs in genetics and genomics (Libbrecht and Noble 2015), medicine (Fatima and Pasha 2017), robotics (Kober et al. 2013), neuroimaging (Kohoutová et al. 2020), etc. When applying ML in their fields, researchers should use general over-disciplinary knowledge as a basis and fine-tune their approach with field-specific solutions. This paper makes guidelines for regional and spatial analysts to better understand better how to include spatial aspects, geographical location and neighbourhood relations into the models.

The complexity of ML modelling finds its reflection in transparency and reproducibility. In many disciplines that heavily use ML, community-driven standards of ML reporting appear. This can be found in biology (Nature Editorial 2021a), for which one proposes DOME (Walsh et al. 2021) and AImE (Matschinske et al. 2021) guidelines for reporting ML results. Present also are general recommendations to increase

the transparency of all necessary steps in computations. It is important that it does not refer to the modelling phase only, but also to data curation, generation and division into training, validation and test datasets. It also covers data, code and model availability (Nature Editorial 2021b). Within life sciences, one considers three standards of reporting, Bronze, Silver and Gold, which differ in rigour for computational reproducibility (Heil et al. 2021). Bioinformatics proposes workflow managers, which are ready-to-use environments assuring shareable, scalable and reproducible biomedical research (Wratton et al. 2021). One can expect those solutions also in regional science to appear soon. However, the specificity of spatial data will require adjusted reporting standards, which may address the geographical information systems (GIS) issues.

Machine learning, which is booming in all computational disciplines, has become a new analytical standard, as OLS or p value was until now. The interdisciplinary dialogue requires using the same language; thus, it is a must essential to accept, use and develop ML methods in regional science research. The presented overview shows that many developments targeted towards spatial data and regional science problems are already available. However, some methodological gaps need new ideas dedicated to regional science problems. The biggest challenge is using information from the neighbourhood still (Hagenauer et al. 2019), accounting for spatial autocorrelation and heterogeneity, using information from satellites and photographs, predicting for new locations and scalability of methods.

Appendix 1: Overview of quantitative concepts

Below is a description of the methods mentioned in the paper—distance metrics, clustering with k -means, PAM and CLARA, hierarchical clustering, spatial clustering with SKATER and REDCAP, DBSCAN clustering, clustering quality measures (silhouette, inertia, Dunn index), k -fold cross-validation, typology of supervised machine learning methods, Naïve Bayes classifier, K -nearest neighbours classifier, random forest classifier, support vector machines, artificial neural networks, maximum entropy classifier, autoencoder-based residual network, gradient boosting and Cubist.

Distance metrics

Clustering algorithms, which are based on mutual distance between points, use different metrics of distance. For two points $X = (x_1, x_2, x_3, \dots, x_n)$ and $Y = (y_1, y_2, y_3, \dots, y_n)$, one can define (Fig. 3):

- *Euclidean distance* $\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$, which measures the shortest way between points. It compares pairs of observations, variable by variable, and calculates the square root of the sum of squares of differences between values of variables.
- *Manhattan (urban, city-block) distance* $\sum_{i=1}^n |x_i - y_i|$, also called urban distance, which uses perpendicular sections to connect points as if moving around the edges of the grid. It compares pairs of observations, variable by variable, and calculates the absolute difference of their values, which is summed up.

- *Minkowski distance* $\sum_{i=1}^n (|x_i - y_i|^p)^{1/p}$, which is a generalisation of Euclidean and Manhattan distances and allows for a nonlinear, curved way between points.

In addition to the above three metrics, one can use the following concepts:

- *Gower distance (also Gower dissimilarity)* introduced by Gower (1971), can be applied to a mix of numerical and categorical variables. It compares pairs of observations, variable by variable, and calculates the average distance score between those observations. Components of the score are from range [0,1] and also their average. For quantitative variables, the score is the absolute value of difference between values of observations divided by the variable range: $|x_i - x_j| / (\max(x) - \min(x))$. For qualitative variables, it gives 0 if they are the same and 1 if they are different. Low values of Gower distance represent higher degrees of similarity.¹³
- *Mahalanobis distance* introduced by Mahalanobis (1936), includes correlations between variables $\sqrt{(x - y)^T \text{cov}(x, y)^{-1} (x - y)}$. To calculate this distance, one follows the procedure¹⁴ given in the following:
 - Take real data (for example, three variables x, y, z) and calculate the average values of each variable; this will give you the vector of (three) average values $(\bar{x}, \bar{y}, \bar{z})$.
 - Take your test data (let us say $x_i = 1, y_i = 4, z_i = 6$).
 - Calculate the vector of differences between your test data and vector of average values $(x_i - \bar{x}, y_i - \bar{y}, z_i - \bar{z}) = (1 - \bar{x}, 4 - \bar{y}, 6 - \bar{z})$; this is a vector of differences from mean values.
 - Calculate the variance–covariance matrix of your data—this will give you a 3×3 matrix. Make an inverse of it.
 - Multiply (as matrix) the vector of differences by the inverse covariance matrix by the vector of differences.
 - Take a square root of this multiplication; this is the Mahalanobis distance.
- *Hamming distance* introduced by Hamming (1950) to compare binary vectors; it gives 0 if elements are the same, and 1 if they are different, and adds up the scores; this counts in how many points the vectors differ. As with the Gower distance for qualitative data, it compares pairs of observations, variable by variable.
- *Cosine distance (cosine similarity)* measures the angle of two vectors. In case of similarity, the angle is 0° and $\cos(0^\circ) = 1$. In cases of dissimilarity, the angle of two vectors increases, and its cosine is in range [0,1]. Two vectors being opposite have a distance of -1 . Cosine distance is expressed as: $\frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$, where A and B are the analysed vectors (variables). The counter is the sum of products of paired values of both variables. The nominator is the total of the squared values of both variables.

¹³ <https://jamesmccaffrey.wordpress.com/2020/04/21/example-of-calculating-the-gower-distance/>.

¹⁴ <https://jamesmccaffrey.wordpress.com/2017/11/09/example-of-calculating-the-mahalanobis-distance/>.

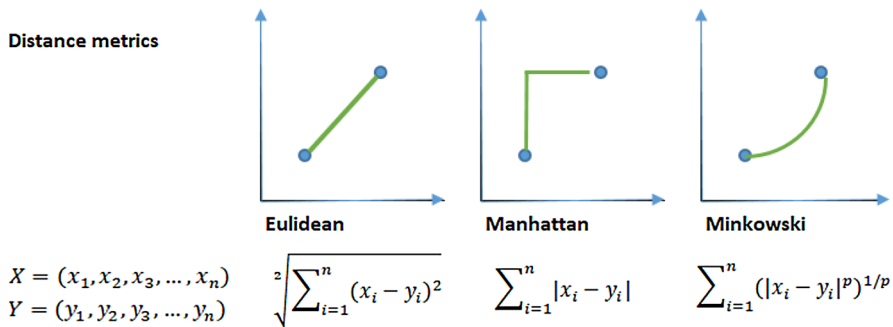


Fig. 3 Distance metrics. Source: Own work

- *Cophenetic distance* proposed by Sokal and Rohlf (1962), applied only to hierarchical clustering. It measures the height of the dendrogram between two clusters, or more precisely, the height of the dendrogram where the two branches that include the two objects merge into a single branch.
- *Levenshtein distance* introduced by Levenshtein (1965), also known as edit distance, mostly used in text analysis. It reflects the minimum number of corrections (delete, insert, substitute) necessary to change one vector into another.¹⁵

Clustering with k-means

The idea of *k*-means was introduced by Steinhaus (1956), the first algorithm was developed by Lloyd (1957), while the term *k*-means was proposed by MacQueen (1967) (see Bock 2007). With the *k*-means method, a number of clusters *k* are assumed a priori, initial multi-dimensional coordinates of the *k* centroids are set, the matrix of distances between all sample points and *k* centroids is calculated, and finally, the location of centroids is optimised by minimising the total distance of points from cores (Fig. 4). Location of centroids is non-restricted and can be in any place on the plane (surface) where the sample data are located. All points are assigned to clusters.

Clustering with PAM and CLARA

The idea of clustering with PAM (*partitioning around medoids*) was introduced by Kaufman and Rousseeuw (1987). As with *k*-means, *k* core points are assumed a priori. However, they are not selected freely as in *k*-means, but must belong to the sample. Finding the best combination of *k* points which become medoids minimising the total distance of points from cores requires an iterative approach (Fig. 5). All points are assigned to clusters.

The CLARA (*clustering large applications*) method is the big data equivalent of PAM. It was proposed by Kaufman and Rousseeuw (1990). It works in the same way as PAM but on a subsample, which classifies points into clusters.

¹⁵ <https://www.baeldung.com/cs/levenshtein-distance-computation>.

The rest of the points are assigned to clusters using the k -nearest neighbours algorithm.

Hierarchical (agglomerative) clustering

Hierarchical clustering was introduced by Breiman et al. (1984). It assumes continuous clustering which can be selected after division. The bottom-up algorithm starts with all observations constituting their own clusters—singletons. The clusters are iteratively merged in bigger groups. In the last stage, all observations belong to one cluster. This division can be presented in a dendrogram. To read an output, one can decide how many clusters to see or at which height to cut the tree. All observations are assigned to some clusters (Fig. 6).

Spatial clustering with SKATER and REDCAP

The SKATER (*spatial “K”luster analysis by tree edge removal*) algorithm was proposed by Assunção et al. (2006). It is based on the pruning of trees constructed as a weighted connectivity graph with edges and nodes. It clusters the values with regard to their location. Clusters of similar values are expected to be located next to each other. For each region, it makes a list of contiguous neighbours, and for each neighbour, it calculates the cost, that is, the total distance between all variables attached to areas. For each region, an algorithm chooses two closest neighbours (in terms of data) and finally groups areas into the most coherent spatially continuous clusters.

The REDCAP (*regionalisation with dynamically constrained agglomerative clustering and partitioning*) algorithm was developed by Guo (2008) as an extension of SKATER. It uses a hierarchical agglomeration method with spatial constraints. It applies three criteria for defining the distance between values [single linkage, average linkage and complete linkage (Fig. 7)] and two “constraining strategies” with regard to spatial location: first-order neighbourhood (sharing a common border) or full-order neighbourhood (having links to all other regions).

DBSCAN clustering

DBSCAN (*density-based spatial clustering of applications with noise*) was proposed by Ester et al. (1996). It does not use distance metrics and nearest neighbours (as with PAM, for example), but examines the spatial density of points to determine dense and sparse areas. The algorithm sets clusters one by one. Starting from a randomly chosen point, it examines the neighbourhood in a given radius ϵ and marks the points belonging to the cluster and the points which constitute noise. All points belonging to the cluster are iteratively tested, and the full cluster is formed. In the same procedure, points which constitute noise against the previously formed cluster are subsequently examined. Points can belong to a cluster (core and border) or stay outside the cluster (noise). DBSCAN requires that

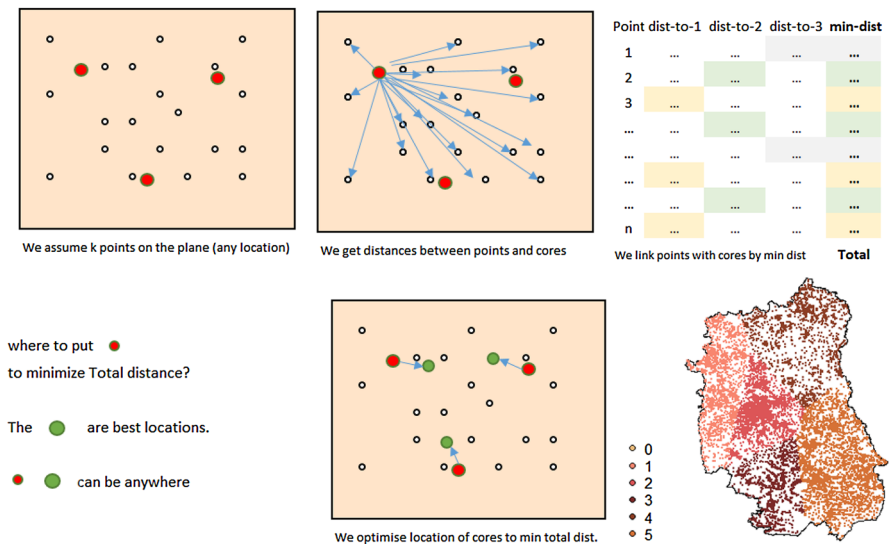


Fig. 4 Clustering with k -means. *Source:* Own work

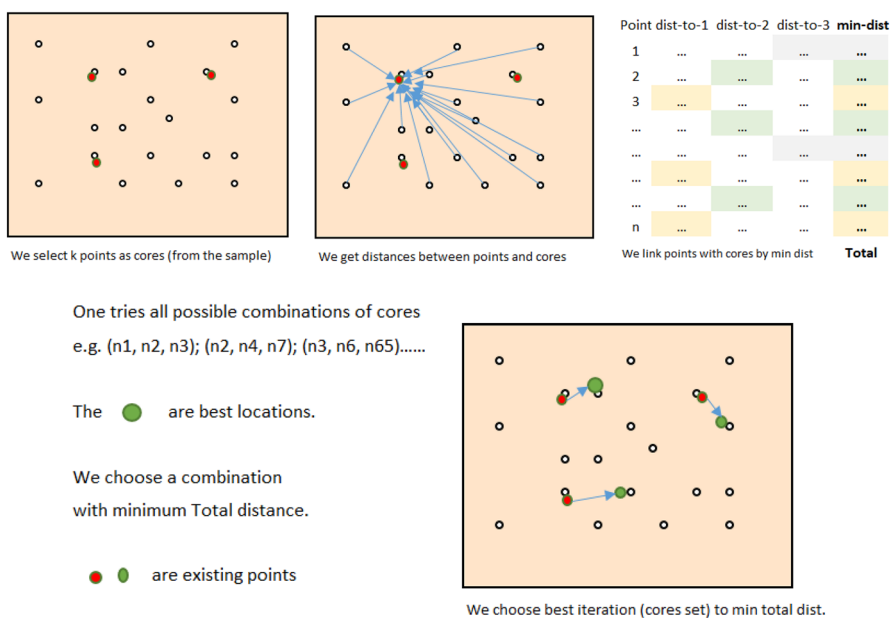


Fig. 5 Clustering with PAM. *Source:* Own work

the radius of epsilon ϵ and the minimum number of points in this radius $MinPts$ are established. For each point, one counts the number of points in radius ϵ and checks whether the points fall into the radius of other points. Core points have at

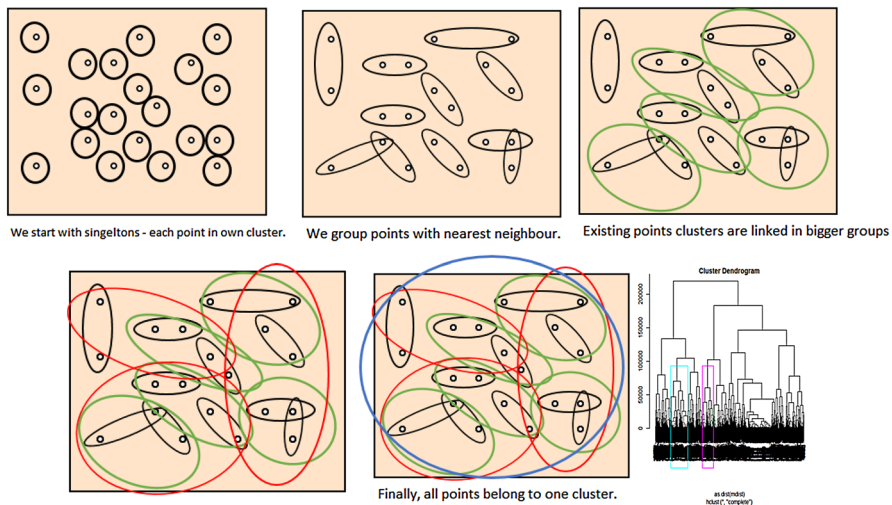


Fig. 6 Hierarchical clustering. Source: Own work

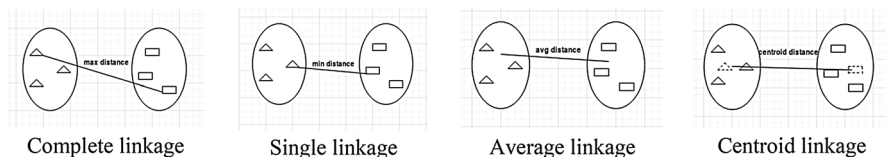


Fig. 7 Definitions of distances between clusters. Source: <https://www.datacamp.com/community/tutorials/hierarchical-clustering-R>

least the minimum number of points (*MinPts*) within a radius of ϵ . Border points are within the radius ϵ from the core point, but do not themselves contain *MinPts* points in their radius ϵ . Noise points are outside the radius of core and boundary points (Fig. 8). Sensitivity analysis is conducted with the number of clusters and percentage of noise depending on ϵ and *MinPts*. Even if the method is known as “unsupervised”, it requires the setting of two parameters by the researcher, which are crucial for the result (Fig. 9).

Clustering quality measure: silhouette

Silhouette statistics are used to test the quality of clustering, in particular, if the number of clusters k was set properly. The individual statistic S_i is given by a formula $S_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}$, where a_i is the average distance from the point to all other objects in the cluster, while b_i is the minimum average distance from the point to other clusters (tested for each cluster separately). Global S is given as $S = \frac{\sum_{i=1}^n S_i}{n}$ (averaged individual S_i). S_i and S statistics are limited $s \in [-1, 1]$. The negative

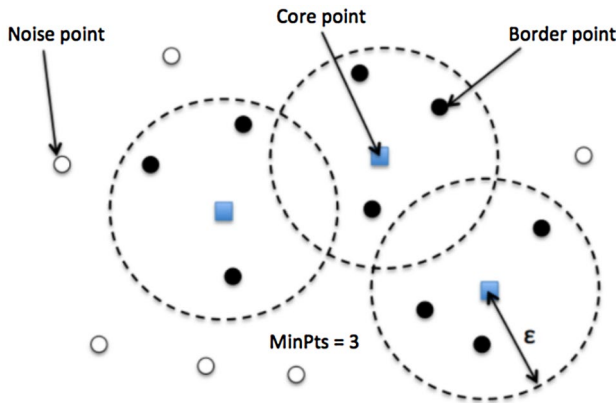


Fig. 8 Search algorithm of DBSCAN method. Source: <https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html>

values of the silhouette statistics are undesirable, because it means that $a_i > b_i$, so the objects in another cluster are closer than the objects in the same cluster. On the contrary, positive values of the silhouette statistics are desirable. The optimal value of S_i and S statistics is close to 1 ($s \sim 1$), which occurs when the distance between the observation and the middle point in the same cluster is minimal. In the interpretation, one looks for the highest values of the *silhouette* statistics for a different number of clusters k .

Clustering quality measure: inertia

Inertia for clusters is a concept similar to analysis of variance and is helpful in deciding which number of clusters works the best. It calculates the sum of the weighted squared distances between observations and their cluster centre (within-cluster inertia, W), between centres of clusters and all observations (between-cluster inertia, B) and between observations and the centre of all observations

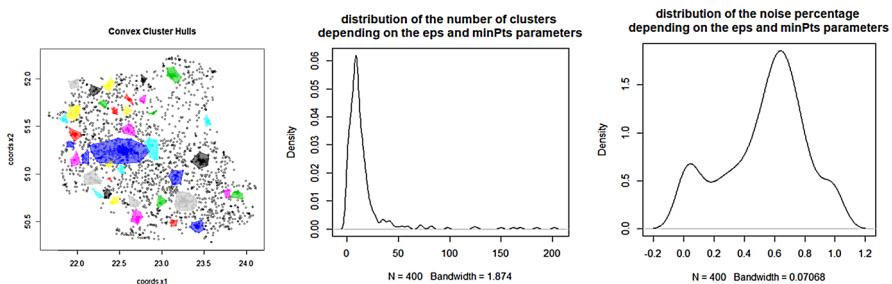


Fig. 9 Results of DBSCAN: **a** geographical clusters, **b** distribution of number of clusters depending on ϵ and minPts, **c** distribution of noise percentage depending on ϵ and minPts. Source: Own work

(total inertia, T). Good clustering is characterised by high inter-cluster inertia (diversity) and low intra-cluster inertia (heterogeneity). For two partitions, one compares their Q_s ($Q = 1 - W/T$) and chooses the partition with higher Q .

Within-cluster (intra-cluster) inertia W , assuming the existence of a P_K partition, is the sum of $I(C_K)$ inertia in all available K ($k = 1, \dots, K$) clusters and is expressed by:

$$W = \sum_{k=1}^K I(C_K)$$

where the individual intra-cluster inertia $I(C_k)$ is determined as:

$$I(C_K) = \sum_{i \in C_K} w_i d_i^2(x_i, g_k)$$

where d_i is the distance between observation x_i and the centre of the cluster g_k , while w_i is the weight assigned to the observation (which specifically may be $1/n$ for n observations). It measures the heterogeneity within clusters—the lower the inertia and thus the heterogeneity, the more coherent the clusters.

Between-cluster inertia B is a measurement of the separation between clusters and is expressed as the sum of the weighted squared distances d_k between the centres of g_k clusters and the centre g of all observations considered together. Hence, the inter-cluster inertia is given as:

$$B = \sum_{k=1}^K \mu_k d_k^2(g_k, g)$$

where μ_k is the sum of the weights assigned to the observations inside the given cluster k :

$$\mu_k = \sum_{i \in C_k} w_i.$$

Total inertia T is the sum of the weighted squared distances d_g between individual observations x_i and the centre g of all observations taken together:

$$T = \sum_{i=1}^n w_i d_g^2(x_i, g).$$

It does not depend on the division into clusters and can be also expressed as the sum of intra-cluster inertia W and inter-cluster inertia B :

$$T = W + B.$$

Clustering quality measure: Dunn index

The Dunn index, introduced by Dunn (1974), is based on extreme values only. It examines the quality of clustering, in particular, whether the number of clusters k was set properly. It compares two parameters of K clusters:

- In counter, the minimum separation of clusters is calculated as the minimum d_{\min} (for all clusters) of the shortest distance $d_{kk'}$ between two clusters (separation between the closest points M of two clusters k and k'):

$$d_{\min} = \min_{k \neq k'} d_{kk'} \text{ where } d_{kk'} = \min_{i \in I_k, j \in I_{k'}} \left\| M_i^{\{k\}} - M_j^{\{k'\}} \right\|.$$

- In numerator, the diameter of the cluster is calculated as the maximum (for all clusters) of the largest distance D_k between points M within given cluster k :

$$d_{\max} = \max_{1 \leq k \leq K} D_k \text{ where } D_k = \max_{i, j \in I_k, i \neq j} \left\| M_i^{\{k\}} - M_j^{\{k\}} \right\|.$$

Thus, the Dunn index is expressed as $\text{Dunn} = d_{\min}/d_{\max}$. In the case of good partitioning, in which clusters are small (small diameter) and well separated (large distance between clusters), the Dunn index will be high.

Much more on measures of clustering quality can be found in Vignettes to R package clusterCrit:¹⁶ or in Tibshirani et al. (2000).

k-fold cross-validation

Currently, there exist two approaches to cross-validation (CV): (i) dividing data into two groups—training and testing or (ii) dividing data into three groups—training, fine-tuning and testing. When dividing data into two groups, the sample is divided into k -folds (parts, subsamples); $k - 1$ folds are used in training of the model and 1 part is used in testing of the model. The process is conducted recursively k times, so each of the k -folds plays a role of testing part of the sample. When dividing data into three groups, part of the data is kept aside for out-of-sample predictions, and these data are not used for model fitting and fine-tuning. The rest of the data is used as it is with the approach of dividing data into two groups. In the case of fivefold cross-validation, the data used for model fitting and fine-tuning are divided into five equal parts (each part consisting of 20% of the data), and in each of five iterations, the model is fitted on 80% of the data and tested on 20% of the data.

Supervised machine learning—typology of methods

Supervised learning tools supplement typical models of regression (with continuous dependent variable) and classification (with few levels of dependent variable).

¹⁶ <https://cran.r-project.org/web/packages/clusterCrit/vignettes/clusterCrit.pdf>.

According to Kuhn and Johnson (2016), regression modelling with the exception of linear regression models, such as ordinary least squares (OLS), includes non-linear regressions (based on neural networks, SVM, KNN) and regression trees and rule-based models (such as regression trees, random forest, cubist, boosting). Similar divisions can be found for classification methods, which include linear models (logistic regression—logit, probit or linear discriminant analysis), non-linear regressions (such as neural networks, support vector machines, K -nearest neighbours, naïve Bayes) and regression trees and rule-based models (regression trees, random forest, boosting).

Naïve Bayes classifier

The Naïve Bayes classifier is a statistical model, based on Bayesian probability formula. In the building phase of the binary choice model (e.g. class yes/no, more levels also possible), it derives the probabilities of each class, and the probabilities that features X (which include features e.g. $\times 1, \times 2, \dots$) interact with each class; in fact, it determines the probabilities of features appearing in a given class $P(\text{yes})$, $P(\text{no})$ and the structure of features in a given class $P(\times 1|\text{yes})$, $P(\times 2|\text{yes})$, $P(\times 1|\text{no})$, $P(\times 2|\text{no})$. It assumes that features X (e.g. $\times 1, \times 2, \dots$) are independent of each other. In the prediction of new data, it calculates the Bayesian posterior probabilities by using (in the case of two features) $P(c|X) = \frac{P(X|c) \cdot P(c)}{P(X)} = \frac{P(x1|c) \cdot P(x2|c) \cdot P(c)}{P(x1) \cdot P(x2)}$. The highest score classifies observations into a given class.

K-nearest neighbours classifier

In k -nearest neighbours classifier, the observations are classified based on the class of their k -nearest neighbours (knn). Firstly, it determines which k training observations are the nearest neighbours for test observation, by calculating multi-dimensional distance; secondly, it checks the classes of knn training observations; and thirdly, with majority (or distance-weighted) voting it chooses the most frequent class. It requires calculating distances between test and all training observations. Good overview of the method can be found in Cunningham and Delany (2007).

Random forest classifier

The random forest classifier is an ensemble method (using the concept of the wisdom of the crowd), based on decision trees, which divide selected features into groups to profile a given class. Random forest is a collection of independent trees; they differ from each other in that observations are selected in bagging (sampling with replacement, bootstrap) and m features are drawn randomly (a few variables from a bigger set). Majority voting aggregates the results from trees—it takes each

class in turn, checks the output (class) at the bottom of each tree and calculates the average of the features' values, which are on the path to a given class. Quality checking follows the out-of-bag (*oob*) scheme: when bagging, one divides observations, keeping approximately 2/3 for training and around 1/3 for testing the model. The number of features m should be small enough to keep trees uncorrelated and large enough to keep trees strong; it is optimised by controlling the *oob* error rate. The *oob* error rate is the frequency that test data did not meet their true value. Variable importance is tested by permuting the values of the m th variable among *oob* observations and checking the prediction of trees; the difference between ratios of correct class prediction in non-permuted and permuted tests is known as variable importance. A technical overview is available in vignettes of random forest software by Breiman and Cutler (see link¹⁷).

Support vector machines

With support vector machines, the observations are separated into classes with lines (in 2D) or hyperplanes (in 3D and more). Support vectors are the points in all classes which are closest to the line/hyperplane; the distance (called the margin) between those points and the line/hyperplane should be maximised. In the event that the points are not linearly separable, they are transformed to make this possible (see introduction in link¹⁸).

Artificial neural networks

An artificial neural network (ANN) is a classifier method which operates on binary input and output. Each type of information (variable, image cell, etc.) is analysed by an individual perceptron. Numerical data are binarised depending on their threshold (e.g. $x > a$), whereas quantitative data are binarised depending on a given feature (yes/no). Dummy outputs of perceptrons are weighted and aggregated in an additive function; this result is again contrasted with the threshold to give a binary answer. The answer given by the ANN is compared with the true state. In case of error (expressed as loss function), the ANN learns by altering the weights to match the true answer (see introduction in link¹⁹).

Maximum entropy classifier

The maximum entropy classifier is a probabilistic model, without assumptions on independence of features (oppositely, it assumes correlations), using the concept of entropy. It is based on the Bayesian probability formula as with the naïve Bayes

¹⁷ https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm.

¹⁸ <https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989>.

¹⁹ <https://www.bmc.com/blogs/neural-network-introduction/>.

classifier, but instead of assuming empirical probabilities, it starts with uniform weights and optimises them (see introduction in link²⁰).

Autoencoder-based residual network

Autoencoder-based residual networks are unsupervised learning models that (similarly to PCA) extract features from wider datasets. The encoder network transforms the input image into the model (with latent variables), while decoder network reconstructs the image. The residual network adds a layer which gradually learns from residuals (see introduction in link²¹).

Gradient boosting

The gradient boosting algorithm, for which the most popular is XGBoots, is, like random forest, based on decision trees. However, instead of growing all trees simultaneously (as with random forest), it works iteratively. Next, the model corrects the mistakes of the previous model—misclassifications are analysed, and wrongly predicted observations are given higher weights in analysis to be more intensively addressed in the next round. The final model is an additive decision tree, which includes all good models (see introduction in link²²).

Cubist

The Cubist algorithm, introduced by Quinlan (1992), is based on a tree. For each path (to the terminal leaf), it creates a rule with a regression multivariate model. Covariates which fulfil the criteria of the tree are used in those models. These models are used for predictions and strengthened (averaged) with neighbouring models (located above in the tree).

References of Appendix 1

- Assunção, R. M., Neves, M. C., Câmara, G., & da Costa Freitas, C. (2006). Efficient regionalisation techniques for socio-economic geographical units using minimum spanning trees. *International Journal of Geographical Information Science*, 20(7), 797–811.
- Bock, H. H. (2007). Clustering methods: a history of *k*-means algorithms. *Selected contributions in data analysis and classification*, 161–172 in [eds.] Brito P., Cucumel P., de Carvalho F. (2007), *Selected contributions in data analysis and classification*. Springer Science & Business Media.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press

²⁰ <https://blog.datumbox.com/machine-learning-tutorial-the-max-entropy-text-classifier/>.

²¹ http://essay.utwente.nl/83138/1/Bhaswara_MA_EEMCS.pdf, <https://bjlkeng.github.io/posts/residual-networks/>.

²² <https://www.datacamp.com/community/tutorials/xgboost-in-python>

- Cunningham, P., Delany, S.J. (2007), *k*-Nearest Neighbour Classifiers, Technical Report UCD-CSI-2007-4,
- Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1), 95–104.
- Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei (1996). *A density-based algorithm for discovering clusters in large spatial databases with noise*. In [eds.] Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231.
- Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, 857–871.
- Guo, D. (2008). Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP). *International Journal of Geographical Information Science*, 22(7), 801–823.
- Hamming, R. W. (April 1950). "Error detecting and error correcting codes" (PDF). *The Bell System Technical Journal*. 29 (2): 147–160. <https://doi.org/10.1002/j.1538-7305.1950.tb00463.x>. ISSN 0005–8580.
- Kaufman, L., Rousseeuw, P.J. (1987), *Clustering by means of Medoids*, in *Statistical Data Analysis Based on the Norm and Related Methods*, edited by Y. Dodge, North-Holland, s. 405–416.
- Kaufman, L., Rousseeuw, P.J. (1990), Clustering Large Applications (Program CLARA) in [eds.] Kaufman, L., Rousseeuw, P.J., *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley Series in Probability and Statistics
- Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26). New York: Springer.
- Левенштейн В. И. (1965). Двоичные коды с исправлением выпадений, вставок и замещений символов [Binary codes capable of correcting deletions, insertions, and reversals]. Доклады Академии Наук СССР (in Russian). 163 (4): 845–848. Appeared in English as: Levenshtein, Vladimir I. (February 1966). "Binary codes capable of correcting deletions, insertions, and reversals". *Soviet Physics Doklady*. 10 (8): 707–710. Bibcode:1966SPhD...10.0.707L.
- LLOYD, S.P. (1957): Least squares quantization in PCM. Bell Telephone Labs Memorandum, Murray Hill, NJ. Reprinted in: *IEEE Trans. Information Theory* IT-28 (1982), vol. 2, 129–137.
- MacQUEEN, J. (1967): Some methods for classification and analysis of multivariate observations. In: L.M. LeCam, J. Neyman (eds.): *Proc. 5th Berkeley Symp. Math. Statist. Probab.* 1965/66. Univ. of California Press, Berkeley, vol. I, 281–297
- Mahalanobis, P. C. (1936). On the generalized distance in statistics. *National Institute of Science of India*.
- Sokal, R. R. and F. J. Rohlf. 1962. The comparison of dendrograms by objective methods. *Taxon*, 11:33–40
- STEINHAUS, H. (1956): Sur la division des corps materiels en parties. *Bulletin de l'Academie Polonaise des Sciences, Classe III*, vol. IV, no. 12, 801–804.
- Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society:*

Series B (Statistical Methodology), 63(2), 411–423. <https://statweb.stanford.edu/~gwalther/gap>,
 Quinlan. Learning with continuous classes. Proceedings of the 5th Australian Joint Conference On Artificial Intelligence (1992) pp. 343–348

Appendix 2: Implementations in R

The majority of methods discussed in this paper have their software implementations in R. None of the other existing software offers comprehensive solutions for either machine learning or spatial data processing and computations. TaskViews of R software (at www.r-project.org) give comprehensive and up-to-date overviews of packages for clustering (Cluster Analysis & Finite Mixture Models²³) and machine learning (Machine Learning & Statistical Learning²⁴). Its applications for environmental data can be found in TaskViews on Analysis of Ecological and Environmental Data.²⁵ Those for spatial analysis can be found in TaskViews on Analysis of Spatial Data²⁶ and Handling and Analysing Spatio-Temporal Data.²⁷

Among a great variety of packages and functions, few examples are particularly notable:

Unsupervised learning and clustering

- **stats::**, **ClusterR::**, **cluster::**, **clustering::**, **fpc::**, **factoextra::**, **FactoMineR::** offer standard clustering a-spatial methods (*k*-means, PAM, CLARA, knn) and their testing, different metrics of distance,
- **NbClust::**, **optCluster::** offers many tests for clustering quality and selection of number of clusters,
- **h2o::** offers a-spatial fuzzy *k*-means algorithms,
- **ClustGeo::** and **rgeoda::** offer simultaneous clustering of values and locations (spatially constrained clustering),
- **spatialClust::** offers spatial clustering using fuzzy geographically weighted clustering,
- **SpODT::** offers a spatial oblique decision tree based on the classification and regression tree,

²³ <https://cran.r-project.org/web/views/Cluster.html>.

²⁴ <https://cran.r-project.org/web/views/MachineLearning.html>.

²⁵ <https://cran.r-project.org/web/views/Environmetrics.html>.

²⁶ <https://cran.r-project.org/web/views/Spatial.html>.

²⁷ <https://cran.r-project.org/web/views/SpatioTemporal.html>.

- **dbscan::** offers density-based clustering with DBSCAN,
- **rgeoda::** offers SKATER and REDCAP algorithms,²⁸
- **automap::** offers many versions of kriging,
- **StatMatch::** offers Gower distance.

Non-covered topics are also widely available in R: in **geoGAM::** (geoadditive models for spatial prediction), **mgcv::** (generalised additive model using splines), **MapGam::** (mapping smoothed effect estimates from individual-level data), **SpatialEpi::** (cluster detection and disease mapping for spatial epidemiology), **rsatscan::** (interface to SaTScan software), **graphscan::** (scan statistics in 2D and 3D), **rflexscan::** (flexible spatial scan statistic).

Supervised learning

- **ranger::**, **randomForest::** offer random forest modelling,
- **xgboost::**, **gbm::**, **plyr::** offer gradient boosting,
- **carret::** offers many classification and regression machine algorithms and fine-tuning of its parameters,
- **nnet::** offers neural networks algorithms, in particular a model-averaged neural network,
- **earth::** offers multivariate adaptive regression splines, also bagged (MARS),
- **cubist::**, **Cubist::** offer Cubist algorithms,
- **kernlab::** offers support vector regression, also with radial basis function kernel regression trees,
- **e1071::** offers naïve Bayes model,
- **party::** offers partitioning and conditional inference tree—regression trees for all types of data.

Appendix 3: Data used in spatial machine learning

A popular source of data is MODIS (*moderate resolution imaging spectroradiometer*), which contains data from NASA (<https://modis.gsfc.nasa.gov/>) for the whole of the Earth's surface for every 1–2 days in 36 spectral bands. The data are divided into four categories:

- MODIS level 1 data (with geolocation, cloud mask and atmosphere products) <http://ladsweb.nascom.nasa.gov/>.
- MODIS land products (with land surface temperature, products, vegetation indices, etc.) <https://lpdaac.usgs.gov/>.
- MODIS cryosphere products (with snow cover and sea ice surface temperature) <http://nsidc.org/daac/modis/index.html>.

²⁸ See **rgeoda::** vignettes <https://rgeoda.github.io/rgeoda-book/> and tutorials. https://geodacenter.github.io/tutorials/spatial_cluster/skater.html.

- MODIS ocean colour and sea surface temperature products (also on carbon, fluorescence line, etc.) <http://oceancolor.gsfc.nasa.gov/>.

The Planetary Habitability Laboratory also offers satellite images and climate data <http://phl.upr.edu/data>. There are also many software packages which are helpful for gathering proper data (such as SAGA, System for Automated Geoscientific Analyses²⁹).

Using three channels (red, green, blue) of aerial image, one can construct so-called spectral predictors, e.g. visible vegetation index (VVI, Planetary Habitability Laboratory), triangular greenness index (TGI), normalised difference vegetation index (NDVI), normalised green–red difference index (NGRDI), green leaf index (GLI), etc. R function `rgb_indices()` from `uavRst::` package³⁰ offers 17 spectral indices. `IndexDataBase`³¹ offers comprehensive specification of formula for spectral indices based on data from 68 different sensors. One can also run PCA on visible spectra and spatial predictors—the first few principal components are used instead of these variables to avoid duplication of the information.

Another popular source of data is LIDAR (light detection and ranging). They are available from many sources³² as OpenTopology, USGS Earth Explorer, United States Inter-agency Elevation Inventory, NOAA Digital Coast, National Ecological Observatory Network (NEON), LIDAR Data Online, etc. It allows variables to be obtained as digital elevation model (DEM), slope and aspect (on the basis of DEM) in, e.g. radians, geolocation variables (such as longitude and latitude), etc.

Additionally, interesting information is Night Light Data, available from World Bank,³³ SOS NOAA (Science on a Sphere, National Oceanic and Atmospheric Administration³⁴) and from NASA³⁵ or Google Earth (earth.google.com).

One can also gather much open data from Open Governmental repositories, as data.gov (USA), data.gov.uk (UK), govdata.de (Germany), <https://www.europeandataportal.eu/en> (European Union), etc.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

²⁹ <http://www.saga-gis.org/en/index.html>.

³⁰ http://finzi.psych.upenn.edu/library/uavRst/html/rgb_indices.html.

³¹ <https://www.indexdatabase.de/db/i-single.php?id=375>.

³² <https://gisgeography.com/top-6-free-lidar-data-sources/>.

³³ <https://datacatalog.worldbank.org/dataset/worldwide-night-time-lights>.

³⁴ <https://sos.noaa.gov/datasets/nighttime-lights/>.

³⁵ <https://www.nasa.gov/feature/goddard/2017/new-night-lights-maps-open-up-possible-real-time-applications>.

References

- Aharon-Gutman M, Burg D (2019) How 3D visualisation can help us understand spatial inequality: on social distance and crime. *Environ Plan B Urban Anal City Sci* 48(4):793–809
- Aharon-Gutman M, Schaap M, Lederman I (2018) Social topography: studying spatial inequality using a 3D regional model. *J Rural Stud* 62:40–52
- Ahn S, Ryu DW, Lee S (2020) A machine learning-based approach for spatial estimation using the spatial features of coordinate information. *ISPRS Int J Geo Inf* 9(10):587
- Alexandrov T, Kobarg JH (2011) Efficient spatial segmentation of large imaging mass spectrometry datasets with spatially aware clustering. *Bioinformatics* 27(13):i230–i238
- Amato F, Guignard F, Robert S et al (2020) A novel framework for spatio-temporal prediction of environmental data using deep learning. *Sci Rep* 10:22243. <https://doi.org/10.1038/s41598-020-79148-7>
- Amiri M, Pourghasemi HR, Ghanbarian GA, Afzali SF (2019) Assessment of the importance of gully erosion effective factors using Boruta algorithm and its spatial modeling and mapping using three machine learning algorithms. *Geoderma* 340:55–69
- Ankerst M, Breunig MM, Kriegel HP, Sander J (1999) OPTICS: ordering points to identify the clustering structure. *ACM SIGMOD Rec* 28(2):49–60
- Appelhans T, Mwangomo E, Hardy DR, Hemp A, Nauss T (2015) Evaluating machine learning approaches for the interpolation of monthly air temperature at Mt. Kilimanjaro, Tanzania. *Spat Stat* 14:91–113
- Arbia G, Ghiringhelli C, Mira A (2019) Estimation of spatial econometric linear models with large datasets: How big can spatial Big Data be? *Reg Sci Urban Econ* 76:67–73
- Assunção RM, Neves MC, Câmara G, da Costa Freitas C (2006) Efficient regionalisation techniques for socio-economic geographical units using minimum spanning trees. *Int J Geogr Inf Sci* 20(7):797–811
- Aydin O, Janikas MV, Assunção R, Lee TH (2018, November) SKATER-CON: unsupervised regionalisation via stochastic tree partitioning within a consensus framework using random spanning trees. In: *Proceedings of the 2nd ACM SIGSPATIAL international workshop on AI for geographic knowledge discovery*, pp 33–42
- Bajocco S, Dragoz E, Gitas I, Smiraglia D, Salvati L, Ricotta C (2015) Mapping forest fuels through vegetation phenology: The role of coarse-resolution satellite time-series. *PLoS ONE* 10(3):e0119811
- Behrens T, Rossel RAV (2020) On the interpretability of predictors in spatial data science: the information horizon. *Sci Rep* 10(1):1–10
- Behrens T, Schmidt K, Viscarra Rossel RA, Gries P, Scholten T, MacMillan RA (2018) Spatial modelling with Euclidean distance fields and machine learning. *Eur J Soil Sci* 69(5):757–770
- Besag J, Newell J (1991) The detection of clusters in rare diseases. *J R Stat Soc A Stat Soc* 154(1):143–155
- Birant D, Kut A (2007) ST-DBSCAN: an algorithm for clustering spatial–temporal data. *Data Knowl Eng* 60(1):208–221
- Brimicombe AJ (2007) A dual approach to cluster discovery in point event data sets. *Comput Environ Urban Syst* 31(1):4–18
- Cai L, Li Y, Chen M, Zou Z (2020) Tropical cyclone risk assessment for China at the provincial level based on clustering analysis. *Geomat Nat Hazards Risk* 11(1):869–886
- Campello RJ, Moulavi D, Sander J (2013) Density-based clustering based on hierarchical density estimates. In: *Pacific-Asia conference on knowledge discovery and data mining*, pp 160–172. Springer, Berlin
- Cecchini S, Savio G, Tromben V (2021) Mapping poverty rates in Chile with night lights and fractional multinomial models. *Reg Sci Policy Pract*. <https://doi.org/10.1111/rsp3.12415>
- Chasco C, Le Gallo J, López FA (2018) A scan test for spatial groupwise heteroscedasticity in cross-sectional models with an application on houses prices in Madrid. *Reg Sci Urban Econ* 68:226–238
- Chen W, Pourghasemi HR, Kornejady A, Zhang N (2017) Landslide spatial modeling: introducing new ensembles of ANN, MaxEnt, and SVM machine learning techniques. *Geoderma* 305:314–327
- Chernick MR, LaBudde RA (2014) *An introduction to bootstrap methods with applications to R*. Wiley
- Chi SH, Grigsby-Toussaint DS, Bradford N, Choi J (2013) Can geographically weighted regression improve our contextual understanding of obesity in the US? Findings from the USDA Food Atlas. *Appl Geogr* 44:134–142

- Cracknell MJ, Reading AM (2014) Geological mapping using remote sensing data: a comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information. *Comput Geosci* 63:22–33
- Czerniawski T, Sankaran B, Nahangi M, Haas C, Leite F (2018) 6D DBSCAN-based segmentation of building point clouds for planar object classification. *Autom Constr* 88:44–58
- Debnath M, Tripathi PK, Elmasri R (2015, September) K-DBSCAN: identifying spatial clusters with differing density levels. In: 2015 International workshop on data mining with industrial applications (DMIA), pp 51–60. IEEE
- Distefano V, Mameli V, Poli I (2020) Identifying spatial patterns with the Bootstrap ClustGeo technique. *Spat Stat* 38:100441
- Du P, Bai X, Tan K, Xue Z, Samat A, Xia J, Liu W (2020) Advances of four machine learning methods for spatial data handling: a review. *J Geovis Spat Anal* 4:1–25
- Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of 2nd international conferences on knowledge discovery data mining
- Estivill-Castro V, Lee I (2002) Argument free clustering for large spatial point-data sets via boundary extraction from Delaunay Diagram. *Comput Environ Urban Syst* 26(4):315–334
- Farrell A, Wang G, Rush SA, Martin JA, Belant JL, Butler AB, Godwin D (2019) Machine learning of large-scale spatial distributions of wild turkeys with high-dimensional environmental data. *Ecol Evol* 9(10):5938–5949
- Fatima M, Pasha M (2017) Survey of machine learning algorithms for disease diagnostic. *J Intell Learn Syst Appl* 9(01):1
- Fotheringham AS, Crespo R, Yao J (2015) Geographical and temporal weighted regression (GTWR). *Geogr Anal* 47(4):431–452
- Fotheringham AS, Yang W, Kang W (2017) Multiscale geographically weighted regression (MGWR). *Ann Am Assoc Geogr* 107(6):1247–1265
- Galán SF (2019) Comparative evaluation of region query strategies for DBSCAN clustering. *Inf Sci* 502:76–90
- Gerassis S, Boente C, Albuquerque MTD, Ribeiro MM, Abad A, Taboada J (2020) Mapping occupational health risk factors in the primary sector—a novel supervised machine learning and Area-to-Point Poisson kriging approach. *Spat Stat* 42:100434
- Goel R, Garcia LM, Goodman A, Johnson R, Aldred R, Murugesan M, Woodcock J (2018) Estimating city-level travel patterns using street imagery: A case study of using Google Street View in Britain. *PLoS ONE* 13(5):e0196521
- Goetz JN, Brenning A, Petschko H, Leopold P (2015) Evaluating machine learning and statistical prediction techniques for landslide susceptibility modeling. *Comput Geosci* 81:1–11
- Goulard M, Laurent T, Thomas-Agnan C (2017) About predictions in spatial autoregressive models: optimal and almost optimal strategies. *Spat Econ Anal* 12(2–3):304–325
- Griffith DA, Liau YT (2020) Imputed spatial data: cautions arising from response and covariate imputation measurement error. *Spat Stat* 42:100419
- Guo D (2008) Regionalisation with dynamically constrained agglomerative clustering and partitioning (REDCAP). *Int J Geogr Inf Sci* 22(7):801–823
- Hagenauer J, Omrani H, Helbich M (2019) Assessing the performance of 38 machine learning models: the case of land consumption rates in Bavaria, Germany. *Int J Geogr Inf Sci* 33(7):1399–1419
- Hall P, Horowitz JL, Jing BY (1995) On blocking rules for the bootstrap with dependent data. *Biometrika* 82(3):561–574
- Heil BJ, Hoffman MM, Markowitz F, Lee SI, Greene CS, Hicks SC (2021) Reproducibility standards for machine learning in the life sciences. *Nat Methods* 18(10):1132–1135
- Helbich M, Brunauer W, Hagenauer J, Leitner M (2013) Data-driven regionalisation of housing markets. *Ann Assoc Am Geogr* 103(4):871–889
- Hengl T, Leenaars JG, Shepherd KD, Walsh MG, Heuvelink GB, Mamo T, Kwabena NA (2017) Soil nutrient maps of Sub-Saharan Africa: assessment of soil nutrient content at 250 m spatial resolution using machine learning. *Nutr Cycl Agroecosyst* 109(1):77–102
- Hengl T, Nussbaum M, Wright MN, Heuvelink GB, Gräler B (2018) Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. *PeerJ* 6:e5518
- Jégou L, Bahoken F, Chickhaoui E, Duperron E, Maisonobe M (2019, August) Spatial aggregation methods: an interactive visualisation tool to compare and explore automatically generated urban

- perimeters. In: 59th ERSa congress “cities, regions and digital transformations: opportunities, risks and challenges”
- Joncour I, Duchêne G, Moraux E, Motte F (2018) Multiplicity and clustering in Taurus star forming region-II. From ultra-wide pairs to dense NESTs. *Astron Astrophys* 620:A27
- Joost S, De Ridder D, Marques-Vidal P, Bacchilega B, Theler JM, Gaspoz JM, Guessous I (2019) Overlapping spatial clusters of sugar-sweetened beverage intake and body mass index in Geneva state, Switzerland. *Nutr Diabetes* 9(1):1–10
- Joshi D, Samal A, Soh LK (2013) Spatio-temporal polygonal clustering with space and time as first-class citizens. *GeoInformatica* 17(2):387–412
- Khan K, Rehman SU, Aziz K, Fong S, Sarasvady S (2014, February) DBSCAN: past, present and future. In: The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014), pp 232–238. IEEE
- Kim J, Cho J (2019) Delaunay triangulation-based spatial clustering technique for enhanced adjacent boundary detection and segmentation of LiDAR 3D point clouds. *Sensors* 19(18):3926
- Kober J, Bagnell JA, Peters J (2013) Reinforcement learning in robotics: a survey. *Int J Robot Res* 32(11):1238–1274
- Kohoutová L, Heo J, Cha S, Lee S, Moon T, Wager TD, Woo CW (2020) Toward a unified framework for interpreting machine-learning models in neuroimaging. *Nat Protoc* 15(4):1399–1435
- Kopczewska K (ed) (2020) Applied spatial statistics and econometrics: data analysis in R. Routledge
- Kopczewska K (2021) Spatial bootstrapped microeconometrics: forecasting for out-of-sample geo-locations in big data, forthcoming
- Kopczewska K, Cwiakowski P (2021) Spatio-temporal stability of housing submarkets. Tracking spatial location of clusters of geographically weighted regression estimates of price determinants. *Land Use Policy* 103:105292
- Koziatek O, Dragičević S (2019) A local and regional spatial index for measuring three-dimensional urban compactness growth. *Environ Plan B Urban Anal City Sci* 46(1):143–164
- Kraamwinkel C, Fabris-Rotelli I, Stein A (2018) Bootstrap testing for first-order stationarity on irregular windows in spatial point patterns. *Spat Stat* 28:194–215
- Kulldorff M (1997) A spatial scan statistic. *Commun Stat Theory Methods* 26(6):1481–1496
- Lee J, Gangnon RE, Zhu J (2017) Cluster detection of spatial regression coefficients. *Stat Med* 36(7):1118–1133
- Li L (2019) Geographically weighted machine learning and downscaling for high-resolution spatiotemporal estimations of wind speed. *Remote Sens* 11:1378
- Li J, Heap AD, Potter A, Daniell JJ (2011) Application of machine learning methods to spatial interpolation of environmental variables. *Environ Model Softw* 26(12):1647–1659
- Li X, Zhang C, Li W (2015) Does the visibility of greenery increase perceived safety in urban areas? Evidence from the place pulse 1.0 dataset. *ISPRS Int J Geo-Inf* 4(3):1166–1183
- Libbrecht M, Noble W (2015) Machine learning applications in genetics and genomics. *Nat Rev Genet* 16:321–332. <https://doi.org/10.1038/nrg3920>
- Liu X (2020) Incorporating spatial autocorrelation in machine learning. Master’s thesis, University of Twente
- Liu RY, Singh K (1992) Moving blocks jackknife and bootstrap capture weak dependence. In: LePage R, Billard L (eds) *Exploring the Limits of Bootstrap*. John Wiley & Sons Inc, New York, pp 225–248
- Liu D, Nosovskiy GV, Sourina O (2008) Effective clustering and boundary detection algorithm based on Delaunay triangulation. *Pattern Recogn Lett* 29(9):1261–1273
- Liu D, Wang X, Cai Y, Liu Z, Liu ZJ (2020a) A novel framework of real-time regional collision risk prediction based on the RNN approach. *J Mar Sci Eng* 8(3):224
- Liu M, Hu S, Ge Y, Heuvelink GB, Ren Z, Huang X (2020b) Using multiple linear regression and random forests to identify spatial poverty determinants in rural China. *Spat Stat* 42:100461
- Lovelace R, Nowosad J, Muenchow J (2019) *Geocomputation with R*. Chapman & Hall/CRC The R Series
- Lu Y, Thill JC (2003) Assessing the cluster correspondence between paired point locations. *Geogr Anal* 35(4):290–309
- Lu W, Han J, Ooi BC (1993, June) Discovery of general knowledge in large spatial databases. In: *Proceedings of Far East workshop on geographic information systems*, Singapore, pp 275–289
- MacQueen J (1967, June) Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol 1, no 14, pp 281–297

- Marco M, Gracia E, Martín-Fernández M, López-Quílez A (2017) Validation of a Google Street View-based neighborhood disorder observational scale. *J Urban Health* 94(2):190–198
- Masolele RN, De Sy V, Herold M, Marcos D, Verbesselt J, Gieseke F, Mullissa A, Martius C (2021) Spatial and temporal deep learning methods for deriving land-use following deforestation: a pan-tropical case study using Landsat time series. *Remote Sens Environ* 264:112600
- Matschinske J, Alcaraz N, Benis A, Golebiewski M, Grimm DG, Heumos L, Kacprowski T, Lazareva O, List M, Louadi Z, Pauling JK, Pfeifer N, Röttger R, Schwämmle V, Sturm G, Traverso A, Van Steen K, Vaz de Freitas M, Silva GCV, Wee L, Wenke NK, Zanin M, Zolotareva O, Baumbach J, Blumenthal DB (2021) The AIME registry for artificial intelligence in biomedical research. *Nat Methods* 18:1128–1131. <https://doi.org/10.1038/s41592-021-01241-0>
- Meyer H, Reudenbach C, Hengl T, Katurji M, Nauss T (2018) Improving performance of spatio-temporal machine learning models using forward feature selection and target-oriented validation. *Environ Model Softw* 101(March):1–9
- Meyer H, Reudenbach C, Wöllauer S, Nauss T (2019) Importance of spatial predictor variable selection in machine learning applications—moving from data reproduction to spatial prediction. *Ecol Model* 411:108815
- Müller S, Wilhelm P, Haase K (2013) Spatial dependencies and spatial drift in public transport seasonal ticket revenue data. *J Retail Consum Serv* 20(3):334–348
- Mustakim IRNG, Novita R, Kharisma OB., Vebrianto R, Sanjaya S, Andriani T, Sari WP, Novita Y, Rahim R (2019) DBSCAN algorithm: twitter text clustering of trend topic pilkada pekanbaru. In: *Journal of physics: conference series*, vol 1363, no 1, p 012001. IOP Publishing
- Editorial N (2021a) Keeping checks on machine learning. *Nat Methods* 18:1119. <https://doi.org/10.1038/s41592-021-01300-6>
- Editorial N (2021b) Moving towards reproducible machine learning. *Nat Comput Sci*. <https://doi.org/10.1038/s43588-021-00152-6>
- Nicolis O, Plaza F, Salas R (2020) Prediction of intensity and location of seismic events using deep learning. *Spat Stat* 42:100442
- Nikparvar B, Thill JC (2021) Machine learning of spatial data. *ISPRS Int J Geo Inf* 10(9):600
- Openshaw S, Charlton M, Wymer C, Craft A (1987) A mark 1 geographical analysis machine for the automated analysis of point data sets. *Int J Geogr Inf Syst* 1(4):335–358
- Park B, Bae JK (2015) Using machine learning algorithms for housing price prediction: the case of Fairfax County, Virginia housing data. *Expert Syst Appl* 42(6):2928–2934
- Pavlis M, Dolega L, Singleton A (2018) A modified DBSCAN clustering method to estimate retail center extent. *Geogr Anal* 50(2):141–161
- Peña MA, Brenning A (2015) Assessing fruit-tree crop classification from Landsat-8 time series for the Maipo Valley, Chile. *Remote Sens Environ* 171:234–244
- Pourghasemi HR, Sadhasivam N, Kariminejad N, Collins AL (2020) Gully erosion spatial modelling: role of machine learning algorithms in selection of the best controlling factors and modelling process. *Geosci Front* 11(6):2207–2219
- Quiñones S, Goyal A, Ahmed ZU (2021) Geographically weighted machine learning model for untangling spatial heterogeneity of type 2 diabetes mellitus (T2D) prevalence in the USA. *Sci Rep* 11:6955. <https://doi.org/10.1038/s41598-021-85381-5>
- Reis J, Costa MU (2015) Incremental DBSCAN for green computing. Working Paper, VisionSpace Technologies
- Rodríguez-Pérez JR, Ordóñez C, Roca-Pardiñas J, Vecín-Arias D, Castedo-Dorado F (2020) Evaluating lightning-caused fire occurrence using spatial generalized additive models: a case study in Central Spain. *Risk Anal* 40(7):1418–1437
- Rolf E, Proctor J, Carleton T et al (2021) A generalizable and accessible approach to machine learning with global satellite imagery. *Nat Commun* 12:4392. <https://doi.org/10.1038/s41467-021-24638-z>
- Ruiz C, Spiliopoulou M, Menasalvas E (2007, May) C-dbscan: Density-based clustering with constraints. In: *International workshop on rough sets, fuzzy sets, data mining, and granular-soft computing*, pp 216–223. Springer, Berlin
- Russ G, Brenning A (2010) Spatial variable importance assessment for yield prediction in precision agriculture. In: *International symposium on intelligent data analysis*. Springer, Berlin, pp 184–195
- Sawada Y (2019) Machine learning accelerates parameter optimisation and uncertainty assessment of a land surface model. <http://arxiv.org/abs/1909.04196> [stat.AP]

- Schratz P, Muenchow J, Iturritxa E, Richter J, Brenning A (2019) Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. *Ecol Model* 406:109–120
- Sergeev AP, Buevich AG, Baglaeva EM, Shichkin AV (2019) Combining spatial autocorrelation with machine learning increases prediction accuracy of soil heavy metals. *CATENA* 174:425–435
- Soltani A, Pettit CJ, Heydari M, Aghaei F (2021) Housing price variations using spatio-temporal data mining techniques. *J Hous Built Environ* 36:1–29
- VoPham T, Hart JE, Laden F, Chiang YY (2018) Emerging trends in geospatial artificial intelligence (geoAI): potential applications for environmental epidemiology. *Environ Health* 17(1):1–6
- Walsh I, Fishman D, Garcia-Gasulla D, Titma T, Pollastri G, ELIXIR Machine Learning Focus Group, Harrow J, Psomopoulos FE, Tosatto SCE (2021) DOME: recommendations for supervised machine learning validation in biology. *Nat Methods* 18:1122–1127. <https://doi.org/10.1038/s41592-021-01205-4>
- Wang C, Ji M, Wang J, Wen W, Li T, Sun Y (2019a) An improved DBSCAN method for LiDAR data segmentation with automatic Eps estimation. *Sensors* 19(1):172
- Wang K, Yu X, Xiong Q, Zhu Q, Lu W, Huang Y, Zhao L (2019b) Learning to improve WLAN indoor positioning accuracy based on DBSCAN-KRF algorithm from RSS fingerprint data. *IEEE Access* 7:72308–72315
- Wang W, Yang J, Muntz R (1997) STING: a statistical information grid approach to spatial data mining. In: *VLDB*, vol 97, pp 186–195
- Weatherill G, Burton PW (2009) Delineation of shallow seismic source zones using K-means cluster analysis, with application to the Aegean region. *Geophys J Int* 176(2):565–588
- Wratten L, Wilm A, Göke J (2021) Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. *Nat Methods* 18:1161–1168. <https://doi.org/10.1038/s41592-021-01254-9>
- Xu L, Li Z (2020) A new appraisal model of Second-Hand housing prices in China's First-Tier cities based on machine learning algorithms. *Comput Econ* 57:1–21
- Yu H, Fotheringham AS, Li Z, Oshan T, Wolf LJ (2020) On the measurement of bias in geographically weighted regression models. *Spat Stat* 38:100453
- Yuan X, Chen H, Liu B (2020) Point cloud clustering and outlier detection based on spatial neighbor connected region labeling. *Meas Control*. <https://doi.org/10.1177/0020294020919869>
- Zhou YW, Hu ZZ, Lin JR, Zhang JP (2019) A review on 3D spatial data analytics for building information models. *Arch Comput Methods Eng* 27:1–15

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