University of Warsaw
Faculty of Economic Sciences

Working Papers
No. 16/2021 (364)

SPATIAL MACHINE LEARNING – NEW OPPORTUNITIES FOR REGIONAL SCIENCE

Katarzyna Kopczewska

Warsaw 2021

WORKING PAPERS 16/2021 (364)

Spatial Machine Learning – New Opportunities for Regional Science

Katarzyna Kopczewska

University of Warsaw, Faculty of Economic Sciences,
kkopczewska@wne.uw.edu.pl

Abstract: This paper is a methodological guide on 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 and traps of using this developing methodology. It catalogues and comments on the usage of spatial clustering methods (for locations and values, separately and jointly) for mapping, bootstrapping, cross-validation, GWR modelling, and density indicators. It shows details of spatial machine learning models, 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 inspirations to transplant modern quantitative methods from other thematic areas to research in regional science.

Keywords: spatial machine learning; clustering; spatial covariates, spatial cross-validation, spatial autocorrelation

JEL codes: C31, R10, C49

Introduction

Machine learning (ML), since its growth in the 1980s, has attracted the attention of many disciplines based on quantitative methods. Machine learning uses automated algorithms to discover patterns from data and enable high-quality forecasts, although the relations between input data have not been widely studied. This is contrary to classic statistics and econometrics, which are designed to make inferences and test hypotheses to conclude on population having a sample and using equations, while forecasts are of secondary importance. ML often works as a black-box, 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 to known groups based on a trained model, and prediction. According to Google Ngrams, its current applications are ca. ten times more frequent than econometrics, but still ca. seven times less frequent than statistics. In many research areas (such as epidemiology, geology, ecology, climate, etc.), it has become a standard, but we still wait for that wave in regional science.

Spatial methods need spatial data. Recent assessment is that around 80% of all data can have a geographic attribute, and many of them can be geo-referenced (VoPham et al., 2018). Spatial information can stem from conventional sources such as statistical offices regional databases, grid datasets and geo-located points. One can also easily get data from OpenStreetMap and GoogleMaps as background maps, points of interest (POI), roads, traffic, etc., as well as from geo-referenced images such as satellite photos, night light photos, drone photos, and also geotagged social media posts on Twitter or climatic sensors. This type of data requires powerful computational methods due to its complexity, diversity and volume.

Machine learning is commonly linked to big data, artificial intelligence and deep learning[1], but it also works alone. One may implement the simple self-standing machine learning forecast on ready-to-use data, or use it with workflow and data processing or end up with artificial intelligence where algorithms make decisions[2]. From the current standard narrative, one can have an impression of ML methods' inaccessibility for a wider audience.

[1] **Artificial intelligence** (AI) is often defined as a "*moving target*" with regards to technological challenges; its main feature is to make decisions. Early examples of AI include computers playing chess, and nowadays it is an autonomous car. **Deep learning** is a part of machine learning. It does not require specifying parameters as in machine learning, as it discovers these through self-teaching with a multi-layered neural network.

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

However, ML has a vast potential in non-big data analyses by using those methods as supplements to spatial statistics and econometrics.

The goal of this paper is to present the methodological overview of machine learning in the spatial context. First, it shows what information ML gives and concludes if ML is substitutive or complementary to the traditional methods. Secondly, it presents two ways that ML has been incorporated into spatial studies – by using typical ML on spatial data and developing new ML methods dedicated to spatial data only. Thirdly, it aims to promote the transfer of ML to regional science. The paper concentrates only on 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 general overview of these methods was presented in Appendix 1 and their R implementation in Appendix 3. Other ML methods as dimension reduction, association rules, reinforcement learning, neural networks and ensemble methods are not addressed.

1. 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 the relations or causality but 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 interest is in geographical location. The methodological question is how to address this unique specificity of spatial data. The separation between observations is measured with distance. It can be an intuitive shortest (Euclidean) distance from a point to 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 replace previous innovations, and some solutions have transpired to be a dead end. The discussion below presents

[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 forget about clustering tasks.

an overview of these diverse methods, with their development track and usefulness in spatial analysis[4].

Clustering of points in space

Geo-located points, independently of having features assigned, are characterised by the longitude and latitude (*x,y*) of 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-size sample *n*, one can use the *k-means* algorithm, mostly with Euclidean distance metrics. It works well for limited *n*, as it requires the computation of resource-consuming *n x 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 to minimise distances between points within a cluster. In 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 x n* distance matrix is solved by sample shrinking with sampling; PAM suffers the same as *k-means*. Quality of clustering is typically tested with silhouette or gap statistics (see Appendix 1). This mechanism can be applied to design catchment areas (e.g. for schools, post offices, supermarkets) or to divide the market for sales representatives – both challenges are to organise individual points around centres, with possible consideration of capacity and/or fixed location of the centre. Aside from statistical grouping, clustering has a huge potential for forecasting. A calibrated clustering model enables the automatic assignment of new points to established clusters. The prediction mechanism works based on *k nearest neighbours*.

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, it builds the dendrogram – continuous division into 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 the state

[4] Increasingly one can find in the literature the comparison 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 *nxn* distance matrix can be simplified by the Fastmap and modified Fastmap algorithm.

when a single cluster is created. The final result – is the assignment of points to clusters, which is the same as in k-means or PAM and CLARA.

Clustering with the *k-means* algorithm has the significant advantages of ease of interpretation, high 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 to run spatially stratified sampling from those clusters (e.g. Russ & Brenning, 2010; Schratz et al., 2019). This solves the problem of inconsistency in bootstrapping (Chernick & LaBudde, 2014; Kraamwinkel et al., 2018) and addresses the autocorrelation in cross-validation (discussed further). *K-means* irregular partitioning can also be applied in the block bootstrap (Hall et al., 1995; Liu & Singh, 1992). Sampling blocks of data from spatially pre-defined subsamples allows for drawing independent blocks of data but lowers the computational efficiency.

Clustering of features regardless of location

Features measured in regions (or territorial units) can also be clustered to form possibly homogenous clusters, which are later mapped. A very interesting example of a spatial study with hierarchical clustering visualised with a dendrogram analyses fire distribution in Sardinia. It evidences phenological metrics as well as spatio-temporal dynamics of the vegetated land surface (NVDI, Normalized Difference Vegetation Index from satellite photos) (Bajocco et al., 2015) of each territorial unit. Hierarchical clustering groups the territorial units into similarly covered areas. For each cluster group, one checks the fire frequency to assess the natural conditions that increase and decrease fire-proneness[6].

[6] Clusters are not always derived with 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 Gi statistics for body mass index (BMI) and sugar-sweetened beverages intake frequency (SSB-IF) and concluded "optically" from visualisation about spatial agglomeration of high and low values of Gi.

Non-spatial *k*-means clustering may also help in the detection of urban sprawl. Liu et al. (2018) 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 $\sum_{i=1}^{i=9} p_i = 1$. 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 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 to a few clusters with some probabilities. This multi-cluster-assignment uncertainty can be mapped with Scaled Shannon Entropy Index (SSEI). In the Hengl et al. (2017) study, SSEI reflected the density of sample points and extrapolation effects.

Clustering of locations and values simultaneously

The clustering of locations and values in individual procedures presented above can be linked. In literature, one can find a few concepts of spatially-restricted clustering. 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 (nonspatial-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*) 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 makes the list of contiguity, and for each neighbour, it calculates the cost – 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).

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 with mixing parameter α. It weights both dissimilarity matrices[7], D_0 for values, and D_1 for locations, to increase the clusters' spatial coherence. ClustGeo (CG) 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 Hamming distance based on dissimilarity measures (Silhouette, Dunn, etc.) and used in CG to obtain final partitioning, which minimises the within-cluster inertia. The BCG approach out-performs 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 & 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 within-cluster sum of squares, which is to subtract within the clusters the individual values from the cluster average. This cluster average was replaced by a magnitude-weighted average, which shifts the centroids of a cluster into the strongest earthquakes.

Spatially-oriented *k*-means 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 - technically being equivalent to time-series. Alexandrov and Kobarg (2011) proposed a spatially-aware *k-means* clustering. As with every *k-means*, it is based on a dissimilarity (distance) matrix between pixels. To compare the distance between pixels, they derive a composite distance between their spectra. Instead of a direct comparison of two spectra of both pixels, it compares two weighted spectra, each averaging the neighbouring spectra in radius *r*, similarly to the spatial lag concept. Even if *k-means* clustering itself has no spatial component, the distances used in clustering include neighbourhood structure.

[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**.

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 from 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 and makes those values challenging to summarise traditionally. Mapping of the clustered regression coefficients enables its efficient overview. As many studies show (e.g. Lee et al., 2017), clusters are predominately continuous over space, even if computations do not include explicitly locational information.

This output – 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 the 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 model public transportation services. Third, one can model spatio-temporal stability (Kopczewska & Ćwiakowski, 2021). For each period, GWR coefficients are estimated and clustered separately. Next, they are rasterised, and for each raster cell one calculates median, or mode values of cluster ID. Finally, one applies the Rand Index and/or Jaccard similarity to test the temporal similarity of the median/mode cluster ID in each cell. This approach, originally applied to housing valuation, can test spatio-temporal stability of clusters in any context. Fourth, 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 deriving robust submarket division.

It is not only GWR coefficients that can be clustered. In general, clustering requires multiple values to be grouped. This appears in bootstrapped regression. The majority of literature runs bootstrapped OLS 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 was not 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.

Clustering based on density

The above-discussed clustering has three main features: a) an algorithm used a distance matrix; b) all points or regions were classified to one of the clusters; 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 simplification, it screens the surroundings of each point iteratively by checking if 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, which automatically increases its efficiency in big data applications. It also does not assume any parametric distributions, cluster shapes or the 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 & 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., 2019). Joshi et al. (2013) run multi-dimensional DBSCAN for polygons, in which spatial ε-neighbourhood (points in a radius of ε) is substituted with spatio-temporal neighbourhood. Khan et al. (2014) and Galán (2019) review the newest advances in DBSCAN and their applications.

DBSCAN finds many applications. Pavlis et al. (2017) estimate with DBSCAN the retail spatial extent. To address local variability, they use 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 finds an application in the classification of objects from imaging with an airborne LIDAR technique (Wang et al., 2019), WLAN Indoor Positioning Accuracy (Wang et al., 2019) and traffic collision risk in maritime transportation (Liu et al., 2020). DBSCAN may also work on text data and computer codes. Mustakim et al. (2019) run DBSCAN on the cosine distance obtained for text representation (Frequency-Inverse Document Frequency and Vector Space Model) and check 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 was 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 introducing DBSCAN, there were a few other concepts of scanning statistics, constructed based on a moving circle - GAM (*Geographical Analysis Machine*), BNS (*Besag-Newell Statistic*) and spatial scan statistics. GAM (Openshaw et al., 1987) works on point data within a rectangle, 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 compares with the expected number of points from Poisson distribution (e.g. population) or other phenomena cases. The significant circle is the output. BNS (Besag & 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 then compares 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 were almost forgotten. An interesting progressive method stemming from GAM is a scan test for spatial group-wise heteroscedasticity in cross-sectional models (Chasco et al., 2018).

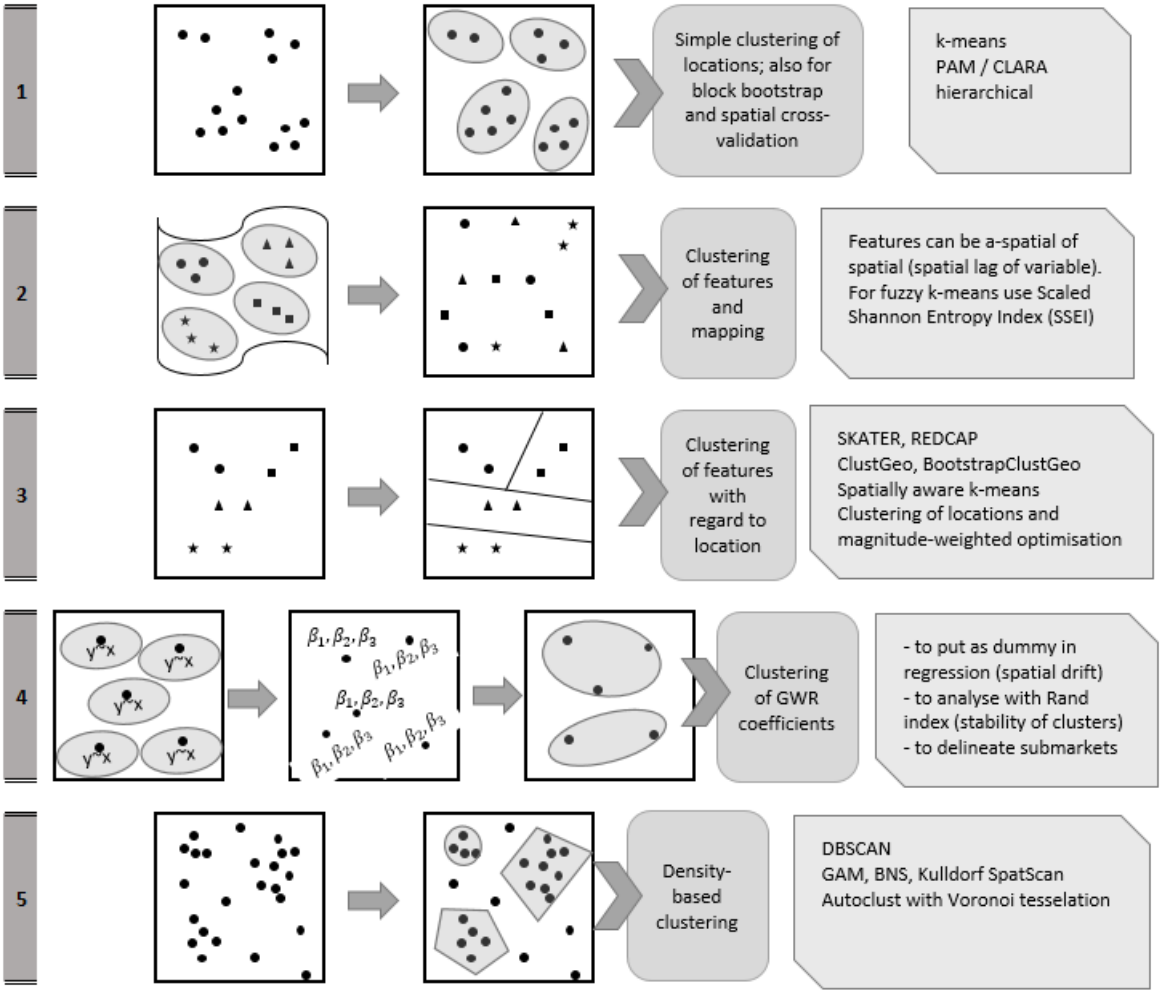
After DBSCAN[8], there appeared a group of methods based on Voronoi / Dirichlet tessellation (Estivill-Castro & Lee, 2002; Lui et al., 2008), called Autoclust. In the Voronoi diagram, for each point, they calculate the mean and standard deviation of the tile's edges. In dense clusters, all edges are short; in the case of border points, a variance of edges increases, as one edge is significantly longer than the other. Analysis of edges and border points delineates the borders of density clusters. The biggest advantage is in self-establishing parameters – a number of clusters, which is not the case of *k-means* or DBSCAN. This approach was forgotten and did not become a part of machine learning due to no solutions for predictions. Recently, as a rebirth, one can find proposals of 3D implementations (Kim & Cho, 2019).

Overview of ML spatial clustering

The above-discussed methods differ in their mechanisms, but their goal is similar. In any case, one may ask the question: to which cluster given spatial point belong? Depending on input data, it can be: i) a cluster of spatially close points, ii) a cluster of feature-similar observations, iii) a cluster of spatial and feature close neighbours, 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 also 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).

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

Figure 1: Unsupervised spatial machine learning models



Source: Own concept

This methodological summary can find applications in many regional science problems. It can help find clusters of features and map them in a smart way for huge geo-located data to check if geo-located segmentation exists and if points (customers) are clustered. It can be applied to analyse (co)location patterns with the values – to answer where our customers are and what else they visit and where to set the business, and who will be the best neighbour. Finally, it can help in the reduction of multi-dimensional data.

Machine learning is a mixture of old statistical concepts refreshed by new challenges. Current methodological research efforts go into better forecasting, improving computing efficiency, especially with big data and finding more sophisticated approaches, such as for spatial techniques. Even if this summary tries to describe spatial clustering designs

comprehensively, one can find more literature concepts. One of them is cluster correspondence analysis for multiple point locations to address the same event in many places (Lu and Thill, 2010).

3. Econometric applications of machine learning to spatial data

Machine learning approaches to the dependency between variables is exhibited in another class of models, which differ from traditional econometrics in few aspects: a) even if input data (*x* and *y*) seem similar, the structure of the model itself is much less transparent; b) as the machine learning modelling recalls "beauty contest" and it 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 on 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*. Knowing both features *x* and classes *y* in supervised machine learning is contrary to unsupervised learning, 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 photo) one extracts features of the land (e.g. vegetation index, water index, land coverage) and adds geographical information (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 distinguish the desired image elements by linking information from the photo with the real class, where an image pixel is an individual observation. Further, the model can detect those elements on new photos to predict the class. This is widely applied in agriculture to distinguish crops, landscaping and land use (Pena & Brenning, 2015). It also works in geological mapping (e.g. Cracknell & Reading, 2014). New possible 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 in Appendix 1). The last years also brought 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 are equivalent to classical approaches. Tab.1 presents the latest studies which use the ML toolbox.

Table 1: Usage of machine learning models in spatial applications

Type of model	Examples of usage	Thematic area	Remarks
Naïve Bayes	Park & Bae, 2015	housing valuation	Model worked not the best, as C4.5 and AdaBoost. Much better was RIPPER.
	Cracknell & Reading, 2014	lithology classification	Model worked not the best, Random Forest was better
k-Nearest Neighbours	Cracknell & Reading, 2014	lithology classification	Model worked not the best, Random Forest was better
Random Forest	Cracknell & 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
	Lui et al., 2020	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 & 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.
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 & 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
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 & 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

Source: Own study

Machine learning models are not only more accurate, but also might be much faster. Sawada (2019) reports that applying machine learning and the Markov Chain Monte Carlo approach to a Land Surface Model decreases computation time by 50,000 times.

The general message is that most machine learning methods in spatial applications do not consider relative location and neighbourhood features and analyse pixels regardless of their surroundings. ML models are spatial only by operating on the map but not by including spatial relations. However, many authors have proposed some techniques to address the spatial dimension, which are presented below.

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 spatial data as with any other kind of data – one understands that data is geo-projected and was gained in specific locations, but no spatial information is included. There are many examples. Appelhans et al. (2015) explain temperatures on Kilimanjaro with elevation, hill slope, aspects, sly-view factor and vegetation index – they use machine learning models in a regression, and the only spatial issue is spatial interpolation with kriging[9]. Similarly, Liu et al. (2020) run 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 is the Moran test of residuals, which evidenced no spatial autocorrelation. The power of the study lies in merging different sources of geo-projected data: surface data for elevation, slope, land cover types and natural disasters (with resolution 30m or 1:2000); point data like access to town, market, hospital, bank, school, or industry taken from POI (Point-Of-Interest) or road density network (in scale 1:120000); and polygonal data for the labour force from a statistical office.

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

Rodríguez-Pérez et al. (2020) model the lightning-caused fire in geo-located grid cells in Spain. They use RF, generalised additive model (GAM) and spatial models, where the fact that lightning-caused fire appeared in a given grid-cell was explained with features observed there 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 landslide (e.g. slope, elevation) with point data of interest. The spatial location and autocorrelation are included in spatial cross-validation.

Another interesting example is mapping rural workers' health condition and severe disease exposure (Gerassis et al., 2020) using a ML approach. The study is based on geo-located medical interviews which provided health data – hard medical data and a person's general health condition. With a ML Bayesian Network (BN), the authors discovered which variables are connected with the patient's condition when flagged as ill. In the next step, with binary logistic regression run on individual cases and thresholds from the BN, one gets model classification, and prediction of high disease risk for a person. Spatial methods appear only for interpolation of illness cases observed, which is a separate model - Gerassis et al. (2020) use the Point-to-Area Poisson kriging model, which deals with Spatial Count Data, unequal territories and diverse population composition. The spatial challenge was in different granulation of data: point data in the study sample and polygonal data as a basis of prediction.

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 non-independence. This works by dividing points into *k* irregular clusters (by using, e.g. *k*-means) 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 Receiver Operator Characteristic Curve*), as it is not biased with 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 & Bae, 2015; Xu & Li, 2020).

Image recognition in spatial classification tasks

One of typical implementations of ML is image recognition in spatial classification tasks. A good example is a supervised lithology classification, i.e. geological mapping (Cracknell & Reading, 2014). As input (X), they use the airborne geophysics and multispectral satellite data, while as output (Y) for a given territory, they use the known lithology classification, given as polygons on the image for each class. They also know the *xy* coordinates of the pixels of those images. In the modelling process, they produce an algorithm which discovers the lithology classification from airborne geophysics and multispectral satellites. They run three kinds of models on pixel data: i) $X \rightarrow Y$, ii) *xy coords* $\rightarrow Y$, iii) X & *xy coords* $\rightarrow Y$, using aforementioned NB, kNN, RF, SVM and ANN. In fact, this phase is image processing to teach software to understand what is in the picture and give a lithology class to each pixel. 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 eleven conditioning factors such as elevation, slope degree, slope aspect, profile and plan curvatures, topographic wetness index, distance to roads, distance to rivers, normalised difference vegetation index, land use and land cover and lithology to predict landslide data. 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) model the earthquakes in Chile. Their dataset of seismic events included a period of 17 years, with 86000 geo-located cases in 6575 days. For each day with an earthquake, they created a grid-based image (1°x1°) of the territory with grid-intensity 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 meaningless for spatial process images such as paintings or faces can predict the environmental phenomena well. This finding was the basis of deepened studies (Behrens et al., 2020) which concluded in two major points. First, spurious correlations without causality raise the danger of meaningless but efficient predictors, which can be mitigated by using domain-relevant and structurally related data. Second, by comparing the variograms of regressors, they recommend using covariates with the same or narrower range of spatial dependence of the

dependent variable. Meyer et al. (2019) derive similar conclusions that highly correlated covariates result in over-fitted models, which replicate data well and fail in spatial predictions.

Mixtures of GWR and machine learning models

An example of taking one step further from the classical analysis is a transformation of Geographically Weighted Regression (GWR) into a machine learning solution. The process behind GWR lies in applying small local regressions on neighbouring points for each observation instead of one global estimation. Additionally, one decides on: 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, iii) the weighting scheme – if 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 better capturing local variability. It gave a 12–16% improvement in R^2 and a decrease in RMSE (Root Mean Square Error).

According to Fotheringham et al. (2017), traditional GWR should rather be substituted by Multiscale Geographically Weighted Regression (MGWR). In MGWR, one decides on bandwidth not only with regard to location / local density but allows for optimisation of covariate-specific bandwidth. MGWR performs better than simple GWR. In both approaches, the problem of bias when "borrowing" data from territories with a different local process is very small (Yu et al., 2020)[10].

Spatial variables in machine learning models

It has become very popular to replace geostatistical models with machine learning solutions to model and interpolate spatial point patterns. In fact, current literature compares geostatistical models such as regression kriging and geographically weighted regression, prediction models such as ordinary kriging and indicator kriging, multiscale methods such as ConMap and ConStat and contextual spatial modelling with ML models.

In 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),

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

simply checked the efficiency of non-spatial ML models in spatial tasks. Mostly they have combined RF or SVM with ordinary kriging or inverse distance squared. Often Random Forest became the most accurate method, which increased its popularity in further studies. This approach is still present. For example, Sergeev et al. (2019) predict the spatial distribution of heavy metals in soil in Russia by applying a hybrid approach: simulating a general non-linear trend using an artificial neural network (ANN) (by applying the generalised regression neural network and multilayer perceptron) and fine-tuning the residuals with the classical geostatistical model (residuals kriging)[11].

Later solutions try to include spatial components among covariates - coordinates or distances between other points. Hengl et al. (2018) promote using a buffer distance among covariates of Random Forest. 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) show on a few empirical examples that this solution is equivalent to regression kriging but 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 use the Random Forest model with spatial information to predict zinc concentration having only its geo-location. They considered PCA reduction of dimensions in distance vectors and used kriging for expanding predictions on new locations. They underline a trade-off between including coordinates, which give lower model precision and do not allow for controlling spatial autocorrelation, but which do not overload computational efficiency, including distance matrix, which works oppositely. 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 cover most autocorrelation (Liu, 2020). The propositions of Ahn et al. (2020) and Liu (2020) may expand implementations of Random Forest for spatial data, which work for prediction of 2D continuous variables with and without covariates, binominal and categorical variables, and also with extreme values, spatio-temporal and multivariate problems (Hengl et al., 2018). In general, Random Forest, compared with geostatistical models, requires less spatial assumptions and performs better with big data.

[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.

An alternative approach in 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, one derives for each point of territory seven EDF covariates: 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.

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 construction 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 removing correlated covariates and regularisation to cope with multicollinearity (Farrell et al., 2019), which do not significantly impact the results - random forest on raw data performed the best; however, spatial autocorrelation was not addressed. There are also some controversies. Meyer et al. (2019) assess the inclusion of spatial covariates by quality measures such as Kappa or RMSE. They claim that longitude, latitude, elevation, the Euclidean distances (also as EDF) can be unimportant or even counterproductive in spatial modelling and recommend eliminating those regressors from models. They underline two other aspects: firstly, contrary to the majority narrative, they do not approve of the high fit of ML models, treating them as over-optimistic and misleading; secondly, they claim that in visual inspection, one observes artificial linear predictions resulting from the inclusion of longitude and latitude, and their elimination helps in making predictions real.

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 (see Fig.2). All of them include spatial components.

1. **Data integration**: The core point of many current spatial machine learning studies lies in integrating 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 photos, spectral data, digital elevation models, vegetation and green leaf indices etc.). The individual observation can be of a diverse form: point, polygon, grid or pixel. Depending on the

researcher's choice of data target granulation, the dataset integration process may be only technical or 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 econometric[12] algorithms when obtaining a mutual relation 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. Predominantly, ML models on spatial data have neglected issues of spatial autocorrelation between observations. The latest studies, however, try to incorporate this element 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 enables not only reproducing the training data well but also predictions in new locations beyond the dataset (Meyer et al., 2019). GWR-like local machine learning regression bridge 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 of ML models when testing on fully randomly sampled observations. Current literature suggests that not addressing autocorrelation falsely improves the model quality, 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-estimating if 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 versions, which expands results from observations on all possible points within the analysed territory. In

[12] Spatial econometrics due to its inherited spatial weights matrix, deals with neighbourhood, tracks the spillover and importance of relative location, and technically improves 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 Central Business District, on the border, at the seaside, in the main city) measure the effect of absolute location and special spatial features.

classification tasks primarily based on pixel data, the calibrated models are fed with a new image that allows running prediction for all input pixels.

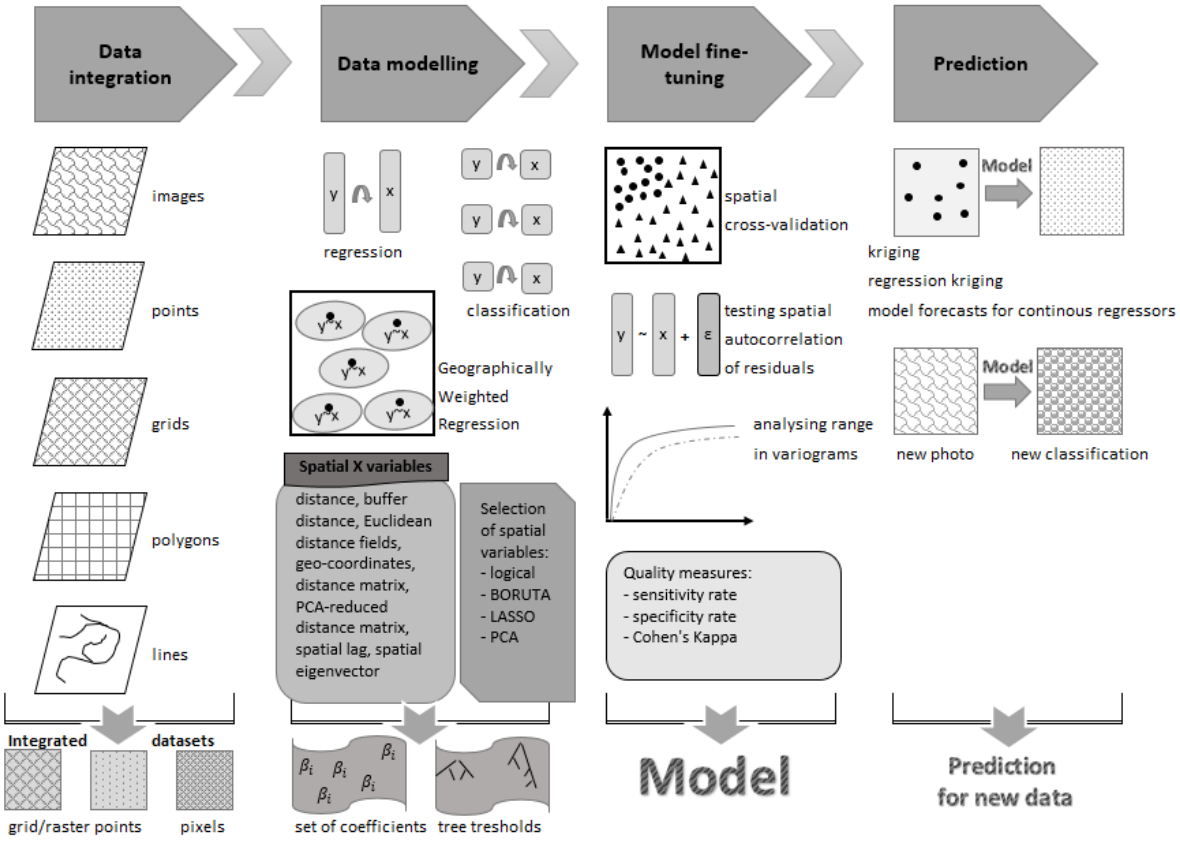
The literature overview shows that spatial machine learning modelling has undergone development with visible progress. In the last decade, one could observe the following approaches:

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 with "4) Classic ML + spatial selected variables + spatial cross-validation". Models estimated with 1), 2) or 3) may not be fully reliable, due to the autocorrelation issues discussed above. The progress and innovations which are coming with approach 5) are mostly referring to ML methods to incorporate spatial components into the algorithms.

It is clear from many studies that unaddressed spatial autocorrelation generates problems: overoptimistic fit of models, omitted information, and/or biased (suboptimal) prediction. Thus, a current toolbox dealing with spatial autocorrelation should be used in all ML models 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 a with 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 makes them more appropriate in the case of big spatial data.

Figure 2: Spatial machine learning modelling



Source: Own concept

4. Perspectives of spatial machine learning

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

Firstly, these methods enable more efficient computation in the case of big data and including new sources of information. Switching from regional data into a lower aggregation level such as individual points or pixels of the image causes datasets to increase in magnitude many times. This low granulation is especially painful for classical spatial econometrics based on an *nxn* spatial weights matrix W or *nxn* 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 already with 30,000 observations, the creation of W is challenging (Kopczewska, 2021). ML models, which are free of W, are automatically quicker, but addressing the autocorrelation issue, currently treated as obligatory, is executed in another way. New sources of data such as lightmaps of terrain (Night Earth, Europe At Night, NASA, etc.) or day photos of landscape

(Google Maps, Street View etc.) bring new insights and information, and due to big-data robust analytics are useful (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 concentrated 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 integrating classical statistics and econometrics with machine learning provides more instruments to the modelling toolbox than a single approach.

Thirdly, the methods open a path for spatio-temporal modelling and studying the similarity of different layers: spatial, multi-dimensional, and spatio-temporal etc. The dynamics connected to location can be addressed in more ways than the classical panel model.

Fourth, these methods allow for better forecasting due to inherited boosting and bootstrapping in ML algorithms. ML results are also more flexible for spatial expanding into new points. Ensemble methods, popular in ML, are supporting researchers in finding the best prediction. A shift towards spatial ML from spatial econometrics is also a change from explanation into 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 out-of-sample data, which were not included in W, thus 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).

Fifth, the methods drive innovations such as new indicators based on vegetation index or light indicators. The methods presented also introduce 3D solutions to spatial studies, such as social topography with 3D inequalities (Aharon-Gutman et al., 2018; Aharon-Gutman & Burg, 2019), 3D Building Information Models (Zhou et al., 2019) or urban compactness growth (Koziatek & Dragićević, 2017). There appear urban studies that rely on information from GoogleStreeView, by counting cars, pedestrians, bikers etc. to predict traffic (Goel et al., 2018), or counting urban disorders such as cigarette butts, trash, empty bottles, graffiti abandoned cars and houses etc. to predict neighbourhood disorder (Marco et al., 2017) or counting green vegetation index to predict safety (Li et al., 2015).

This all shows that spatial modelling built on econometrics, statistics and machine learning is the most effective approach. It finds wide applications in epidemiology, health, crime, the safety of surroundings, customers' location, business, real estate valuation, socio-economic development, and environmental impact etc.

Beyond all of that, the ML approach can still answer typical questions, which were asked over the last years in quantitative regional studies. On the one hand, they are to track invisible policy and its impact on observable phenomena – by studying policy flows, core-periphery patterns and its persistence, urban sprawl patterns, diffusion and spillover from the core to periphery, cohesion and convergence mechanisms, institutional rent, effects of administrative division, the role of infrastructure or agglomeration effects. On the other hand, these can be opposite studies, analysing visible spatial patterns to conclude about unobservable policy, such as studying clusters, tangible flows such as trade or migrations, similarity and dissimilarity of locations, spatio-temporal trends, spatial regularities on labour markets, in GDP and its growth, in education, customers' location and movements as well as business development, location and co-location. Those studies mostly answer the questions on spatial accessibility, spatial concentration and agglomeration, spatial separation, spatial interactions and spatial range.

Progress in science over the past decades involves interdisciplinary transfers of knowledge and methods. Regional science also waits for that transfer. The presented overview of recent papers proves that it has already started, but still waits for mass interest from researchers.

References

Aharon-Gutman, M., Schaap, M., & Lederman, I. (2018). Social topography: Studying spatial inequality using a 3D regional model. Journal of Rural Studies, 62, 40-52.

Aharon-Gutman, M., & Burg, D. (2019). How 3D visualisation can help us understand spatial inequality: On social distance and crime. Environment and Planning B: Urban Analytics and City Science

Ahn, S., Ryu, D. W., & Lee, S. (2020). A Machine Learning-Based Approach for Spatial Estimation Using the Spatial Features of Coordinate Information. *ISPRS International Journal of Geo-Information*, *9*(10), 587.

Alexandrov, T., & Kobarg, J. H. (2011). Efficient spatial segmentation of large imaging mass spectrometry datasets with spatially aware clustering. Bioinformatics, 27(13), i230-i238.

Amiri, M., Pourghasemi, H. R., Ghanbarian, G. A., & Afzali, S. F. (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, M. M., Kriegel, H. P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. ACM Sigmod record, 28(2), 49-60.

Appelhans, T., Mwangomo, E., Hardy, D. R., Hemp, A., & Nauss, T. (2015). Evaluating machine learning approaches for the interpolation of monthly air temperature at Mt. Kilimanjaro, Tanzania. Spatial Statistics, 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?. Regional Science and Urban Economics, 76, 67-73.

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.

Aydin, O., Janikas, M. V., Assunção, R., & Lee, T. H. (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., Schmidt, K., Viscarra Rossel, R. A., Gries, P., Scholten, T., & MacMillan, R. A. (2018). Spatial modelling with Euclidean distance fields and machine learning. European journal of soil science, 69(5), 757-770.

Behrens, T., & Rossel, R. A. V. (2020). On the interpretability of predictors in spatial data science: The information horizon. Scientific Reports, 10(1), 1-10.

Besag, J., & Newell, J. (1991). The detection of clusters in rare diseases. Journal of the Royal Statistical Society: Series A (Statistics in Society), 154(1), 143-155.

Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. Data & knowledge engineering, 60(1), 208-221.

Brimicombe, A. J. (2007). A dual approach to cluster discovery in point event data sets. Computers, environment and urban systems, 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. Geomatics, Natural Hazards and Risk, 11(1), 869-886.

Campello, R. J., 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, Heidelberg.

Cecchini, S., Savio, G., & Tromben, V. (2021) Mapping Poverty Rates in Chile with Night Lights and Fractional Multinomial Models. Regional Science Policy & Practice.

Chasco, C., Le Gallo, J., & López, F. A. (2018). A scan test for spatial groupwise heteroscedasticity in cross-sectional models with an application on houses prices in Madrid. Regional Science and Urban Economics, 68, 226-238.

Chen, W., Pourghasemi, H. R., Kornejady, A., & Zhang, N. (2017). Landslide spatial modeling: Introducing new ensembles of ANN, MaxEnt, and SVM machine learning techniques. *Geoderma*, *305*, 314-327.

Chernick, M. R., LaBudde, R. A. (2014). An introduction to bootstrap methods with applications to R. John Wiley & Sons.

Chi, S. H., Grigsby-Toussaint, D. S., Bradford, N., & Choi, J. (2013). Can geographically weighted regression improve our contextual understanding of obesity in the US? Findings from the USDA Food Atlas. Applied Geography, 44, 134-142.

Cracknell, M. J., & Reading, A. M. (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. *Computers & Geosciences*, *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. Automation in Construction, 88, 44-58.

Debnath, M., Tripathi, P. K., & 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. Spatial Statistics, 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. Journal of Geovisualization and Spatial Analysis, 4, 1-25.

Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. 2nd Int. Conf. Knowl. Discovery Data Mining.

Estivill-Castro, V., & Lee, I. (2002). Argument free clustering for large spatial point-data sets via boundary extraction from Delaunay Diagram. Computers, Environment and urban systems, 26(4), 315-334.

Farrell, A., Wang, G., Rush, S. A., Martin, J. A., Belant, J. L., Butler, A. B., & Godwin, D. (2019). Machine learning of large-scale spatial distributions of wild turkeys with high-dimensional environmental data. Ecology and evolution, 9(10), 5938-5949.

Fotheringham, A. S., Crespo, R., & Yao, J. (2015). Geographical and temporal weighted regression (GTWR). Geographical Analysis, 47(4), 431-452.

Fotheringham, A. S., Yang, W., & Kang, W. (2017). Multiscale geographically weighted regression (MGWR). Annals of the American Association of Geographers, 107(6), 1247-1265.

Galán, S. F. (2019). Comparative evaluation of region query strategies for DBSCAN clustering. Information Sciences, 502, 76-90.

Gerassis, S., Boente, C., Albuquerque, M. T. D., Ribeiro, M. M., 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. Spatial Statistics, 100434.

Goetz, J. N., Brenning, A., Petschko, H., & Leopold, P. (2015). Evaluating machine learning and statistical prediction techniques for landslide susceptibility modeling. Computers & geosciences, 81, 1-11.

Goel, R., Garcia, L. M., 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.

Goulard, M., Laurent, T., & Thomas-Agnan, C. (2017). About predictions in spatial autoregressive models: Optimal and almost optimal strategies. Spatial Economic Analysis, 12(2-3), 304-325.

Griffith, D. A., & Liau, Y. T. (2020). Imputed spatial data: cautions arising from response and covariate imputation measurement error. Spatial Statistics, 100419.

Guo, D. (2008). Regionalisation with dynamically constrained agglomerative clustering and partitioning (REDCAP). International Journal of Geographical Information Science, 22(7), 801-823.

Hall, P., Horowitz, J. L., & Jing, B. Y. (1995). On blocking rules for the bootstrap with dependent data. Biometrika, 82(3), 561-574.

Helbich, M., Brunauer, W., Hagenauer, J., & Leitner, M. (2013). Data-driven regionalisation of housing markets. Annals of the Association of American Geographers, 103(4), 871-889.

Hengl, T., Leenaars, J. G., Shepherd, K. D., Walsh, M. G., Heuvelink, G. B., Mamo, T., ... & Kwabena, N. A. (2017). Soil nutrient maps of Sub-Saharan Africa: assessment of soil nutrient content at 250 m spatial resolution using machine learning. Nutrient Cycling in Agroecosystems, 109(1), 77-102.

Hengl, T., Nussbaum, M., Wright, M. N., Heuvelink, G. B., & 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, É., & 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. Astronomy & Astrophysics, 620, A27.

Joost, S., De Ridder, D., Marques-Vidal, P., Bacchilega, B., Theler, J. M., Gaspoz, J. M., & Guessous, I. (2019). Overlapping spatial clusters of sugar-sweetened beverage intake and body mass index in Geneva state, Switzerland. Nutrition & diabetes, 9(1), 1-10.

Joshi, D., Samal, A., & Soh, L. K. (2013). Spatio-temporal polygonal clustering with space and time as first-class citizens. GeoInformatica, 17(2), 387-412.

Khan, K., Rehman, S. U., 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.

Kopczewska, K. (2020) (eds). Applied Spatial Statistics and Econometrics: Data Analysis in R. Routledge.

Kopczewska, K., & Ćwiakowski, 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.

Kopczewska, K. (2021) Spatial bootstrapped microeconometrics: forecasting for out-of-sample geo-locations in big data, forthcoming

Koziatek, O., & Dragićević, S. (2019). A local and regional spatial index for measuring three-dimensional urban compactness growth. Environment and Planning B: Urban Analytics and City Science, 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. Spatial statistics, 28, 194-215.

Kulldorff, M. (1997). A spatial scan statistic. Communications in Statistics-Theory and methods, 26(6), 1481-1496.

Lee, J., Gangnon, R. E., & Zhu, J. (2017). Cluster detection of spatial regression coefficients. Statistics in medicine, 36(7), 1118-1133.

Li, J., Heap, A. D., Potter, A., & Daniell, J. J. (2011). Application of machine learning methods to spatial interpolation of environmental variables. *Environmental Modelling & Software*, *26*(12), 1647-1659.

Li L.,2019, Geographically Weighted Machine Learning and Downscaling for High-Resolution Spatiotemporal Estimations of Wind Speed, Remote Sensing 11 (1378)

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 International Journal of Geo-Information, 4(3), 1166-1183.

Liu, D., Nosovskiy, G. V., & Sourina, O. (2008). Effective clustering and boundary detection algorithm based on Delaunay triangulation. Pattern Recognition Letters, 29(9), 1261-1273.

Liu, D., Wang, X., Cai, Y., Liu, Z., & Liu, Z. J. (2020). A novel framework of real-time regional collision risk prediction based on the RNN approach. Journal of Marine Science and Engineering, 8(3), 224.

Liu, M., Hu, S., Ge, Y., Heuvelink, G. B., Ren, Z., & Huang, X. (2020). Using multiple linear regression and random forests to identify spatial poverty determinants in rural China. Spatial Statistics, 100461.

Liu, R. Y., & Singh, K. 1992, in Exploring the Limits of Bootstrap, ed. R. LePage & L. Billard (New York: Wiley), 225

Liu, X. (2020). Incorporating spatial autocorrelation in machine learning (Master's thesis, University of Twente).

Lovelace, R., Nowosad, J., & Muenchow, J. (2019). Geocomputation with R. Chapman & Hall/CRC The R Series

Lu, W., Han, J., & Ooi, B. C. (1993, June). Discovery of general knowledge in large spatial databases. In *Proc. Far East Workshop on Geographic Information Systems, Singapore* (pp. 275-289).

Lu, Y., & Thill, J. C. (2003). Assessing the cluster correspondence between paired point locations. Geographical Analysis, 35(4), 290-309.

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. Journal of Urban Health, 94(2), 190-198.

Meyer, Hanna, Christoph Reudenbach, Tomislav Hengl, Marwan Katurji, and Thomas Nauss. 2018. "Improving Performance of Spatio-Temporal Machine Learning Models Using Forward Feature Selection and Target-Oriented Validation." Environmental Modelling & Software 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. Ecological Modelling, 411, 108815

Müller, S., Wilhelm, P., & Haase, K. (2013). Spatial dependencies and spatial drift in public transport seasonal ticket revenue data. Journal of Retailing and Consumer Services, 20(3), 334-348.

Nicolis, O., Plaza, F., & Salas, R. (2020). Prediction of intensity and location of seismic events using deep learning. Spatial Statistics, 100442.

Mustakim, I..R. N. G., Novita, R., Kharisma, O. B., Vebrianto, R., Sanjaya, S., Andriani, T., Sari, W.P, 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.

Openshaw, S., Charlton, M., Wymer, C., & Craft, A. (1987). A mark 1 geographical analysis machine for the automated analysis of point data sets. International Journal of Geographical Information System, 1(4), 335-358.

Pavlis, M., Dolega, L., & Singleton, A. (2018). A modified DBSCAN clustering method to estimate retail center extent. Geographical Analysis, 50(2), 141-161.

Park, B., & Bae, J. K. (2015). Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. Expert systems with applications, 42(6), 2928-2934.

Peña, M. A., & Brenning, A. (2015). Assessing fruit-tree crop classification from Landsat-8 time series for the Maipo Valley, Chile. Remote Sensing of Environment, 171, 234-244.

Pourghasemi, H. R., Sadhasivam, N., Kariminejad, N., & Collins, A. L. (2020). Gully erosion spatial modelling: Role of machine learning algorithms in selection of the best controlling factors and modelling process. Geoscience Frontiers, 11(6), 2207-2219.

Reis, J., Costa, M. U. (2015). Incremental DBSCAN for Green Computing. Working Paper, VisionSpace Technologies

Rodríguez-Pérez, J. R., 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 analysis, 40(7), 1418-1437.

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, Heidelberg

Ruß, G., Brenning, A. (2010). Spatial variable importance assessment for yield prediction in precision agriculture.

In International Symposium on Intelligent Data Analysis (pp. 184-195). Springer, Berlin, Heidelberg.

Sawada Y., 2019, Machine learning accelerates parameter optimisation and uncertainty assessment of a land surface model, arXiv:1909.04196 [stat.AP]

Sergeev, A. P., Buevich, A. G., Baglaeva, E. M., & Shichkin, A. V. (2019). Combining spatial autocorrelation with machine learning increases prediction accuracy of soil heavy metals. *Catena*, *174*, 425-435.

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. Ecological Modelling, 406, 109-120.

Soltani, A., Pettit, C. J., Heydari, M., & Aghaei, F. Housing price variations using spatio-temporal data mining techniques. Journal of Housing and the Built Environment, 1-29.

VoPham, T., Hart, J. E., Laden, F., & Chiang, Y. Y. (2018). Emerging trends in geospatial artificial intelligence (geoAI): potential applications for environmental epidemiology. Environmental Health, 17(1), 1-6.

Wang, W., Yang, J., & Muntz, R. (1997). STING: A statistical information grid approach to spatial data mining. In VLDB (Vol. 97, pp. 186-195).

Wang, C., Ji, M., Wang, J., Wen, W., Li, T., & Sun, Y. (2019). 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. (2019). Learning to improve WLAN indoor positioning accuracy based on DBSCAN-KRF algorithm from RSS fingerprint data. IEEE Access, 7, 72308-72315.

Weatherill, G., & Burton, P. W. (2009). Delineation of shallow seismic source zones using K-means cluster analysis, with application to the Aegean region. Geophysical Journal International, 176(2), 565-588.

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. Computational Economics, 1-21.

Yu, H., Fotheringham, A. S., Li, Z., Oshan, T., & Wolf, L. J. (2020). On the measurement of bias in geographically weighted regression models. Spatial Statistics, 38, 100453.

Yuan, X., Chen, H., & Liu, B. (2020). Point cloud clustering and outlier detection based on spatial neighbor connected region labeling. Measurement and Control, 0020294020919869.

Zhou, Y. W., Hu, Z. Z., Lin, J. R., & Zhang, J. P. (2019). A review on 3D spatial data analytics for building information models. Archives of Computational Methods in Engineering, 1-15.

Appendix 1: Overview of quantitative concepts

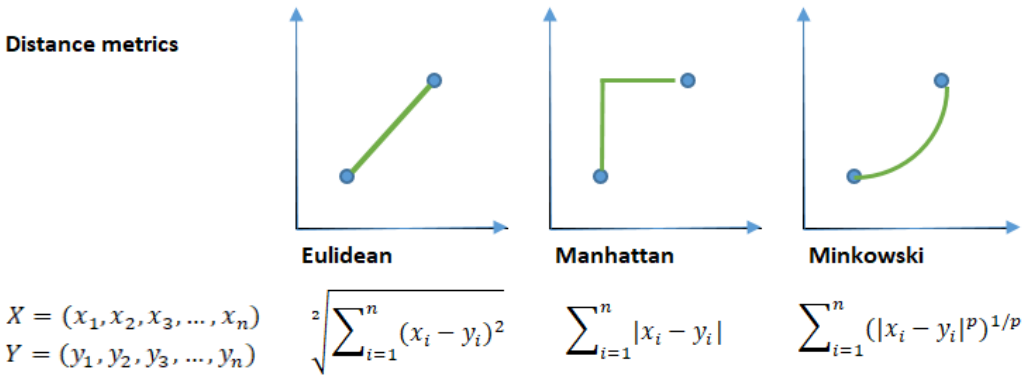
Below one can find the description of the methods mentioned in the paper (distance metrics, clustering with k-means, PAM and CLARA, hierarchical clustering, spatial clustering with SKATER and REDCAP, DBSCAN clustering, Clustering quality measure: silhouette, inertia, Dunn index, k-fold cross-validation, 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, Cubist).

1. 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 (see Fig.A1.1):

- **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 computes the square root of summed up squares of differences between values of variables.
- **Manhattan (urban, city-block) distance** $\sum_{i=1}^{n}|x_i - y_i|$, called also urban distance, which uses perpendicular sections to connect points as moving around the edges of the grid. It compares pairs of observations, variable by variable, and computes 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 generalisation of Euclidean and Mahnattan distance and allows for non-linear curve way between points

Figure A1.1: Distance metrics



Source: Own work

Beyond those three metrics one can use more concepts:

- **Gower distance (also Gower dissimilarity)** – introduced by Gower (1971), can be applied to the mixture of numerical and categorical variables. It compares pairs of observations variable by variable and computes the average distance score between those observations. Components of score are from range [0,1], and their average too. 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 are interpreted as (close) similarity[13].
- **Mahalanobis distance** – introduced by Mahalanobis (1936), includes correlations between variables $\sqrt{(x-y)^T cov(x,y)^{-1}(x-y)}$. To calculate this distance one follows the procedure[14]:
  1. Take real data (let's say three variables x, y, z) and calculate average values of each variable – you get the vector of (three) average values $(\bar{x}, \bar{y}, \bar{z})$
  2. Take your test data (let's say x_i=1, y_i=4, z_i=6)
  3. Calculate 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
  4. Calculate variance-covariance matrix of your data – you get 3x3 matrix – make an inverse of it
  5. Multiply (as matrix): vector of differences * inverse covariance matrix * vector of differences
  6. Take a square root of this multiplication – this is 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 sums up the scores – this counts in how many points the vectors differ. As 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 $cos(0°)$=1. In case of dissimilarity, angle of two vectors increases, and its cosine is in range [0,1). Two vectors being opposite have distance of -1.

[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/

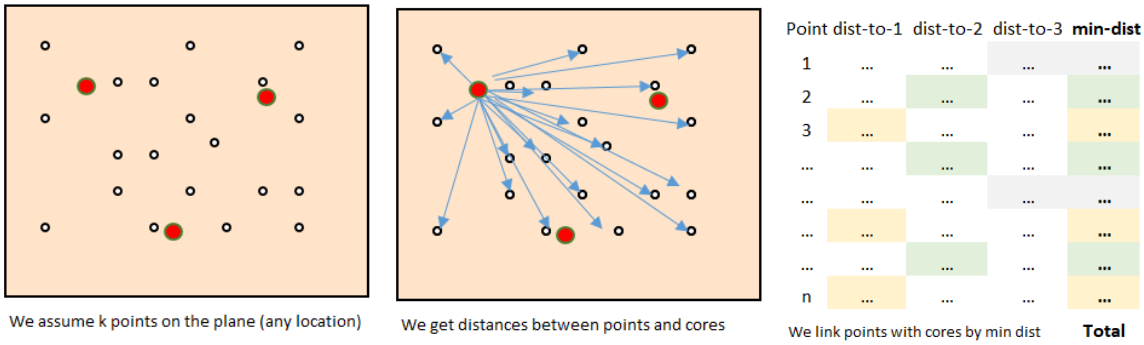
Cosine distance is expresses as: $\frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$, where A and B are analysed vectors (variables). Counter is a sum of products of paired values of both variables. In nominator one gets total of squared values of both variables.

- **Cophenetic distance** – proposed by Sokal and Rohlf (1962), applied only in hierarchical clustering, measures the height of dendrogram between two clusters – 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), called also edit distance, mostly used in text analysis, reflects the minimum number of necessary corrections (delete, insert, substitute) to change one vector into another[15].

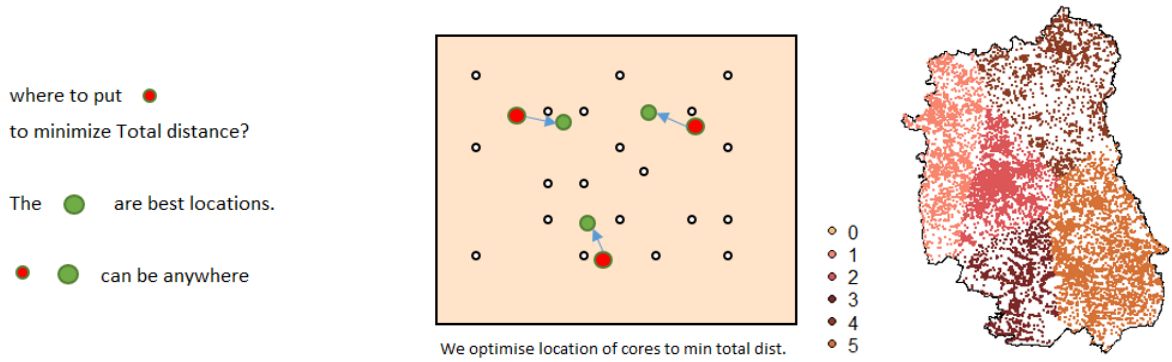
2. Clustering with k-means

The idea of *k*-means was introduced by Steinhaus (1956), the first algorithm was developed by Lloyd (1957), while k-means term was proposed by MacQueen (1967) (see Bock, 2007). In k-means method one assumes a-priori number of clusters *k*, sets initial multi-dimensional coordinates of these *k* centroids, calculates the matrix of distances between all sample points and *k* centroids, and finally optimizes the location of centroids, by minimizing the total distance of points from cores (see Fig.A1.2). Location of centroids is non-restricted and can be in any place of the plane (surface) where the sample data are located. All points are assigned to clusters.

Figure A1.2: Clustering with k-means



[15] https://www.baeldung.com/cs/levenshtein-distance-computation

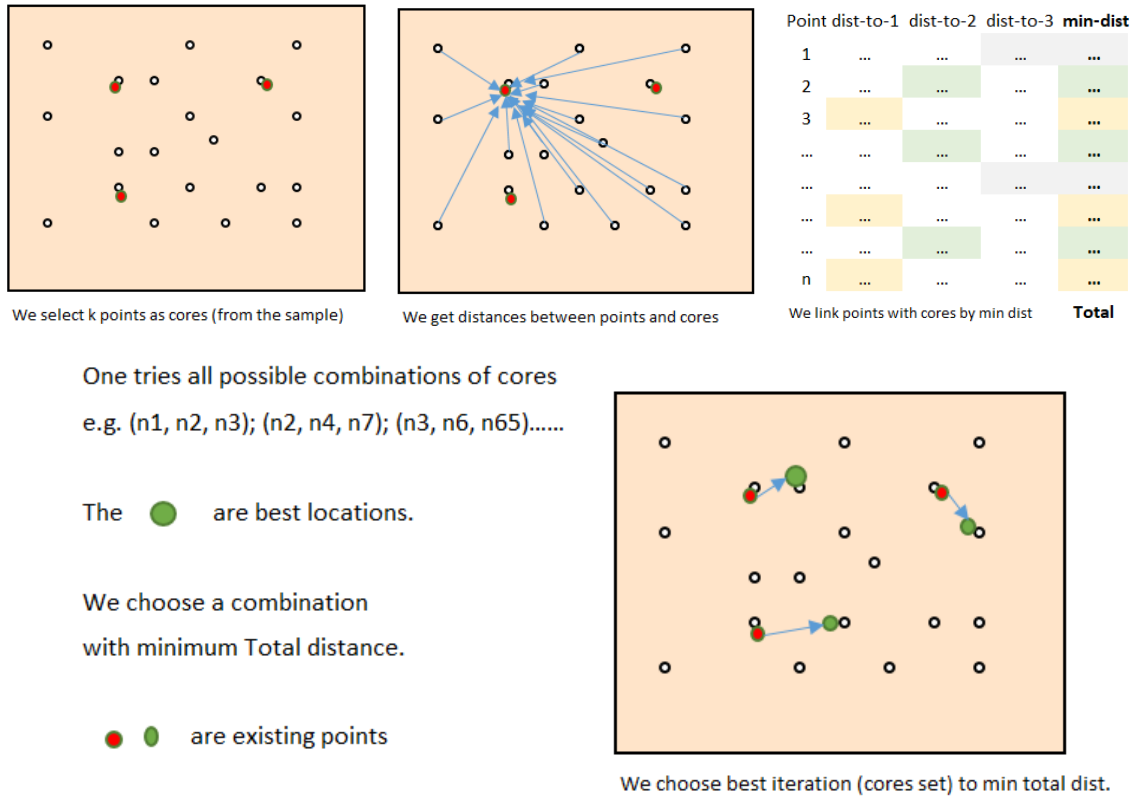


Source: Own work

3. Clustering with PAM and CLARA

The idea of clustering with PAM (*Partitioning Around Medoids*) was introduced by Kaufman and Rousseeuw (1987). Like *k*-means, one assumes a priori *k* core points. 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 minimizing the total distance of points from cores requires iterative approach (see Fig.A1.3). All points are assigned to clusters.

Figure A1.3: Clustering with PAM



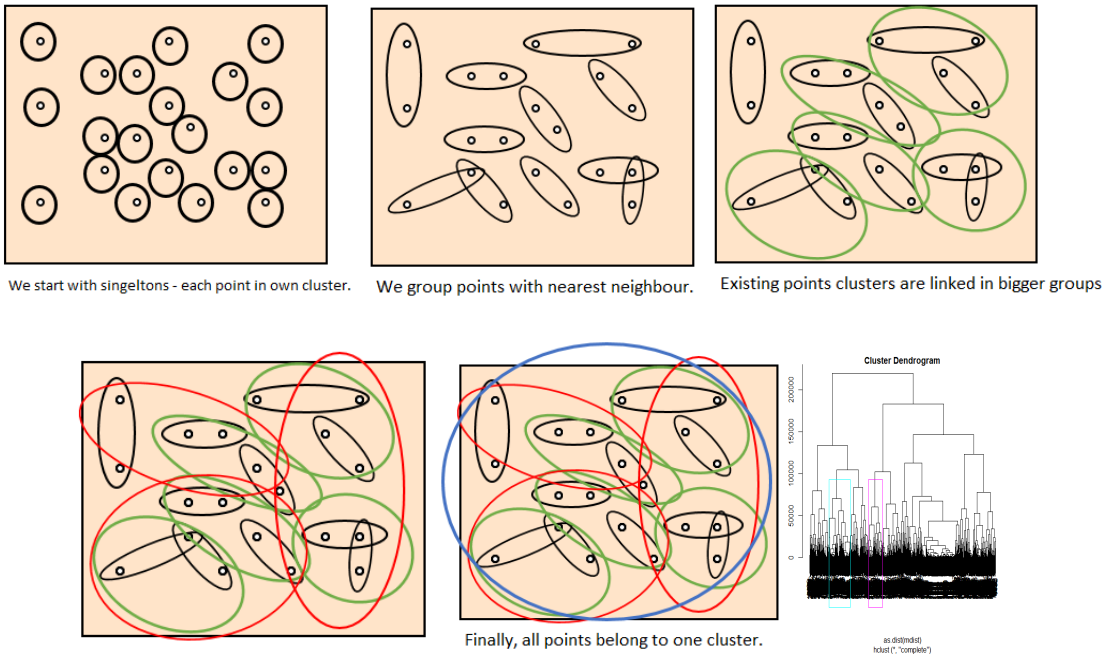
Source: Own work

CLARA (*Clustering Large Applications*) method is big data equivalent of PAM. It was proposed by Kaufman and Rousseeuw (1990). It works as PAM but on subsample, which classifies points to clusters. The rest of points is assigned to clusters using k nearest neighbours.

4. 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. Iteratively, the clusters are merged in bigger groups. In last stage, all observations belong to one cluster. This division can be visualised as dendrogram. To read an output one can decide how many clusters to see or on which high to cut the tree. All observations are assigned to some clusters.

Figure A1.4: Hierarchical clustering



Source: Own work

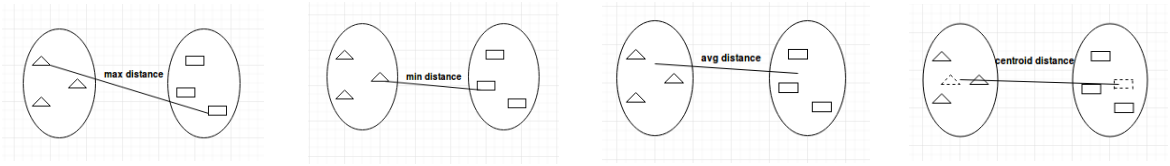
5. Spatial clustering with SKATER and REDCAP

SKATER (*Spatial 'K'luster analysis by tree edge removal*) was proposed by Assuncão et al. (2006). It uses pruning of the trees constructed as a weighted connectivity graph with edge 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 the list of contiguity, and for each neighbour, it calculates the cost – 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.

REDCAP (*Regionalisation with dynamically constrained agglomerative clustering and partitioning*) algorithm was developed by Guo (2008) as an answer for SKATER. It uses hierarchical agglomeration method with spatial constraints. It applies three criteria of defining the distance between values (single linkage, average linkage and complete linkage, see Fig.A1.5) and two "constraining strategies" with regard to spatial location: first order neighbourhood (sharing common border) or full-order neighbourhood (links to all other regions).

Figure A1.5: Definitions of distances between clusters



Complete linkage Single linkage Average linkage Centroid linkage

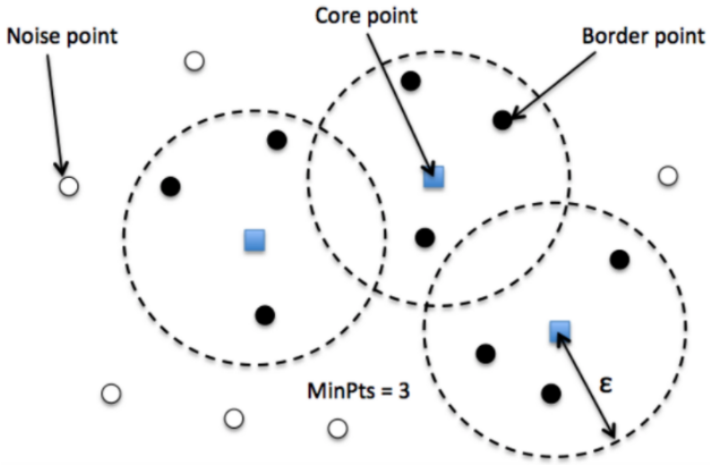
Source: https://www.datacamp.com/community/tutorials/hierarchical-clustering-R

6. 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 e.g. PAM), but examines the spatial density of points to get 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 constituting noise. All points belonging to the cluster are iteratively tested and the full cluster is formed. Subsequently, in the same procedure points are examined, which constitute noise against the previously formed cluster. Points can belong to a cluster (core and border) or stay outside the cluster (noise). DBSCAN requires setting the radius of epsilon ε and the minimum number of points in this radius *MinPts*. For each point one counts the number of points in radius ε and checks if the points fall into the radius of other points. Core points have at least the minimum number of

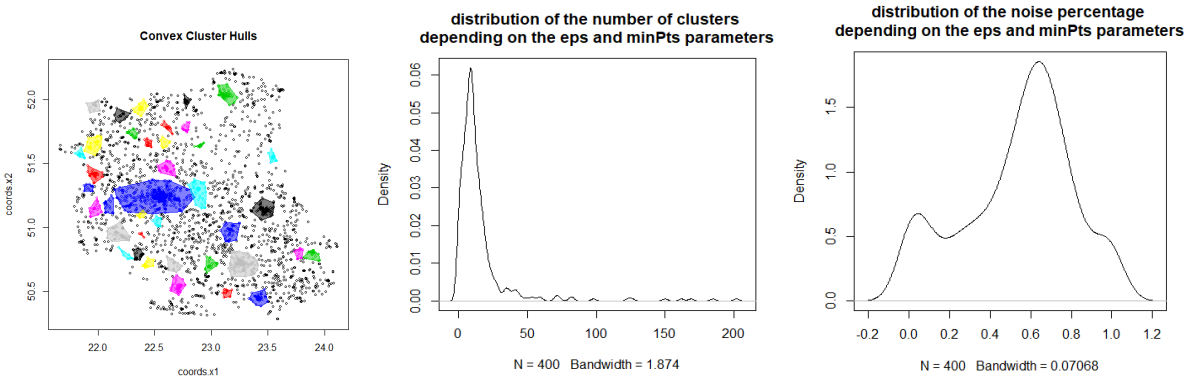
MinPts points within a radius of ε. Border points are in the radius ε from the core point, but do not themselves contain the minimum number of *MinPts* points in their radius ε. Noise points are outside the radius of core and boundary points (see Fig.A1.6). Sensitivity analysis – number of clusters and percentage of noise depending on ε and *MinPts*. Even if method is known as „unsupervised" it requires setting by researcher two parameters, which are crucial for the result (see Fig.A1.7).

Figure A1.6: Search algorithm of DBSCAN method



Source: https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html

Figure A1.7: 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

7. Clustering quality measure: silhouette

Silhouette statistics to test the quality of clustering – in particular if 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∈[-1,1]. The negative values of the silhouette statistics are undesirable, because it means that $a_i > b_i$, so the remaining clusters are closer than your own 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~1), which occurs when the distance between the observation and the middle point in your own cluster is minimal. In the interpretation one is looks for the highest values of the *silhouette* statistics for a different number of clusters k.

8. Clustering quality measure: inertia

Inertia for clusters is a concept similar to analysis of variance and helps deciding which number of clusters works the best. It sums up the weighted squared distance between observations and their cluster center (within-cluster inertia, W); centers of clusters and all observations (between-clusters intertia, B); observations and center of all observations (total inertia, T). Good clustering is characterized by high inter-cluster inertia (diversity) and low intra-cluster inertia (heterogeneity). For two partitioning one compares their Qs (Q=1-W/T) and choses the partitioning 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 ($k = 1$, .., K) clusters and is expressed by:

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

where the individual intra-cluster inertia I(Ck) are determined as:

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

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 in particular 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-clusters inertia B, measures 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:

$$\mu_k = \sum_{i \epsilon C_k} w_i$$

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:

$$T = W + B$$

9. Clustering quality measure: Dunn index

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

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

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

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

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

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

Much more on measures of clustering quality one can find in Vigennets to R package clusterCrit::[16] or in Tibshirani et al. (2000).

10. k-fold cross-validation

Currently one can find two approaches to cross-validation (CV): i) by dividing data into two groups – training and testing, or ii) by dividing data into three groups – training, fine-tuning and testing. When dividing data into two groups – sample is divided into *k* folds (parts, subsamples), k-1 folds is used in training of the model and 1 part is used in testing the model. Process goes recursively *k* times, so each of k folds play a role of testing part of sample. When dividing data into three groups – one keeps part of the data for out-of-sample predictions and does not use these data for model fitting and fine-tuning. The rest of data is used as in approach of dividing data into two groups. In case of 5-fold cross-validation, the data used for model fitting and fine-tuning are divided into 5 equal parts with 20% of data each, and in each of 5 iterations, model is fitted on 80% of data and tested on 20% of data.

11. 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). According to Kuhn and Johnson (2016) regression modelling, except linear regression models (like OLS, Ordinary Least Squares) includes non-linear regressions (based on neural networks, SVM, KNN) and regression trees and rule-based models (like regression trees, random forest, cubist, boosting). Similar division one can have for classification methods, which includes linear models (logistic regression – logit, probit or linear discriminant analysis), non-linear regressions (like neural networks, Support Vector Machines, K Nearest Neighbours, Naïve Bayes), and regression trees and rule-based models (regression trees, random forest, boosting).

[16] https://cran.r-project.org/web/packages/clusterCrit/vignettes/clusterCrit.pdf

12. Naïve Bayes classifier

Naïve Bayes classifier is a statistical model, based on Bayesian probability formula. In phase of building the binary choice model (e.g. class yes/no, more levels also possible), it derives probabilities of each class, and that features X (which include features e.g. x1, x2, …) interact with class - in fact it collects probabilities of appearing given class P(yes), P(no) and features' structure in given class P(x1|yes), P(x2|yes), P(x1|no), P(x2|no). It assumes that features X (e.g. x1, x2, …) are independent of each other. In prediction for new data, it calculates the Bayesian posterior probabilities by using (in 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 observation to given class.

13. 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; 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)

14. Random Forest classifier

Random Forest classifier is an ensemble method (using wisdom of crowds), based on decision trees, which divide selected features into groups to profile given class. Random Forest is a collection of independent trees – they differ as observations are selected in bagging (sampling with replacement, bootstrap) and *m* features are drawn randomly (few variables from bigger set). Majority voting aggregates the results from trees – it takes class by class, checks in the bottom of each tree the output (class) and averages the features' values which are on the path to given class. Quality check follows out-of-bag (*oob*) scheme – when bagging, one divides observations by keeping ca. 2/3 for training and ca. 1/3 for testing the model. Number of features *m* is to be small enough to keep trees uncorrelated and large enough to keep trees strong; it is optimised by controlling the *oob* error rate. *Oob* error rate is frequency that test data did not meet their true value. Variable importance is tested by permuting the values of *m*-th variable among *oob* observations and checking the prediction of trees; difference between ratios

of correct class prediction in non-permutated and permutated tests is variable importance. Good technical overview is available in vignettes of Random Forest software by Breiman and Cutler (see link[17]).

15. Support Vector Machines

In Support Vector Machines the observations are separated into classes with lines (in 2D) or hyperplane (in 3D and more). Support vectors are points in all classes which are closest to the line/hyperplane; distance (called margin) between those points and line/hyperplane should be maximised. In case the points are not linearly separable, they are transformed to make it possible (see introduction in link[18]).

16. Artificial Neural Networks

Artificial Neural Networks (ANN) is classifier method, which operates on binary input and output. Each kind of information (variable, image cell etc.) is analysed by individual perceptron. Numeric data are binarized depending of threshold (e.g. x>a), quantitative data depending on given feature (yes/no). Dummy outputs of perceptrons are weighted and aggregated in additive function – this result is again contrasted with threshold to give binary answer. Answer of ANN is compared with true state. In case of error (expressed as loss function), ANN learns by changing the weights to match the true answer (see introduction in link[19]).

17. Maximum entropy classifier

Maximum entropy classifier is probabilistic model, without assumptions on features independence (oppositely, assumes correlations), using entropy concept. It is based on Bayesian probability formula as Naïve Bayes classifier, but instead of assuming empirical probabilities, it starts with uniform weights and optimizes them (see introduction in link[20]).

18. Autoencoder-based residual network

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

Autoencoder-based residual network are unsupervised learning models, that similarly to PCA extract features from wider dataset. Encoder network transform input image into model (with latent variables), while decoder network reconstructs the image. Residual network adds a layer which gradually learns from residuals (see introduction in link[21]).

19. Gradient boosting

Gradient boosting, which most popular XGBoots, like random forest, is based on decision trees. However, instead of simultaneous growing of all trees (as in random forest), it works iteratively. Next model corrects the mistakes of previous model – misclassifications are analysed, and wrongly predicted observations get higher weights in analysis to be more intensively addressed in the next round. Final model is an additive decision tree, which includes all good models (see introduction in link[22]).

20. Cubist

Cubist algorithm, introduced by Quinlan (1992), is based on tree. For each path (to the ending leaf) it creates a rule with regression multivariate model. Covariates which fulfil the criteria of tree are used in those models. These models are used for predictions and strengthened (averaged) with neighbouring model (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.https://link.springer.com/content/pdf/10.1007%2F978-3-540-73560-1.pdf

[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

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.https://www.amazon.com/Classification-Regression-Wadsworth-Statistics-Probability/dp/0412048418

Cunningham, P., Delany, S.J. (2007), k-Nearest Neighbour Classifiers, Technical Report UCD-CSI-2007-4,https://www.researchgate.net/publication/228686398_k-Nearest_neighbour_classifiers

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. CiteSeerX 10.1.1.121.9220. ISBN 1-57735-004-9.

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.https://www.tandfonline.com/doi/abs/10.1080/13658810701674970

Hamming, R. W. (April 1950). "Error detecting and error correcting codes" (PDF). The Bell System Technical Journal. 29 (2): 147–160. doi: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 https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470316801.ch3

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26). New York: Springer.https://link.springer.com/content/pdf/10.1007/978-1-4614-6849-3.pdf

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..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 https://onlinelibrary.wiley.com/doi/abs/10.2307/1217208

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

Majority of methods discussed in the paper have their software implementations in R. None of other existing software does not offer comprehensive solutions both machine learning and 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 Models[23]) and machine learning (Machine Learning & Statistical Learning[24]). Its applications to environmental data one can find in TaskViews on Analysis of Ecological and Environmental Data[25]. For spatial analysis, one should look into TaskViews to Analysis of Spatial Data[26] and Handling and Analyzing Spatio-Temporal Data[27].

Among huge variety of packages and functions, one can list few which are very interesting:

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 spatial oblique decision tree based on the classification and regression tree,
- **dbscan::** offers density-based clustering with DBSCAN
- **rgeoda::** offers SKATER and REDCUP algorithms[28]

[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
[28] See rgeoda:: vignettes https://rgeoda.github.io/rgeoda-book/ and tutorials https://geodacenter.github.io/tutorials/spatial_cluster/skater.html

- **automap::** offers may versions of kriging
- **StatMatch:**: offers Gower distance

Also, non-covered topics are 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 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

Popular source of data is **MODIS** (*Moderate Resolution Imaging Spectroradiometer*), which contains data from NASA (https://modis.gsfc.nasa.gov/) for whole Earth's surface for every 1-2 days in 36 spectral bands. 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 and ice surface temperature) http://nsidc.org/daac/modis/index.html
- MODIS ocean color and sea surface temperature products (also on carbon, fluorescence line etc.) http://oceancolor.gsfc.nasa.gov/

Planetary Habitability Laboratory offers also satellite images and climate data http://phl.upr.edu/data. There are also many softwares which help in getting proper data (as SAGA, System for Automated Geoscientific Analyses[29]).

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), Normalized Difference Vegetation Index (NDVI), Normalized Green Red Difference Index (NGRDI), Green Leaf Index (GLI) etc. R function rgb_indices() from uavRst:: package[30] 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 analysis on visible spectrum and spatial predictors – first few principal components are used instead of these variables to avoid doubling the information.

Another popular source of data is **LIDAR** (Light Detection and Ranging). They are available from many sources[32] 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 to get variables as Digital Elevation Model (DEM), Slope and aspect (on the basis of DEM) in e.g. radians, geolocation variables as longitude and latitude etc.

[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/

Other interesting information are **Night Light Data**, available from World Bank[33], SOS NOAA (Science on a Sphere, National Oceanic and Atmospheric Administration[34]) and from NASA[35] or Google Earth (earth.google.com).

Many **open data** one can also get from Open Governmental repositories, as data.gov (United States), data.gov.uk (United Kingdom), govdata.de (Germany), https://www.europeandataportal.eu/en (European Union) etc.

[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

University of Warsaw
Faculty of Economic Sciences
44/50 Długa St.
00-241 Warsaw
www.wne.uw.edu.pl