# Dealing with Spatial Autocorrelation when Learning Predictive Clustering Trees

Daniela Stojanova[a,b], Michelangelo Ceci[c], Annalisa Appice[c], Donato Malerba[c], Sašo Džeroski[a,b,d]

[a]*Jožef Stefan Institute, Department of Knowledge Technologies, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[b]*Jožef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia*
[c]*Dipartimento di Informatica, Università degli Studi di Bari "Aldo Moro", via Orabona 4, 70125 Bari*
[d]*Centre of Excellence for Integrated Approaches in Chemistry and Biology of Proteins, Jamova 39, 1000 Ljubljana, Slovenia*

**Abstract**

Spatial autocorrelation is the correlation among data values which is strictly due to the relative spatial proximity of the objects that the data refer to. Inappropriate treatment of data with spatial dependencies, where spatial autocorrelation is ignored, can obfuscate important insights. In this paper, we propose a data mining method that explicitly considers spatial autocorrelation in the values of the response (target) variable when learning predictive clustering models. The method is based on the concept of predictive clustering trees (PCTs), according to which hierarchies of clusters of similar data are identified and a predictive model is associated to each cluster. In particular, our approach is able to learn predictive models for both a continuous response (regression task) and a discrete response (classification task). We evaluate our approach on several real world problems of spatial regression and spatial classification. The consideration of the autocorrelation in the models improves predictions that are consistently clustered in space and that clusters try to preserve the spatial arrangement of the data, at the same time providing a multi-level insight into the spatial autocorrelation phenomenon. The evaluation of SCLUS in several ecological domains (e.g. predicting outcrossing rates within a conventional field due to the surrounding genetically modified fields, as well as predicting pollen dispersal rates from two lines of plants) confirms its capability of building spatial aware models which capture the spatial distribution of the target variable. In general, the maps obtained by using SCLUS do not require further post-smoothing of the results if we want to use them in practice.

*Keywords:* spatial autocorrelation, predictive clustering trees, machine learning

## 1. Introduction

Environmental and ecological data most often have a spatial component or refer to objects/ points placed at specific spatial locations. When analyzing such data, we frequently encounter

*Email addresses:* daniela.stojanova@ijs.si (Daniela Stojanova), michelangelo.ceci@uniba.it (Michelangelo Ceci), annalisa.appice@uniba.it (Annalisa Appice), donato.malerba@uniba.it (Donato Malerba), saso.dzeroski@ijs.si (Sašo Džeroski)

the phenomenon of spatial autocorrelation. Spatial autocorrelation is the correlation among the values of a single variable (i.e., object property) strictly attributable to the relatively close position of objects on a two-dimensional surface, introducing a deviation from the independent observations assumption of classical statistics (Dubin, 1998). Intuitively, it is a property of random variables taking values, at pairs of locations a certain distance apart, that are more similar (positive autocorrelation) or less similar (negative autocorrelation) than expected for pairs of observations at randomly selected locations (Moran, 1950). Positive autocorrelation is common in spatial phenomena, while it has been argued that negative autocorrelation is an artifact of poorly devised spatial units (Goodchild, 1986). Spatial positive autocorrelation occurs when the values of a given property are highly uniform among spatial objects in close proximity, i.e., in the same neighborhood. In geography, spatial autocorrelation is justified by Tobler's first law (Tobler, 1970), according to which "everything is related to everything else, but near things are more related than distant things". This means that by picturing the spatial variation of some observed variables in a map, we may observe regions where the distribution of values is smoothly continuous, with some boundaries possibly marked by sharp discontinuities.

The causes of spatial autocorrelation depend on the specific domain we are dealing with. For instance, in ecological and environmental modeling, where data are typically geo-referenced, four factors are particularly common (Legendre et al., 2002; Legendre, 1993): 1) Biological processes of speciation, extinction, dispersal or species interactions are typically distance-related; 2) Non-linear relationships may exist between species and environments, but these relationships may be incorrectly modeled as linear; 3) Classical statistical modeling may fail in the identification of the relationships between different kinds of data without taking into account their spatial arrangement (Besag, 1974); 4) The spatial resolution of data should be taken into account: Coarser grains leads to spatial smoothing of data.

Most data mining methods and statistical models are based on the assumption that the values of the variable observed in different samples are independent of each other. This is in contrast with spatial autocorrelation, which clearly indicates a violation of this assumption. As observed by LeSage and Pace (2001), "anyone seriously interested in mining sample data which exhibits spatial dependence should consider a spatial model", since this model can accommodate different forms of spatial autocorrelation. They showed how the consideration of autocorrelation of the dependent variable in a predictive modeling task provides an improvement in fitting, as well as a dramatic difference in the significance and impact of explanatory variables included in the predictive model. In addition to predictive data mining tasks, this consideration can also be applied to descriptive tasks, such as spatial clustering. In general, the analysis of spatial autocorrelation is crucial and can be fundamental for building a spatial component into (statistical) models for spatial data.

In this paper, we propose a data mining method that explicitly considers spatial autocorrelation when learning predictive models. The modeling task combines predictive modeling and clustering and takes autocorrelation into account in this context. Predictive clustering combines elements from both prediction and clustering. As in clustering, clusters of data that are similar to each other are identified, but a predictive model is associated to each cluster. This predictive model provides a prediction for the target property of new examples that are recognized to belong to the cluster. The benefit of using predictive clustering methods is that, as in conceptual clustering (Michalski and Stepp, 1983), besides the clusters themselves, they also provide symbolic descriptions of the constructed clusters. However, unlike conceptual clustering, predictive clustering is a form of supervised learning.

We exploit Predictive Clustering Trees (PCTs) (Blockeel et al., 1998), which are an appealing class of models that addresses both predictive and descriptive goals. They are tree structured models that generalize decision trees. In traditional PCTs, the clustering phase is performed by maximizing variance reduction. This heuristic guarantees, in principle, accurate models since it reduces the error on the training set. However, it neglects the possible presence of spatial autocorrelation in the training data. To address this issue, we propose a different clustering phase which uses distances appropriately defined for the spatial domains in order to exploit the spatial structure of the data in the PCT induction phase and obtain predictive models that naturally deal with the phenomenon of spatial autocorrelation.

The consideration of spatial autocorrelation in clustering has been already investigated in the literature (Glotsos et al., 2004; Jahani and Bagherpour, 2011). Motivated by the demonstrated benefits of considering autocorrelation, in this paper, we exploit some characteristics of autocorrelated data to improve the quality of PCTs. The consideration of spatial autocorrelation in clustering offers several advantages, since it allows us to:

- determine the strength of the spatial arrangement on the variables in the model;
- evaluate stationarity and heterogeneity of the autocorrelation phenomenon across space;
- identify the possible role of the spatial arrangement/ distance decay on the predictions associated with each of the nodes of the tree;
- focus on the “spatial neighborhood” to better understand the effects that it can have on other neighborhoods and vice versa.

These advantages of considering spatial autocorrelation in clustering, identified by Getis (Arthur, 2008), fit well into the case of PCTs. Moreover, as recognized by Griffith (2003), autocorrelation implicitly defines a zoning of a (spatial) phenomenon: Taking this into account reduces the effect of autocorrelation on prediction errors. Therefore, we propose to perform clustering by maximizing both variance reduction and cluster homogeneity (in terms of autocorrelation) simultaneously when considering the different attempts for adding a new node to the tree.

In PCTs induced by considering autocorrelation, different effects of autocorrelation can be identified and considered at each node of the tree. This non-stationary view of autocorrelation is global at the root node and local at the leaves. At the same time, the tree structure allows us to deal with the so-called “ecological fallacy” problem (Robinson, 1950), according to which individual sub-regions do not have the same data distribution as the entire region.

Although there is no theoretical proof that the consideration of autocorrelation may increase the accuracy of the learned models, our intuition is that it should still improve predictions that are consistently clustered across the space. This is due to the fact that clusters try to preserve the spatial arrangement of the data. In spite of the fact that the learned models are not obtained by purely minimizing the error on the training set, it is possible that considering autocorrelation in the training data makes the learned predictive model able to contain better knowledge of the underlying data distribution. This knowledge may yield better performance in the testing set than that obtained by a predictive model induced by maximizing only the variance reduction on the training data.

An initial implementation of the method for considering spatial autocorrelation when learning PCTs for regression was presented at a machine learning conference (Stojanova et al., 2011). In this paper, we present significant further developments and extensions in the following directions:

- The method for taking into account autocorrelation when learning PCTs has been extended to consider classification tasks in addition to regression tasks.
- New empirical evidence is provided on the importance of considering spatial autocorrelation in predictive tasks and in particular on the ability of the proposed method to capture autocorrelation within the learned PCTs by analyzing the autocorrelation of the errors on an extensive set of different data.
- The scope of the performance evaluation is broadened to include clustering evaluation in terms of spatial dispersion of extracted clusters.
- We give a comprehensive review of state-of-the-art methods, proposed both in spatial statistics and data mining, which explicitly consider autocorrelation.

## 2. Related Work

The motivation for this work comes from research reported in the literature for spatial autocorrelation and predictive clustering. In the following subsections, we report related work from both research lines.

### *2.1. Spatial Autocorrelation*

Most theoretical research in Statistics and Econometrics exploits the so called "spatial autoregressive" (SAR) model in order to measure autocorrelation in a "lattice". More formally, the spatial autoregressive model is defined as:

$$\hat{e}_i = \rho \sum_{j=1}^{N} w_{ij}\, e_j + \epsilon_i \qquad i = 1, \ldots, N \tag{1}$$

where $N$ is the number of examples in a training set, $e_j = Y_j - \overline{Y}$ is the prediction residual (where prediction is based on the average), $w_{ij}$ is the weight of the spatial proximity between the pair of examples $i$ and $j$, $\rho$ is a parameter that expresses the spatial dependence in the lattice and the error $\epsilon_i$ follows a normal distribution.

As recognized by Li et al. (2007), in order to informally assess the strength of the spatial dependence, exploratory data analysis should be based on estimating $\rho$ in the autoregressive model (see Equation 1). This means that the parameter $\rho$ plays a crucial role in representing autocorrelation in the data.

One common solution to estimate $\rho$ is to use a modified least squares estimator, which is the solution to the following quadratic equation in $\rho$,

$$\mathbf{e}^T(\mathbf{I} - \rho\mathbf{W})^T\mathbf{W}(\mathbf{I} - \rho\mathbf{W})\mathbf{e} = 0 \tag{2}$$

where $\mathbf{W}$ is the matrix representation of $w_{ij}$, $\mathbf{I}$ is the identity matrix and $\mathbf{e}$ is the vector of $e_i$ values. Although this estimator is consistent (Li et al., 2007), its computation is not straightforward, since it would require the computation of the Maximum Likelihood Estimator of $\rho$.

In a theoretical study, LeSage and Pace (2001) use the Maximum Likelihood Estimator to take into account autocorrelation in data. They stress that the presence of spatial dependence requires an appropriate treatment of spatial correlation effects. However, the computation of the Maximum Likelihood Estimator is impractical when large datasets need to be processed

(Li et al., 2007). Therefore, instead of computing an estimate of $\rho$, the Pearson's correlation coefficient and its variants, as well as entropy based measures, are commonly used in spatial data mining applications (LeSage and Pace, 2001).

For example, a spatial autocorrelation measure has been adopted by Zhang et al. (2003) to efficiently process similarity based range queries and joins and take autocorrelation into account when retrieving spatial time series. Scrucca (2005) has proposed a clustering procedure for identifying spatial clusters, based on the contiguity structure of objects and their attribute information. The procedure uses a K-means algorithm that incorporates the spatial structure of the data through the use of measures of spatial autocorrelation.

At present, several methods (such as Geographically Weighted Regression (Fotheringham et al., 2002), Kriging (Cressie, 1990) and Inverse Distance Weighting (Shepard, 1968)) which accommodate the phenomenon of spatial autocorrelation have been already developed in the area of spatial statistics. These methods are designed to consider spatial autocorrelation in two forms: spatial error where correlations across space exist in the error term, or spatial lag, where the dependent variable at a given position in space is affected by the independent variables at that position, as well as the variables (dependent or independent), at positions close to the given one (Anselin and Bera, 1998). In spatial data mining, there have been several attempts of treating spatial autocorrelation both in classical data mining (Rinzivillo and Turini, 2004; Pace and Barry, 1997; LeSage and Pace, 2001; Appice et al., 2010) and in relational data mining (Ceci and Appice, 2006; Malerba et al., 2005b). In any case, all these methods propose a way to account for spatial autocorrelation in a predictive modeling task (either regression or classification).

#### *2.1.1. Spatial Autocorrelation in Classification and Regression Tasks*

Zhao and Li (2011) have proposed a spatial entropy-based decision tree that differs from a conventional tree in the way that it considers the phenomenon of spatial autocorrelation in the classification process. A spatially-tailored formulation of the traditional entropy measure (i.e., "spatial entropy" (Li and Claramunt, 2006)) is used in the tree induction. This measure evaluates the dispersion of the entropy measure over some neighborhoods by looking for a split which minimizes the intra-distance computed for the examples in the majority class and maximizes the inter-distance between these and the examples in different classes.

A different formulation of spatially-aware entropy is provided by Rinzivillo and Turini (2004, 2007). They have proposed to compute the entropy for a spatial measurement of each example and to use the information gain based on such spatial entropy measure for the induction of spatial decision trees. In particular, the spatial entropy is computed for each weighted sum of the spatially related (e.g., overlapping) examples.

Bel et al. (2009) modify Breiman's classification trees (Breiman et al., 1984) to take into account the irregularity of sampling by weighting the data according to their spatial pattern (using Voronoi tessellations, a regular grid and kriging). Huang et al. (2004) propose and empirically validate methods based on logistic regression and Bayesian classification that explicitly take the spatial dimension into account.

For the regression task, a standard way to take into account spatial autocorrelation in spatial statistics is Geographically Weighted Regression (GWR) (Fotheringham et al., 2002). GWR assumes a local form of the regression surface depending on the site $(u, v)$. The linear regression model is extended by weighting each training observation in accordance with its proximity to point $(u, v)$, so that the weighting of an example is no longer constant in the calibration but varies with $(u, v)$. The local model for a response attribute $y$ at site $(u, v)$ takes the following form:

$$y(u,v) = \alpha_0(u,v) + \sum_k \alpha_k(u,v)x_k(u,v) + \epsilon_{(u,v)} \tag{3}$$

where $\alpha_k(u,v)$ is a realization of the continuous function $\alpha_k(u,v)$ at point $(u,v)$, while $\epsilon_{(u,v)}$ denotes random noise. Each coefficient $\alpha_k$ is locally estimated at location $(u,v)$ from measurements close to $(u,v)$.

$$\alpha(u,v) = (\mathbf{X}^T\mathbf{W}_{(u,v)}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}_{(u,v)}y \tag{4}$$

$\mathbf{W}_{(u,v)}$ is a diagonal weight matrix which states the influence of each observation for the estimation of $\alpha(u,v)$, defined by functions such as Gaussian or bi-square. $\mathbf{W}_{(u,v)}$ introduces the spatial lag depending on the position $(u,v)$ in the regression model. In this way, GWR takes advantage of positive autocorrelation between neighboring examples in space and provides valuable information on the nature of the processes being investigated.

Kriging (Bogorny et al., 2006) is another spatial regression technique which explicitly takes advantage of autocorrelation. It applies an optimal linear interpolation method to estimate unknown response values $y(u,v)$ at each site $(u,v)$. $y(u,v)$ is decomposed into three terms: a structural component, which represents a mean or constant trend, a random but spatially correlated component, and a random noise component, which expresses measurement errors or variation inherent to the attribute of interest.

Finally, the problem of dealing with autocorrelation in mining spatial data has been also addressed in multi-relational data mining. For example, Ceci and Appice (2006) propose a spatial associative classifier that learns, in the same learning phase, both spatially defined association rules and a classification model (on the basis of the extracted rules). Malerba et al. (2005a) present a multi-relational clustering algorithm (CORSO) that expresses the spatial structure of data by resorting to the First Order Logic representation formalism and uses learning in the Normal ILP setting in order to take into account for autocorrelation in the data. In this case, autocorrelation is implicit in spatial relations, such as overlap, to_right, to_left, between spatial objects to be clustered.

For the regression task, Malerba et al. (2005b) present a relational regression method (Mrs-SMOTI) that captures both global and local spatial effects of the predictive attributes, while building a regression model tightly integrated with a spatial database. The method considers the geometrical representation and relative positioning of the spatial objects to decide the split condition for the tree induction (e.g., towns crossed by any river and towns not crossed by any river). For the splitting decision a classical heuristic based on error reduction is used.

However, when resorting to multi-relational data mining, it is possible that the presence of autocorrelation in spatial phenomena can bias feature selection (Jensen and Neville, 2002). In particular, the distribution of scores for features formed from related objects with concentrated linkage (i.e., high concentration of objects linked to a common neighbor) has a surprisingly large variance when the class attribute has high autocorrelation. This large variance causes feature selection algorithms to be biased in favor of these features, even when they are not related to the class attribute, that is, they are randomly generated. In this case, conventional hypothesis tests, such as the $\chi^2$-test for independence, which evaluate statistically significant differences between proportions for two or more groups in a dataset, fail to discard uninformative features.

### 2.2. Building Predictive Clustering Trees

Predictive Clustering Trees (Blockeel et al., 1998) view a decision tree as a hierarchy of clusters: the top-node corresponds to one cluster containing all data, which is recursively partitioned

into smaller clusters while moving down the tree. The key properties of PCTs are that *i)* they can be used to predict many properties of an example at once (multi-target prediction), *ii)* they can be applied to a wide range of prediction tasks (classification and regression), and *iii)* they can handle examples represented by means of a complex representation (Džeroski et al., 2007), which is achieved by plugging in a suitable distance for the task at hand. The task of learning predictive clustering trees can be formalized as follows:
*Given*

- a descriptive space $\mathbf{X} = \{X_1, X_2, \ldots X_m\}$ spanned by $m$ independent (or predictor) variables $X_j$,
- a target variable $Y$, that can be either discrete or continuous,
- a set $T$ of training examples, $(x_i, y_i)$ with $x_i \in \mathbf{X}$ and $y_i \in Y$

*Find* a tree structure $\tau$ which represents:

- A set of hierarchically organized clusters on $T$ such that for each $u \in T$, a sequence of clusters $C_{i_0}, C_{i_1}, \ldots, C_{i_r}$ exists for which $u \in C_{i_r}$ and the containment relation $T = C_{i_0} \supseteq C_{i_1} \supseteq \ldots \supseteq C_{i_r}$ is satisfied. Clusters $C_{i_0}, C_{i_1}, \ldots, C_{i_r}$ are associated to the nodes $t_{i_0}, t_{i_1}, \ldots, t_{i_r}$, respectively, where each $t_{i_j} \in \tau$ is a direct child of $t_{i_{j-1}} \in \tau$ $(j = 1, \ldots, r)$ and $t_{i_0}$ is the root of the structure $\tau$.
- A predictive piecewise function $f : \mathbf{X} \to Y$, defined according to the hierarchically organized clusters. In particular,

$$\forall u \in \mathbf{X},\ f(u) = \sum_{t_i \in leaves(\tau)} D(u, t_i) f_{t_i}(u) \tag{5}$$

  where $D(u, t_i) = \begin{cases} 1 & \text{if } u \in C_i \\ 0 & \text{otherwise} \end{cases}$ and $f_{t_i}(u)$ is a (discrete or continuous valued) prediction function associated to the leaf $t_i$.

Note that this general formulation of the problem allows us to take into account a clustering phase that can consider the autocorrelation effect due to the spatial nature of the data. Moreover, it works for both classification and regression tasks.

The induction of PCTs performed by the system CLUS (Blockeel et al., 1998), is not very different from that of standard decision trees (for example, see the C4.5 algorithm (Quinlan, 1993)): At each internal node $t$, a test has to be selected according to a given evaluation function. The main difference is that PCTs select the best test by maximizing the (inter-cluster) variance reduction, defined as:

$$\Delta_Y(E, \mathcal{P}) = Var(E) - \sum_{E_k \in \mathcal{P}} \frac{|E_k|}{|E|} Var(E_k) \tag{6}$$

where $E$ represents the cluster of training examples falling in $t$ and $\mathcal{P}$ defines the partition $\{E_1, E_2\}$ of $E$. The partition is defined according to a Boolean test on a predictor variable in $\mathbf{X}$. By maximizing variance reduction, cluster homogeneity is maximized, improving at the same time the predictive performance.

If the variance $Var(\cdot)$ and the predictive functions $f(\cdot)$ are considered as parameters, instantiated for the specific learning task at hand, it is possible to easily adapt PCTs to different domains and different tasks. To construct a regression tree, for example, the variance function $Var(\cdot)$ returns the variance of the target variable of the examples in the partition $E$ (i.e., $Var(E) = Var(Y)$), whereas the predictive function is the average of the response values in a cluster. To construct a classification tree, on the other hand, the variance function $Var(\cdot)$ returns the Gini index of the target variable of the examples in the partition $E$ (i.e., $Var(E) = 1 - \sum_{y \in Y} p(y, E)^2$, where $p(y, E)$ is the probability that an instance in $E$ belongs to the class $y$), whereas the predictive function is the majority class for the target variable. Indeed, by appropriately defining the variance and predictive functions, PCTs have been used for clustering (Blockeel et al., 1998), classification and regression (Blockeel et al., 1998; Demšar et al., 2005), and time series data analysis (Džeroski et al., 2007).

In this paper, we extend the approach of constructing PCTs by introducing an adequate variance measure that takes into account the spatial dimension of the examples in addition to the descriptive and target spaces. This is achieved by explicitly considering the spatial autocorrelation.

## 3. Learning PCTs by taking Spatial Autocorrelation into account

In order to formalize the learning task we are referring to, we need to define the spatial dimension of the data with the goal of explicitly taking spatial autocorrelation into account. For this purpose, in addition to the descriptive space **X** (which, does not represent spatial relationships, although, in general, it can include spatially-aware attributes, such as distance to the nearest object) and the target space **Y**, it is necessary to add information on the spatial structure of the data in order to be able to capture the spatial arrangement of the objects (e.g., the coordinates of the spatial objects involved in the analysis or the pairwise distances between them) [1].

Moreover, we have to consider different aspects:

***i)*** What attributes should be considered in the tests in the internal nodes of the tree?

***ii)*** Which evaluation measure for the tests, taking spatial dimension into account, would lead to the best clustering?

***iii)*** Which distance measure should be used when taking into account spatial autocorrelation?

Concerning *i)*, a naïve solution would consider both the descriptive and the spatial attributes as candidates in a test associated to the split. However, this solution would lead to models that would be difficult to apply in the same domain, but in different spatial contexts. For this reason, following Ester et al. (1997), we do not consider spatial information in the candidate tests. This limitation of the search space allows us to have more general models, at the price of possible loss in predictive power of the induced models.

Concerning *ii)*, CLUS uses variance reduction as an evaluation measure. However, in order to take spatial autocorrelation into account when partitioning the descriptive space, a different

[1] Coherently with other works that use spatial autocorrelation in predictive data mining (LeSage and Pace, 2001), we only consider autocorrelation on response variable. However, spatial autocorrelation can also be taken into account on the descriptive space, but it requires a more sophisticated setting (e.g., multi-relational setting) that we plan to explore in future work.

measure is necessary. In spatial analysis, several spatial autocorrelation statistics have been defined. The most common ones are Global Moran's $I$ (Moran, 1950) and Global Geary's $C$ (Legendre, 1993). These statistics require a spatial weights matrix that reflects the intensity of the spatial relationship between examples in a neighborhood.

Formula (7) defines the Global Moran's $I$ on the response variable $Y$ as follows:

$$I_Y = \frac{N \sum_i \sum_j w_{ij}(y_i - \overline{Y})(y_j - \overline{Y})}{W \sum_i (y_i - \overline{Y})^2} \tag{7}$$

where $N$ is the number of spatial objects (examples) indexed by $i$ and $j$; $Y$ is the variable of interest; $y_i$ and $y_j$ are the values of the variable $Y$ for the objects $o_i$ and $o_j$, respectively; $\overline{Y}$ is the overall mean of $Y$; and $W = \sum_{i,j} w_{ij}$ is the sum of spatial weights $w_{ij}, i, j = 1, \ldots, N$. Dubin (1998) shows that under randomization or normality the expected value of Moran's $I$ is

$$E(I_Y) = \frac{-1}{N-1} \tag{8}$$

where high absolute values of $I$ indicate high autocorrelation in the data. Positive values indicate positive autocorrelation, while negative values indicate negative autocorrelation. The values of Moran's $I$ generally range from -1 to +1, where 0 indicates a random distribution of the data.

Note that the Moran's $I$ was originally defined to measure autocorrelation of a numeric variable. In our case, we intend to also address the classification task, in addition to the regression task. Let $Y$ be a discrete variable which admits $q$ distinct values in its domain. To compute the Moran's index for $Y$ we resort to a one-versus-all solution. In particular, we transform the discrete variable $Y$ into $q$ $0/1$ binary variables $Y_1, Y_2, \ldots, Y_q$. Then we compute $I_Y$ as the average of Moran's $I$ computed for each $Y_j$ ($j = 1, 2, \ldots, q$), that is:

$$I_Y = \frac{1}{q} \sum_{j=1}^{q} I_{Y_j} \tag{9}$$

An alternative measure of spatial autocorrelation is the Global Geary's $C$, defined as:

$$C_Y = \frac{(N-1) \sum_i \sum_j w_{ij}(y_i - y_j)^2}{2W \sum_i (y_i - \overline{Y})^2} \tag{10}$$

Its values typically range from 0 (positive autocorrelation) to 2 (negative autocorrelation), where 1 indicates a random distribution of the data. Also in this case, we deal with the discrete response variable on a classification task by resorting to the same strategy presented above for the Moran's $I$ statistic.

While both statistics reflect the spatial dependence of values, they do not provide identical information: $C$ emphasizes the differences in values between pairs of observations, while $I$ emphasizes the covariance between the pairs. This means that Moran's $I$ is smoother, whereas Geary's $C$ is more sensitive to differences in small neighborhoods.

Concerning *iii)*, the weights $w_{ij}$ used in equations (7) and (10) are defined as a function of the spatial distance measure. The basic idea is that the examples close to a specific example have more influence in the estimation of its response value than examples farther away. A popular choice is to use the Gaussian-like similarity measure:

$$w_{ij} = \begin{cases} e^{-\frac{d_{ij}^2}{b^2}} & \text{if } d_{ij} < b \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where $b$ is referred to as the bandwidth and $d_{ij}$ is the Euclidean spatial distance between examples $o_i$ and $o_j$. If $o_i$ and $o_j$ are at the same location, $w_{ij} = 1$. The weighting of other data will decrease according to a Gaussian-like curve, as the distance between $o_i$ and $o_j$ increases. If $o_i$ and $o_j$ are farther away from each other than the bandwidth $b$, then they are not considered in the analysis. We refer to this weighting function as "Gaussian".

As an alternative, it is possible to use a discrete weighting function (see Equation (12)) and a bisquare density function (see Equation (13)):

$$w_{ij} = \begin{cases} 1 - \frac{d_{ij}}{b} & \text{if } d_{ij} < b \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$w_{ij} = \begin{cases} (1 - \frac{d_{ij}^2}{b^2})^2 & \text{if } d_{ij} < b \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

which we refer to as "Euclidean" and "Modified", respectively.

Whatever weighting schema is employed, the choice of the bandwidth $b$ plays a crucial role. This means that the main problem is how to select the optimal bandwidth $b$. This problem is described and tackled in Section 3.2.

### *3.1. The Algorithm*

We can now proceed to describe the top-down induction algorithm for building Spatial PCTs (Algorithm 1). It is a recursive algorithm which takes as input a set of spatial training examples $E$ and partitions the descriptive space until a stopping criterion is satisfied (Algorithm 1, line 2). Since the implementation of this algorithm is based on the implementation of the CLUS algorithm, we will call this algorithm SCLUS (for Spatial CLUS).

The main loop (Algorithm 1, lines 6-10) searches for the best attribute-value test $v^*$ that can be associated to a node $t$. It associates the best test $v^*$ with the internal node $t$ and calls itself recursively to construct a subtree for each subset (cluster) in the partition $P^*$ induced by $v^*$ on the training example set.

As discussed above concerning *i)*, splits are derived only from the space $\mathbf{X}$ of the predictor variables. Possible tests are of the form $X \leq \beta$ for continuous variables, and $X \in \{x_{i_1}, x_{i_2}, \ldots, x_{i_o}\}$ (where $\{x_{i_1}, x_{i_2}, \ldots, x_{i_o}\}$ is a subset of the domain $D_X$ of $X$) for discrete variables. For continuous variables, possible values of $\beta$ are found by sorting the distinct values of $X$ in the training set associated with $t$, then considering a threshold between each pair of adjacent values. Therefore, if the cases in $t$ have $d$ distinct values for $X$, at most $d - 1$ thresholds are considered. When selecting a subset of values for a discrete variable, we rely on a non-optimal greedy strategy (Mehta et al., 1996). It starts with an empty set $Left_t = \oslash$ and a full set $Right_t = D_X$, where $D_X$ is the domain of $X$. It moves one element from $Right_t$ to $Left_t$, such that the move results in an increased heuristic $h$. It is noteworthy that this greedy strategy to explore the space of candidate

**Algorithm 1** Top-down induction of SpatialPCTs.

```
1:  procedure SpatialPCT($E$) returns tree
2:  if stop(E) then
3:      return leaf(Prototype($E$))
4:  else
5:      $(v^*, h^*, \mathcal{P}^*) = (null, 0, \emptyset)$
6:      for each Boolean test $v$ do
7:          $\mathcal{P}$ = partition induced by $v$ on $E$
8:          $h = \alpha \cdot \widehat{\Delta_Y}(E, \mathcal{P}) + (1 - \alpha) \cdot S_Y(E, \mathcal{P})$
9:          if $(h > h^*)$ then
10:             $(v^*, h^*, \mathcal{P}^*) = (v, h, \mathcal{P})$
11:     for each $E_k \in \mathcal{P}^*$ do
12:         $tree_k$ = SpatialPCT($E_k$)
13:     return node($v^*$, $\bigcup_k \{tree_k\}$)
```

splits on a discrete variable does not require the definition of an apriori ordering on the possible values of $D_X$ as used in traditional decision tree induction (Breiman et al., 1984).

The algorithm evaluates the best split according to the formula (14) reported in Algorithm 1, line 8.

$$h = \alpha \cdot \widehat{\Delta_Y}(E, \mathcal{P}) + (1 - \alpha) \cdot S_Y(E, \mathcal{P}) \tag{14}$$

This formula is a linear combination of the scaled variance reduction $\widehat{\Delta_Y}(E, \mathcal{P})$ (see Equation (19)) and autocorrelation measure $S_Y(E, \mathcal{P})$ (see Equation (17)). Both, variance and autocorrelation are computed for the $Y$ variable (the class). In the case of multiple target variables, the average values of both, variance reduction $\Delta_Y(E, \mathcal{P})$ and autocorrelation $S_Y(E, \mathcal{P})$ are taken over the set of target variables, where each target variable contributes equally to the overall $h$ value.

The influence of these two parts of the linear combination when building the PCTs is determined by a user-defined coefficient $\alpha$ that falls in the interval [0, 1]. When $\alpha = 0$, SCLUS uses only spatial autocorrelation, when $\alpha = 0.5$ it weights equally variance reduction and spatial autocorrelation, while when $\alpha = 1$ it works as the original CLUS algorithm. If autocorrelation is present, examples with high spatial autocorrelation (close to each other in space) will fall in the same cluster and will have similar values of the response variable. In this way, we are able to keep together spatially close examples without forcing spatial splits (which can result in losing generality of the induced models).

According to the discussion above of *ii)*, $S_Y(E, \mathcal{P})$ can be defined in terms of either the Moran's $I$ or the Geary's $C$. However, since $I_Y$ and $C_Y$ range in different intervals (even though they are consistently monotonic), it is necessary to appropriately scale them. Since variance reduction is non-negative, we decided to scale both in the interval [0, 1], where 1 means high positive autocorrelation and 0 means high negative autocorrelation.

For Moran's $I$, $S_Y(E, \mathcal{P})$ is:

$$S_Y(E, \mathcal{P}) = \frac{1}{|E|} \cdot \sum_{E_k \in \mathcal{P}} |E_k| \cdot \widehat{I_Y}(E_k) \tag{15}$$

where $\widehat{I_Y}(E_k)$ is the scaled Moran's $I$ computed on $E_k$, i.e.,

$$\widehat{I_Y}(E_k) = \frac{I_Y(E_k) + 1}{2} \tag{16}$$

This scales Moran's $I$ from the interval [-1, 1] to [0, 1].

For Geary's $C$ , $S_Y(E, \mathcal{P})$ is:

$$S_Y(E, \mathcal{P}) = \frac{1}{|E|} \cdot \sum_{E_k \in \mathcal{P}} |E_k| \cdot \widehat{C_Y}(E_k) \tag{17}$$

where $\widehat{C_Y}(E_k)$ is the scaled Geary's $C$ computed on $E_k$, i.e.,

$$\widehat{C_Y}(E_k) = \frac{2 - C_Y(E_k)}{2} \tag{18}$$

This scales Geary's $C$ from the interval [0, 2] to [0, 1].

The choice of the scaling interval does not affect the heuristic computation, therefore other scaling intervals are possible as well, provided that, in all cases, the same scaling is performed and the monotonicity of the scaled measure is maintained.

Moreover, in order to guarantee a fair combination of the variance reduction and the autocorrelation statistic $S_Y(\mathcal{P}, E)$, we also need to scale the variance reduction to the interval [0, 1]. For that purpose, we use a common scaling function:

$$\widehat{\Delta_Y}(E, \mathcal{P}) = \frac{\Delta_Y(E, \mathcal{P}) - \Delta min}{\Delta max - \Delta min} \tag{19}$$

where $\Delta max$ and $\Delta min$ are the maximum and the minimum values of $\Delta_Y(E, \mathcal{P})$ over the possible splits.

The search stops when the number of examples falling in a leaf is smaller than $\sqrt{N}$, which is considered as a good locality threshold that does not permit to lose too much in accuracy also for rule based classifiers (Gora and Wojna, 2002). A further stopping criterion is based on the exact Fisher test (F-test) that is performed to check whether a given split /test in an internal node of the tree results in a reduction in $h$ that is statistically significant at a given significance level. In order to estimate the optimal significance level among the values in the set $\{1, 0.125, 0.1, 0.05, 0.01, 0.005, 0.001\}$, we optimize the MSE obtained with an internal 3-fold cross validation. When one of the stopping criterion is satisfied, the algorithm creates a leaf and labels it with a predictive function (in the regression case, the average; and in the classification case, the mode of the class values) defined for the examples falling in that leaf (see lines 2-3 of Algorithm 1).

In SCLUS, the pruning is the pessimistic error pruning strategy which is also implemented in several regression/ model tree learners (including M5' and CLUS). This means that a subtree is added only if the error committed at the leaf is greater than the errors commented by the subtree multiplied by a scaling factor (Wang and Witten, 1997). The results that we present in this paper are those of the pruned tree models learned by SCLUS.

### *3.2. Choosing the Bandwidth*

The choice of the bandwidth (denoted by $b$ in Equation (11)) is perhaps the most critical decision to be taken in the modeling process. This parameter controls the degree of smoothing. A small bandwidth results in a very rapid distance decay, whereas a larger value results in a smoother weighting scheme. At the same time, this parameter influences the calculation of autocorrelation.

The bandwidth may be defined manually or by using some adaptive method on the whole training space, such as cross validation and the corrected Akaike Information Criterion (AIC) used in GWR (Fotheringham et al., 2002). A wrapper solution would significantly increase (by a logarithmic factor, in the worst case) the complexity of the algorithm. In this study, for the selection of the bandwidth, we minimize the leave-one-out cross validated - Root Mean Square Error (RMSE). Moreover, in this automatic determination of the bandwidth, the selection is not performed directly on the bandwidth $b$, but on $b^{\%}$, that is, the bandwidth expressed as a percentage of the maximum distance between two examples. This means that the algorithm implicitly considers different bandwidth values $b$ at different nodes of the tree depending on the maximum distance between connected examples falling in that node of the tree. The bandwidth $b^{\%}$ ranges in the interval $[0, 100\%]$.

Minimization is performed by means of the Golden section search (Brent, 1973) that recursively partitions the $b\%$ domain. Golden section search is similar to binary search, improving it by splitting the range in two intervals with a length ratio of $\gamma$ instead of 1 (equal parts). *Golden ratio* has the value $\gamma = \frac{1+\sqrt{5}}{2}$.

The share maintains a pair of minimum and maximum bandwidth values, $b_1^{\%}$ and $b_2^{\%}$ (at the first iteration, they are initialized as the minimum and maximum bandwidth in the interval $[0, 100\%]$). At each step, the algorithm identifies a point $b_3^{\%}$ between them, according to the golden ratio and computes the cross-validated error for that point ($error_{b_3^{\%}}$). The values of the function at these points are $f(b_1^{\%})$, $f(b_3^{\%})$, and $f(b_2^{\%})$ and, collectively, these are known as a "triplet". The algorithm than identifies the only parabola with a vertical axis that intersects the points $\{(b_1^{\%}, error_{b_1^{\%}}), (b_3^{\%}, error_{b_3^{\%}}), (b_2^{\%}, error_{b_2^{\%}})\}$. On the basis of the position of the minimum of this parabola, the system decides whether to consider $(b_1^{\%}, b_3^{\%})$ or $(b_3^{\%}, b_2^{\%})$ as the next pair of (minimum and maximum) $b^{\%}$ values.

The search stops when there is no cross-validated error reduction. In the regression case, error is the RMSE computed by fitting a weighted linear model for the example left out. In the case of classification, similarly to the regression case, the error is the RMSE on the example left out, but it is computed as the average of RMSEs obtained by fitting a weighted linear model for each binary target variable obtained after the binarization of the discrete response variable. Weights are defined according to (11).

### *3.3. Time Complexity*

The computational complexity of the algorithm depends on the computational complexity of adding a splitting node $t$ to the tree, which in fact depends on the complexity of selecting a splitting test for $t$. A splitting test can be either continuous or discrete. In the former case, a threshold $\beta$ has to be selected for a continuous variable. Let $N$ be the number of examples in the training set $E$; Then the number of distinct thresholds can be $N$-1 at worst. They can be determined after sorting the set of distinct values. If $m$ is the number of predictor variables, the determination of all possible thresholds has a complexity $O(m * N * logN)$ when an optimal algorithm is used for sorting.

For each variable, the system has to compute the evaluation measure $h$ for all possible thresholds. This computation has, in principle, time-complexity $O((N-1) * (N + N^2))$; where $N-1$ is the number of thresholds, $O(N)$ is the complexity of the computation of the variance reduction $\Delta_Y(E, \mathcal{P})$ and $O(N^2)$ is the complexity of the computation of autocorrelation $S_Y(E, \mathcal{P})$. However, for each threshold, it is not necessary to recompute values from scratch since partial sums in both variance reduction computation [2] and in autocorrelation computation can be used. In particular, partial sums can be incrementally updated depending on the examples that are iteratively moved from the right to the left branch. This optimization makes the complexity of the evaluation of the splits for each variable $O(N^2)$. This means that the worst case complexity of creating a splitting node on a continuous attribute, in the case of a continuous target variable is $O(m * (NlogN + N^2))$ and in the case of a discrete target variable is $O(m * (NlogN + q * N^2))$, where $q$ is the number of classes.

Similarly, for a discrete splitting test (for each variable), the worst case complexity, in the case of a continuous target variable, is $O((d - 1) * (N + N^2))$, where $d$ is the maximum number of distinct values of a discrete descriptive variable ($d \leq N$) and in the case of a discrete target variable is $O((d - 1) * (N + q * N^2))$. This complexity takes the same optimization proposed for continuous splits into account.

Therefore, finding the best splitting node (either continuous or discrete), in the case of continuous target variable, has a complexity of $O(m*(NlogN+N^2))+O(m*(d-1)*(N+N^2))$, that is $O(m*N*(logN+d*N))$, where $m$ is the number of descriptive variables, $N$ is the number of examples and $d$ is the maximum number of distinct values of a discrete variable. Analogously, finding the best splitting node (either continuous or discrete), in the case of discrete target variable, has a complexity of $O(m * N * (logN + q * d * N))$.

## 4. Materials and Methods

In this section, we first provide a description of the datasets used in this study. We then give a short description of the experimental methodology used in the evaluation.

### *4.1. Datasets*

In this experimental evaluation, we use real world data that includes a spatial component. We consider nine datasets for the regression task and four datasets for the classification task. The regression datasets are FF, NWE, SIGMEA_MS and SIGMEA_MF, Foixa, GASD, River, Kenya and Malawi. The classification datasets are Foixa_01, Foixa_045, Prim and Kras.

The **Forest Fires** (FF) (Cortez and Morais, 2007) dataset is publicly available for research purposes from the UCI Machine Learning Repository [3]. It contains 517 forest fire observations from the Montesinho park in Portugal. The data, collected from January 2000 to December 2003, includes the coordinates of the forest fire sites, the burned area of the forest given in *ha* (response variable), the Fine Fuel Moisture Code (FFMC), the Duff Moisture Code (DMC), the Drought Code (DC), the Initial Spread Index (ISI), the temperature in degrees Celsius, the relative humidity, the wind speed in $km/h$ and the outside rain in *mm* within the Montesinho park map.

The **NWE** (North-West England)[4] dataset contains census data concerning North West England. The data include the percentage of mortality (target variable) and measures of deprivation level in the ward, including index scores such as the Jarman Underprivileged Area Score,

[2] both in terms of variance and Gini index

[3] `http://archive.ics.uci.edu/ml/`

[4] http://www.ais.fraunhofer.de/KD/SPIN/project.html

Townsend score, Carstairs score and the Department of the Environment Index. The spatial coordinates of the ward centroid are given as well. The spatial unit of analysis is a ward i.e., a sub-area of a region.

The **SIGMEA_MS** and **SIGMEA_MF** (MS and MF) (Demšar et al., 2005) datasets are derived from one multi-target regression dataset containing measurements of pollen dispersal (crossover) rates from two lines of plants (target variables), that is, the transgenic male-fertile (MF) and the non-transgenic male-sterile (MS) line of oilseed rape. The predictor variables are the cardinal direction and distance of the sampling point from the center of the donor field, the visual angle between the sampling plot and the donor field, and the shortest distance between the plot and the nearest edge of the donor field, as well as the coordinates of the sampling point.

The **Foixa** dataset (Debeljak et al., 2012) contains measurements of the rates of outcrossing (target variable) at sampling points located within a conventional field that comes from the surrounding genetically modified (GM) fields within a 400 ha maize-oriented production area in the Foixa region in Spain. The independent variables include the number and size of the surrounding GM fields, the ratio of the size of the surrounding GM fields and the size of conventional field, the average distance between conventional and GM fields, as well as the coordinates of the sampling points.

The **GASD** dataset (USA Geographical Analysis Spatial Dataset) (Pace and Barry, 1997) contains observations on US county votes cast in the 1980 presidential election. Specifically, it contains the total number of votes cast per county (target variable), the population above 18 years of age in each county, the number of owner-occupied housing units, the aggregate income and the coordinates of the centroid of the county.

The **River** (Ohashi et al., 2010; Macchia et al., 2011) dataset contains water information of the Portuguese rivers Douro and Paiva in 2009. This dataset includes measurements of the pH level, conductivity, turbidity and common bacteria like *Escheria Coli* and *Coliformi Bacteria* taken at control points along the rivers. In addition, the navigation distance between the control points is also available. The goal is to predict river pollution and the pH level is considered as the target variable, since it is recognized to be a good indicator of river pollution.

The **Kenya** and **Malawi** datasets contain observations of the Headcount poverty index (target variable) and other data found in the poverty mapping reports for the countries Kenya and Malawi [5]. Specifically, they contain the total number and density of poor people, the average number of years of schooling of adults, the number of active community groups, the accessibility of resources like water, roads and electricity, the average resource consumption, and the coordinates of the spatial unit (administrative level).

The **Foixa_01** and **Foixa_045** (Debeljak et al., 2012) classification datasets are derived from the **Foixa** dataset through a discretization proposed by a domain-expert. Discretization is performed on the target variable (outcrossing rate) and is based on the thresholds 0.1% and 0.45%. These thresholds correspond to the farmers' internal standard that keeps them on the safe side of maize production (0.45%) and the requirements of the starch industry for the purity of maize at the entrance to the production process (0.1%). In particular, in Foixa_01 dataset, the class *low* refers to fields with outcrossing rate in the interval [0, 0.1%] whereas, the class *high* refers to fields with outcrossing rate in the interval [0.1%, 100%]. Similarly, in Foixa_045 dataset, the class *low* refers to fields with outcrossing rate in the interval [0, 0.45%] whereas, the class *high* refers to fields with outcrossing rate in the interval [0.45%, 100%].

[5] `http://sedac.ciesin.columbia.edu/povmap/ds_info.jsp`

Table 1: Descriptions of the datasets used in the evaluation. For each dataset, we give the Task, $N$ – number of examples, *#Attr.* – number of descriptive attributes and $b$ – automatically chosen bandwidth values.

| Dataset | Task | $N$ | *#Attr.* | auto.chosen $b^{\%}$ |
|---|---|---|---|---|
| FF | Regression | 517 | 12 | 100% |
| NWE | Regression | 970 | 4 | 7.7% |
| MS | Regression | 817 | 4 | 4.8% |
| MF | Regression | 817 | 4 | 9.1% |
| Foixa | Regression | 420 | 7 | 64.6% |
| GASD | Regression | 3106 | 4 | 2.5% |
| River | Regression | 32 | 8 | 100% |
| Kenya | Regression | 121 | 9 | 63.8% |
| Malawi | Regression | 3004 | 10 | 8.1% |
| Kras | Classification | 1439 | 158 | 100% |
| Prim | Classification | 2024 | 104 | 100% |
| Foixa_01 | Classification | 420 | 7 | 100% |
| Foixa_045 | Classification | 420 | 7 | 100% |

The **Kras** and **Prim** (Stojanova et al., 2012) classification datasets contain data on fire outbreaks (target variable) in the Kras and Primorska region in Slovenia. The data consists of GIS data (spatial coordinates, altitude, forest coverage, percentage of agricultural areas, percentage of urban areas, distance from roads, highways, railways, cities, etc.), multi-temporal MODIS data (average temperature and average net primary production), weather forecast data from ALADIN (temperature, humidity, solar energy, evaporation, speed and direction of the wind, transpiration, etc.). For the Kras dataset, we additionally have vegetation height and vegetation structure data obtained from LIDAR and LANDSAT images to which a predictive models learned from LIDAR data and LANDSAT images was applied. Together with the **Foixa_01** and **Foixa_045** datasets described above, we consider four classification datasets.

A description of the datasets in terms of general properties is provided in Table 1, where we also report the automatically chosen bandwidth as described in Section 3.2. Spatial autocorrelation of the MF dataset is illustrated in Figure 4 (a).

### *4.2. Experimental Setup*

The experiments are performed on an Intel Xeon CPU @2.00GHz server running the Linux Operating System. For each dataset, we evaluate the effectiveness of SCLUS in terms of accuracy, model complexity, learning time as well as quality of extracted clusters. All of these performance measures are estimated by using 10-fold cross validation. In particular, the accuracy is measured in terms of the Root Mean Squared Error (RMSE) for regression tasks and Precision and Recall for classification tasks, while the model complexity is measured in terms of the number of leaves in the learned trees. The computation time is measured in seconds. The quality of the clusters is measured in terms of their *spatial dispersion*. More formally, let $C = \{C_1, \ldots, C_s\}$ be the set of clusters associated with the leaves of a PCT, similar to what is suggested in Sampson and Guttorp (1992), we compute the *spatial dispersion* as the average intra-cluster distance, that is:

$$SD = avg_{\ C_k \in C} \left( \sum_{o_i, o_j \in C_k} \frac{euclideanDist(o_i, o_j)}{(N_k^2)} \right) \tag{20}$$

where $N_k$ is the number of examples in $C_k$ and $euclideanDist(o_i, o_j)$ is the spatial distance between the objects $o_i$ and $o_j$. According to our assumptions, the smaller the value of $SD$, the better the clustering.

We first evaluate the performance of SCLUS using different bandwidth values (b=1%, 5%, 10%, 20%) and weighting functions (Euclidian (Euc.), Modified Euclidian (Mod.) and Gaussian (Gauss.)) in order to understand their impact on the accuracy of the models. Then, SCLUS is run with automatic bandwidth determination, with the two different spatial autocorrelation measures, Global Moran's $I$ (SCLUS_Moran) and Global Geary's $C$ (SCLUS_Geary), and with $\alpha \in \{0, 0.5\}$. These different configurations of SCLUS are considered as a part of its definition and depend on the nature of the modeling problems considered. SCLUS is compared with the original CLUS (Blockeel et al., 1998) algorithm. We also compare SCLUS to a modification of CLUS, where the response variable set is extended with the spatial coordinates as additional response variables. This is done only for the computation of the split evaluation measure. In this way, we are able to implicitly take autocorrelation into account. Henceforth, we refer to this configuration of CLUS as CLUS*. For the regression task, SCLUS is compared to other competitive regression algorithms i.e., M5' Regression Trees (RT) and Support Vector Regression (SVR) (implemented in the WEKA software suite (Witten and Frank, 2005)) and Geographically Weighted Regression (GWR) (Fotheringham et al., 2002). Only GWR, SCLUS and CLUS* take autocorrelation into account.

Besides the *quantitative* analysis of results, we also provide an extended *qualitative* analysis of extracted models. The latter provides a clear idea of the differences among the models extracted by CLUS, GWR and SCLUS from the ecological datasets considered. This qualitative analysis is performed both in terms of the structure of extracted models and in terms of visual differences in the predictions they return.

## 5. Results and Discussion

In this Section, we present an empirical evaluation of the system SCLUS that implements the method SpatialPCTs presented in Section 3. First, we investigate the performance of the system along the dimensions of the weighting functions and autocorrelation measures used in the evaluation of splits, as well as the sensitivity of SCLUS to the choice of the bandwidth $b$ and to the setting of $\alpha$. Second, we evaluate the system for automatic determination of the bandwidth, presented in Section 3.2. Third, we compare the performance of SCLUS to the baseline performance of CLUS, as well as to the performance of the competitive (spatial) regression and classification methods on real world datasets. Finally, we give a qualitative analysis of the results on several ecological datasets (MS, MF and FOIXA), both in terms of the structure of the learned models and in terms of the visual differences in their predictions.

### *5.1. Regression tasks*

Table 2 shows the effect of the bandwidth and of the weighting function as used within the autocorrelation part of the splitting criterion. The bandwidth (b=1%, 5%, 10%, 20%) is given as a percentage of the maximum spatial distance between any two examples in space. Specifically, in Table 2, we report the RMSE results of the SCLUS_Moran models learned with different weighting functions (Euclidian (Euc.), Modified Euclidian (Mod.) and Gaussian (Gauss.)), evaluation measures (SCLUS_Moran and SCLUS_Geary), as estimated by 10-fold CV. For comparison, we consider only spatial autocorrelation and ignore the variance reduction in the splitting criterion ($\alpha = 0$).

While the selection of the bandwidth very much influences the results, there is no larger difference in performance when using different weighting functions (see equations (11), (12) and (13)). Experiments show that manual (non-automatic) tuning of the bandwidth is difficult, since there is no unique bandwidth value which exhibits the best predictive capabilities for each dataset/weighting schema. On the other hand, the mechanism we propose for the automatic choice of bandwidth seems to be effective, since it generally leads to higher accuracy.

In Table 3, we report the RMSE results, as estimated by 10-fold CV, for SCLUS (with Global Moran's I and Global Geary's C as measures of spatial autocorrelation, with an automatically chosen bandwidth and with $\alpha \in \{0, 0.5\}$). We also report the results for CLUS*, GWR, SVR and M5' Regression Trees. The implementation of CLUS* supports only datasets that contain coordinates and not relative distances between the examples in the dataset which is the case with the River dataset. Note that SCLUS does not have such implementation problems, whereas CLUS does not use the spatial information at all. M5' Regression Trees and SVR do not consider spatial autocorrelation while GWR incorporates it and accounts for non-stationarity.

The results show that the difference in the performance of SCLUS with Global Moran's *I* and Global Geary's *C* as measures of spatial autocorrelation is not large. This is not surprising since both measures evaluate the strength of spatial autocorrelation by emphasizing the covariance (differences in values) between pairs of observations. On the other hand, the selection of the user-defined parameter $\alpha$ is a very important step with a strong external influence on the learning process since it is not dependent on the properties of the data, as in the case of the bandwidth and the measures of spatial autocorrelation. The simplest solution is to set this parameter to 0 (consider only the spatial statistics) or 1 (consider only the variance reduction for regression, as in the original CLUS algorithm). Any other solution will combine the effects, allowing both criteria to influence the split selection.

Table 2: The RMSEs (estimated by 10-fold CV) of the SCLUS_Moran models learned with different weighting functions, evaluation measures, bandwidth values and $\alpha$=0. The best results for each bandwidth value are given in bold.

| Dataset | 1% | | | 5% | | | 10% | | | 20% | | | auto. chosen b (%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mod. | Gauss. | Euc. | Mod. | Gauss. | Euc. | Mod. | Gauss. | Euc. | Mod. | Gauss. | Euc. | Mod. | Gauss. | Euc. |
| FF | 47.22 | 47.22 | 47.22 | 47.22 | 47.22 | 47.22 | 47.22 | 47.22 | 47.22 | 47.22 | 47.22 | 47.22 | **42.82** | 47.21 | 47.21 |
| NWE | 2.48 | 2.48 | 2.48 | 2.48 | 2.48 | 2.48 | 2.46 | 2.46 | 2.46 | 2.45 | 2.45 | 2.45 | **2.16** | 2.36 | 2.39 |
| Foixa | 2.58 | **2.57** | 2.58 | 2.58 | 2.64 | **2.55** | 2.58 | **2.55** | 2.56 | 2.57 | 2.59 | **2.54** | 2.53 | 2.55 | **2.41** |
| GASD | **0.17** | **0.17** | 0.18 | **0.17** | **0.17** | 0.19 | **0.17** | 0.19 | 0.19 | 0.19 | 0.19 | 0.19 | 0.18 | 0.18 | **0.17** |
| MF | 2.46 | **2.19** | 2.47 | 2.45 | **2.19** | 2.47 | 2.34 | **2.19** | 2.47 | 2.44 | **2.19** | 2.47 | 2.47 | **2.04** | 2.47 |
| MS | 6.00 | **4.77** | 5.74 | 5.97 | **3.92** | 5.97 | 5.81 | **4.02** | 5.81 | 5.81 | **4.02** | 5.81 | **5.44** | 5.66 | 5.59 |
| River | 0.33 | **0.28** | **0.28** | 0.33 | **0.28** | **0.28** | 0.33 | **0.28** | **0.28** | 0.33 | **0.28** | **0.28** | **0.28** | 0.33 | 0.33 |
| Kenya | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | **0.15** | 0.17 |
| Malawi | 0.12 | 0.12 | 0.12 | 0.12 | 0.12 | 0.12 | 0.12 | 0.12 | **0.10** | 0.12 | **0.10** | 0.69 | 0.12 | 0.11 | **0.10** |

Table 3: The RMSEs (estimated by 10-fold CV) of the models obtained with SCLUS, CLUS, CLUS*, GWR, SVR and M5' Trees. Best results are given in bold. Results for NWE are multiplied by $10^3$. The implementation of CLUS* supports only datasets that contain coordinates and not relative distances between the examples in the dataset which is the case with the River dataset. Note that SCLUS does not have such implementation problems, whereas CLUS does not use the spatial information at all.

| Dataset | auto. chosen b (%) | SCLUS_Moran | | | | | | SCLUS_Geary | | | | | | CLUS ($\alpha$ = 1) | CLUS* | GWR | SVR | M5' |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\alpha$=0 | | | $\alpha$=0.5 | | | $\alpha$=0 | | | $\alpha$=0.5 | | | | | | | |
| | | Euc. | Mod. | Gauss. | Euc. | Mod. | Gauss. | Euc. | Mod. | Gauss. | Euc. | Mod. | Gauss. | | | | | |
| FF | 100 | **42.82** | 47.21 | 47.21 | 56.55 | 56.55 | 57.76 | **42.82** | 56.55 | 57.76 | **42.82** | 57.76 | 57.76 | 49.21 | 47.22 | 373.3 | 64.58 | 63.70 |
| NWE | 7.7 | **2.16** | 2.36 | 2.39 | 2.48 | 2.40 | 2.45 | 2.47 | 2.40 | 2.40 | 2.49 | 2.55 | 2.53 | 2.46 | 2.47 | 2.38 | 2.50 | 2.50 |
| Foixa | 64.2 | 2.53 | 2.55 | 2.41 | **2.34** | 2.55 | 2.44 | 2.56 | 2.58 | 2.55 | 2.55 | 2.56 | 2.59 | 2.65 | 2.52 | 2.54 | 2.95 | 2.86 |
| GASD | 2.5 | 0.18 | 0.18 | 0.17 | 0.17 | 0.18 | 0.17 | 0.18 | 0.18 | 0.17 | 0.18 | 0.17 | 0.17 | 0.16 | 0.16 | 0.35 | **0.14** | 0.16 |
| MF | 9.1 | 2.47 | **2.04** | 2.47 | 2.29 | 2.29 | 2.29 | 2.47 | 2.65 | 3.00 | 2.47 | 2.30 | 2.30 | 2.35 | 2.54 | 2.85 | 2.80 | 2.46 |
| MS | 4.8 | **5.44** | 5.66 | 5.59 | 5.81 | 5.81 | 5.81 | 6.60 | 5.50 | 5.50 | 5.80 | 5.83 | 5.55 | 5.64 | 6.68 | 6.77 | 8.60 | 6.37 |
| River | 100 | **0.28** | 0.33 | 0.33 | 0.30 | **0.28** | **0.28** | **0.28** | 0.31 | 0.30 | 0.32 | 0.30 | 0.30 | 0.30 | - | 0.52 | 0.30 | 0.30 |
| Kenya | 63.8 | 0.17 | 0.15 | 0.17 | 0.18 | **0.14** | 0.15 | 0.16 | 0.15 | 0.15 | 0.15 | 0.15 | **0.14** | 0.15 | 0.15 | 0.32 | 0.15 | 0.15 |
| Malawi | 8.1 | 0.12 | 0.11 | 0.10 | **0.07** | 0.09 | 0.09 | 0.12 | 0.12 | 0.11 | 0.08 | **0.07** | 0.09 | 1.21 | 1.26 | 0.09 | 0.32 | 0.38 |

Table 4: Regression: each tabulated value represents the number of wins minus the number of losses when comparing the errors obtained by using SCLUS to all other models, over all datasets.

| | SCLUS vs CLUS | SCLUS vs CLUS* | SCLUS vs GWR | SCLUS vs SVR | SCLUS vs M5' |
|---|---|---|---|---|---|
| Moran $\alpha$=0.0, Euc. | 3 | 2 | 7 | 5 | 3 |
| Moran, $\alpha$=0.0, Mod. | 0 | 3 | 3 | 4 | 2 |
| Moran, $\alpha$=0.0, Gauss. | 1 | 4 | 5 | 3 | 1 |
| Moran, $\alpha$=0.5, Euc. | -2 | -2 | 6 | 4 | 4 |
| Moran, $\alpha$=0.5, Mod. | 3 | 2 | 4 | 7 | 7 |
| Moran, $\alpha$=0.5, Gauss. | 2 | 3 | 6 | 6 | 6 |
| Geary, $\alpha$=0.0, Euc. | -1 | 1 | 3 | 5 | 1 |
| Geary, $\alpha$=0.0, Mod. | 0 | -1 | 1 | 2 | 2 |
| Geary, $\alpha$=0.0, Gauss | 1 | -1 | 1 | 3 | 3 |
| Geary, $\alpha$=0.5, Euc. | 0 | 1 | 5 | 4 | 4 |
| Geary, $\alpha$=0.5, Mod. | -3 | -1 | 5 | 3 | 3 |
| Geary, $\alpha$=0.5, Gauss. | 2 | 0 | 4 | 4 | 4 |

In addition, the results in Table 3 show that, in most cases, there is at least one configuration of SCLUS that outperforms CLUS in terms of RMSE error (except for the GASD dataset) and that there is great difference among the results for some of the datasets in favor of SCLUS. Moreover, the two modifications of CLUS, SCLUS and CLUS*, outperform by a great margin the standard regression method GWR that incorporates spatial autocorrelation and accounts for non-stationarity. Furthermore, SCLUS compares very well to standard regression tree-based methods like M5' Regression Trees, as well as non tree-based methods as SVR that do not consider autocorrelation.

Next, focusing on the comparison between SCLUS and CLUS*, we have to emphasize the fact that both SCLUS and CLUS* are modifications of CLUS that are designed to improve (if possible) both the spatial homogeneity and the accuracy of the CLUS models by modifying/enhancing the heuristic (variance reduction) used to evaluate each split in the process of tree construction. Whereas SCLUS accounts for autocorrelation that is often present in the data, CLUS* accounts for the spatial coordinates (usually presented in pairs (x, y) or (latitude, longitude)) in the case of the data obtained from spatial datasets by considering them as response variables in addition to the actual ones. This means that CLUS* aims at generating PCTs that will maximize the inter-cluster variance reduction of both the target and the coordinates[6]. Moreover, much higher importance is given to the spatial information in CLUS* than to the actual response, as they all are normalized at the beginning of the modeling process and equal importance is given to them: with a single tree target and two coordinates x and y as additional targets. This solution makes the predictions of the CLUS* models more coherent in space than those of the CLUS models and (if possible) increases the accuracy of the models.

However, CLUS* models cannot deal with non-stationary autocorrelation (i.e., when autocorrelation varies significantly throughout the space), while SCLUS models deal with non-stationarity appropriately. In SCLUS, two different geographical regions that have the same distribution of the independent variables and target values which exhibit autocorrelation, can be covered by one leaf of the tree: In CLUS*, the data will need to be split into different regions due to the spatial homogeneity. Moreover, CLUS* cannot handle different definitions of the regression problem that can arise from different definitions of the space, e.g., using different similarity / proximity measures.

To summarize the results presented in Table 3, we count the number of wins/losses of the twelve configurations of SCLUS when compared to all other methods.

[6] This is possible as CLUS is designed to deal with multi-target prediction problems.

Table 5: The Precision (estimated by 10-fold CV) of the models obtained with SCLUS, CLUS and CLUS*. Best results are given in bold.

| Dataset | SCLUS_Moran | | SCLUS_Geary | | CLUS | CLUS* |
|---|---|---|---|---|---|---|
| | $\alpha$=0 | $\alpha$=0.5 | $\alpha$=0 | $\alpha$=0.5 | ($\alpha$ = 1) | |
| Kras | 0.35 | 0.35 | 0.35 | **0.42** | 0.41 | 0.40 |
| Prim | 0.78 | 0.69 | 0.69 | 0.76 | 0.78 | **0.80** |
| Foixa_01 | 0.78 | 0.76 | 0.75 | 0.79 | 0.77 | **0.80** |
| Foixa_045 | 0.30 | 0.35 | 0.43 | **0.46** | 0.45 | 0.40 |

Table 6: The Recall (estimated by 10-fold CV) of the models obtained with SCLUS, CLUS and CLUS*. Best results are given in bold.

| Dataset | SCLUS_Moran | | SCLUS_Geary | | CLUS | CLUS* |
|---|---|---|---|---|---|---|
| | $\alpha$=0 | $\alpha$=0.5 | $\alpha$=0 | $\alpha$=0.5 | ($\alpha$ = 1) | |
| Kras | 0.99 | 0.99 | 0.99 | 0.98 | **1.00** | **1.00** |
| Prim | 0.97 | 0.97 | 0.98 | 0.97 | **1.00** | **1.00** |
| Foixa_01 | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 |
| Foixa_045 | **1.00** | **1.00** | **1.00** | **1.00** | 0.99 | 0.99 |

In Table 4, we report an overall number representing the number of wins minus the number of losses, when each of the twelve configurations of SCLUS (see Table 3) are compared to CLUS*, GWR, SVR and M5' Regression Trees. A positive number means that SCLUS has more wins than losses than the other method, whereas a negative number means the opposite. The results are in favor of SCLUS in most cases. The best results are obtained using SCLUS with Global Moran's $I$, *alpha* = 0.5 and Modified /Gaussian weighing function. Section 5.3 discusses some further characteristics of the learned PCTs.

### *5.2. Classification tasks*

In Tables 5 and 6, we report the precision and recall, estimated by 10-fold CV, obtained by applying SCLUS, CLUS and CLUS* to the spatially defined classification datasets. For this comparison, we have run SCLUS using the automatically chosen bandwidth and the Euclidean weighting schema. In terms of precision, Geary's $C$ seems to lead to more accurate (precise) predictive models than Moran's $I$, whereas no difference is observed when analyzing the recall. When comparing errors obtained by SCLUS with errors obtained by CLUS and CLUS*, we note that, differently from in the case of regression tasks, there are only few cases where SCLUS outperforms the other systems. However, in the remaining cases, the precision and recall values of SCLUS are comparable to those of the competitors. Also for classification, Section 5.3 discusses some further characteristics of the learned PCTs.

### *5.3. Properties of the models: size, spatial dispersion and learning times*

Table 7 shows the average size of the trees (number of leaves) built by SCLUS, CLUS, CLUS* and M5' Trees (only on the regression problems). The results suggest several considerations.

First, the average size of the trees learned by SCLUS is independent of the autocorrelation measure used. Exceptions are the datasets FF and Foixa_045, where trees obtained with Geary's $C$ are larger than their counterparts obtained with Moran's $I$.

Second, the average size of trees learned by SCLUS depends of the use of the autocorrelation measure used in the clustering phase. Trees induced by SCLUS by using only the autocorrelation measure in the clustering phase ($\alpha = 0$) are smaller, in over 90% of the cases, than trees computed by CLUS ($\alpha = 1$) and CLUS*. Similarly, trees induced by SCLUS using $\alpha = 0.5$ in the clustering

Table 7: Average size of the trees learned by SCLUS, CLUS, CLUS* and M5' Trees.

| Dataset | SCLUS_Moran | | SCLUS_Geary | | CLUS | CLUS* | M5' |
|---|---|---|---|---|---|---|---|
| | $\alpha$=0.0 | $\alpha$=0.5 | $\alpha$=0.0 | $\alpha$=0.5 | ($\alpha$ = 1) | | |
| FF | 1.0 | 1.4 | 1.5 | 4.8 | 1.8 | 1.0 | 1.0 |
| NWE | 1.3 | 4.4 | 1.7 | 5.3 | 5.6 | 2.3 | 4.0 |
| Foixa | 1.8 | 2.3 | 2.4 | 2.3 | 4.7 | 6.1 | 3.0 |
| GASD | 10 | 31.2 | 8.0 | 28.9 | 27.7 | 23.8 | 49.0 |
| MF | 1.0 | 4.2 | 1.1 | 4.2 | 5.1 | 19.7 | 6.0 |
| MS | 1.4 | 6.1 | 1.1 | 5.3 | 6.0 | 19.2 | 6.0 |
| River | 1.0 | 1.1 | 1.0 | 1.0 | 1.1 | - | 1.0 |
| Kenya | 2.1 | 3.7 | 2.7 | 2.9 | 3.4 | 3.1 | 5.0 |
| Malawi | 12.9 | 67.8 | 10.5 | 69.8 | 30.1 | 26.4 | 70.0 |
| Kras | 3.5 | 3.4 | 4.4 | 3.2 | 6.3 | 5.0 | - |
| Prim | 3 | 4.0 | 1.3 | 3.9 | 4.2 | 4.0 | - |
| Foixa_01 | 6.4 | 9.2 | 9.8 | 9.3 | 12.6 | 13.0 | - |
| Foixa_045 | 12.7 | 15.7 | 21.2 | 22.0 | 15.0 | 18.0 | - |

Table 8: Average autocorrelation of the prediction errors committed on the testing sets, performed by PCTs learned by SCLUS, CLUS, CLUS* and M5', as well as SVR and GWR. For each dataset, the best results (the smallest in absolute value) are given in bold.

| Dataset | SCLUS_Moran | | SCLUS_Geary | | CLUS | CLUS* | GWR | SVR | M5' |
|---|---|---|---|---|---|---|---|---|---|
| | $\alpha$=0.0 | $\alpha$=0.5 | $\alpha$=0.0 | $\alpha$=0.5 | ($\alpha$ = 1) | | | | |
| FF | **-0.02** | **-0.02** | **-0.02** | **-0.02** | 1.00 | **-0.02** | **-0.02** | **-0.02** | 0.98 |
| NWE | **0.00** | -0.01 | 0.06 | 0.07 | 0.84 | -0.01 | -0.01 | -0.01 | -0.01 |
| Foixa | -0.02 | -0.02 | -0.02 | **0.01** | 0.96 | -0.02 | **0.01** | -0.03 | -0.02 |
| GASD | 0.19 | 0.19 | 0.26 | 0.15 | 1.00 | 0.08 | 0.39 | **0.01** | 0.37 |
| MF | **-0.01** | 0.15 | 0.08 | 0.20 | 0.88 | 0.15 | 0.19 | **-0.01** | 0.14 |
| MS | 0.13 | 0.24 | 0.24 | 0.19 | 0.66 | 0.13 | 0.38 | **-0.01** | 0.34 |
| River | -0.03 | -0.03 | -0.03 | -0.03 | -0.03 | - | -0.03 | -0.03 | -0.03 |
| Kenya | **-0.05** | -0.06 | **-0.05** | -0.07 | 0.85 | 0.94 | 0.85 | -0.08 | 0.92 |
| Malawi | **0.03** | -0.20 | 0.32 | -0.16 | 0.52 | 0.34 | 0.82 | 0.30 | 0.82 |

phase are smaller, in 75% of the cases than trees computed by CLUS ($\alpha = 1$) and CLUS*. This means that clustering spatial data according to autocorrelation guarantees smaller trees. Smaller trees are usually more comprehensible and simpler to interpret.

The models built with CLUS* are smaller (in 61% of the cases) than the ones built with CLUS, but this is not systematic because CLUS* cannot handle different definitions (in terms of similarity/ proximity measures) of the regression problem that can arise from different definitions of the underlying space, which differs in different datasets. The predictions of the PCTs learned by CLUS* are more coherent in space in comparison with the PCTS learned by CLUS, but differently from SCLUS, this happens at the price of increasing the size of the tree models. While SCLUS can consider two different geographical regions that have the same distribution of attribute and target values including autocorrelation in one leaf of the tree, CLUS* will split these due to the spatial homogeneity. This is the reason of the increase of the tree size.

A final consideration concerns M5' that, although it builds model trees (theoretically a model tree is more accurate than a regression tree in prediction), the fact that it ignores the spatial dimension of the data leads to the construction of larger trees. In 83% of the cases SCLUS builds trees that are smaller than the ones built with M5'.

In Table 8 we present the autocorrelation of the errors achieved by SCLUS, CLUS and CLUS*. In addition, we also present the errors obtained by using the competitive solutions M5' Regression Trees, SVR and GWR. Here, autocorrelation is computed by means of the Moran's $I$ on the errors estimated on the test set. We analyze the obtained models in terms of this measure in order to show that PCTs learned by SCLUS can capture the autocorrelation present in the data and generate predictions that exhibit small (absolute) autocorrelation in the errors.

Table 9: The spatial dispersion of the clusterings produced by SCLUS, CLUS and CLUS*. For each dataset, the best results (the smallest in absolute value) are given in bold.

| Dataset | SCLUS_Moran | | SCLUS_Geary | | CLUS | CLUS* |
|---|---|---|---|---|---|---|
| | $\alpha$=0.0 | $\alpha$=0.5 | $\alpha$=0.0 | $\alpha$=0.5 | ($\alpha$ = 1) | |
| FF | **0.00** | 0.02 | **0.00** | **0.00** | 0.04 | **0.00** |
| NWE | **71.22** | 962.72 | 201.54 | 562.71 | 222.56 | 348.20 |
| Foixa | **25.69** | 40.90 | 69.03 | 969.34 | 571.54 | 450.33 |
| GASD | 124146.78 | 232436.63 | **76724.80** | 204045.23 | 180685.45 | 88681.55 |
| MF | **0.01** | 0.20 | 0.21 | 0.45 | 0.55 | 0.66 |
| MS | **0.17** | 0.32 | 0.22 | 0.42 | 0.21 | 0.59 |
| River | **0.00** | **0.00** | **0.00** | **0.00** | 7.44 | - |
| Kenya | **0.02** | 0.03 | 1.49 | 1.48 | 0.03 | **0.02** |
| Malawi | **1762.93** | 7156.39 | 2879.73 | 10204.51 | 2812.67 | 2499.54 |
| Kras | 378.22 | **366.74** | 542.82 | 435.31 | 1438.42 | 890.46 |
| Prim | 594.15 | 619.56 | 625.70 | 618.91 | 553.88 | **379.36** |
| Foixa_01 | 232.12 | 280.48 | 329.71 | 329.71 | 245.96 | **200.39** |
| Foixa_045 | 571.80 | 743.26 | 1977.52 | 1296.96 | 1355.76 | **522.83** |

Table 10: Learning times (seconds) for SCLUS, CLUS, CLUS*, GWR, SVR and M5' Trees.

| Dataset | SCLUS_Moran | | SCLUS_Geary | | CLUS | CLUS* | GWR | SVR | M5' |
|---|---|---|---|---|---|---|---|---|---|
| | $\alpha$=0.0 | $\alpha$=0.5 | $\alpha$=0.0 | $\alpha$=0.5 | ($\alpha$ = 1) | | | | |
| FF | 1.50 | 1.03 | 2.57 | 2.19 | 0.04 | 0.04 | 11.33 | 1.03 | 1.23 |
| NWE | 2.31 | 1.62 | 0.85 | 0.58 | 0.11 | 0.11 | 55.90 | 1.22 | 1.94 |
| Foixa | 0.69 | 0.49 | 1.88 | 2.01 | 0.02 | 0.02 | 31.25 | 0.49 | 0.88 |
| GASD | 27.86 | 20.79 | 20.25 | 23.56 | 0.04 | 0.04 | 1808.53 | 30.45 | 20.43 |
| MF | 2.68 | 1.39 | 2.08 | 2.81 | 0.04 | 0.03 | 24.17 | 1.00 | 2.25 |
| MS | 3.39 | 1.41 | 2.03 | 1.73 | 0.04 | 0.03 | 27.12 | 0.59 | 3.55 |
| River | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 | - | 1.46 | 0.09 | 0.05 |
| Kenya | 1.18 | 1.08 | 1.19 | 1.34 | 0.02 | 0.02 | 12.53 | 0.02 | 1.20 |
| Malawi | 62.92 | 69.33 | 66.64 | 100.73 | 0.46 | 0.42 | 23.45 | 10.28 | 7.36 |
| Kras | 508.09 | 700.00 | 355.97 | 700.00 | 0.91 | 1.09 | - | - | - |
| Prim | 743.89 | 1154.38 | 449.87 | 699.79 | 1.53 | 1.48 | - | - | - |
| Foixa_01 | 0.61 | 0.80 | 0.87 | 0.91 | 0.02 | 0.02 | - | - | - |
| Foixa_045 | 0.62 | 0.77 | 0.85 | 1.02 | 0.02 | 0.02 | - | - | - |

The analysis of the results reveals that SCLUS handles the autocorrelation better than CLUS. The autocorrelation of the errors of PCTs learned by SCLUS is much lower than the one obtained by CLUS. In fact, coherently with the discussion reported in Section 2.1, SCLUS is able to correctly remove the effect of autocorrelation when making predictions. Thus, it is able to obtain spatially consistent predictions. This analysis also reveals that CLUS* is able to capture autocorrelation better than CLUS, but less than SCLUS. CLUS* gives lower autocorrelation of the errors than CLUS in 78% of the cases and than SCLUS in 30% of the cases. This is expected, according to the differences between SCLUS and CLUS*, already discussed in this Section. Moreover, as expected, the autocorrelation of the errors is lower when $\alpha = 0$.

Classical data mining methods like M5', have some level of spatial autocorrelation left in the errors which means that the errors obtained by using these methods may be underestimated (Davis, 1986) and the i.i.d. assumption violated. On the other hand, in the case of GWR, the level of spatial autocorrelation in the residuals of the linear model is much lower that M5', but higher than the one measured in the SCLUS models. Only in 17% of the cases autocorrelation in the errors is lower than the autocorrelation left in the errors of the SCLUS models. This means that SCLUS can better remove the effect of the autocorrelation in the errors of the models, whereas GWR failures to include or adequately measure autocorrelated variables, in most cases.

The spatial dispersion of the clusterings produced by SCLUS, CLUS and CLUS* are reported in Table 9. This measure is computed for the clustering as applied to the testing unseen data[7]. The analysis of these results reveals that, overall, SCLUS (in 70% of the cases) and CLUS* (in 75% of the cases) clusterings are more compact than the CLUS clusterings. Moreover, as expected, most of the PCTs induced by SCLUS with $\alpha = 0$ are more compact than PCTs induced with $\alpha = 0.5$, as the former uses only spatial autocorrelation as a splitting criterion when building the PCT. For $\alpha = 0.5$ in SCLUS, the produced clusters are less dispersed (have higher spatial dispersion), than those learned by CLUS*.

Finally, the overall spatial dispersion of the clusters induced by CLUS* and SCLUS with $\alpha = 0.0$ are comparable. This is expected, considering how the spatial dispersion is computed. In fact, its definition relies on the average Euclidean intra-distance for each pair of examples that fall in the same cluster and CLUS*, by considering coordinates as target variables, tends to add splits to the tree which group examples with similar (close) coordinates. Note that the models that have better spatial dispersion tend to have lower autocorrelation of their errors.

Finally, Table 10 reports the average learning times for SCLUS, CLUS, CLUS*, GWR, SVR and M5' Regression Trees. Overall, the smallest learning times are obtained by using the CLUS algorithm. The learning times for CLUS* are similar (slightly larger) than the running times of CLUS, as in this configuration CLUS is run by considering the spatial coordinates as responses and the time complexity of PCTs induction remains roughly the same. The learning times for SCLUS are longer than the learning times for CLUS, because the consideration of the autocorrelation introduces additional computations and increases the complexity of building a PCT. This is in line with the time complexity analysis reported in Section 3.3. The learning times for SVR and M5' Trees are smaller than those of SCLUS and longer than those of CLUS. The learning times for GWR are much longer than those of all other algorithms, because GWR creates many different local models.

### *5.4. Comparison of the predictive models*

Besides having difference predictive performance (accuracy), the obtained models also have different structure. In this Section, we show the differences the among models learned by CLUS and SCLUS on ecological datasets (such as MF, MS and FOIXA). In Figure 1(a), 1(b), Figure 2(a) and 2(b), we analyze the regression trees learned from the MF and MS datasets by using CLUS/ SCLUS (with Global Moran's *I* and automatically estimated bandwidth). These datasets contain measurements of pollen dispersal (crossover) rates from two lines of plants (oilseed rape): the transgenic male-fertile (MF) and the non-transgenic male-sterile (MS). The measurements are taken at sampling points concentrically distributed around a central donor field.

We observe that the pair of trees learned by CLUS from the MF and MS datasets (see Figure 1(a) and Figure 2(a)), as well as the pair of trees learned by SCLUS from same datasets (see Figure 1(b) and Figure 2(b)), leave the same split nodes at their roots. The MF and MS datasets share the same predictor variables and there exists a strong dependence between response variables of both datasets.

The trees learned by the same algorithm, which uses the same heuristic function, are thus similar across the two datasets. CLUS and SCLUS, which use different heuristics, learn differently structured trees from the same dataset. In both datasets, SCLUS chooses the spatially-aware distance predictor variable (i.e., the distance of the sampling point from the center of the donor field) at the top level(s) of the tree, with other predictors at lower levels of the tree (see the case

[7] Zero values in Table 9 are due to the rounding down of intra-distances which are less than $10^{-3}$ to zero

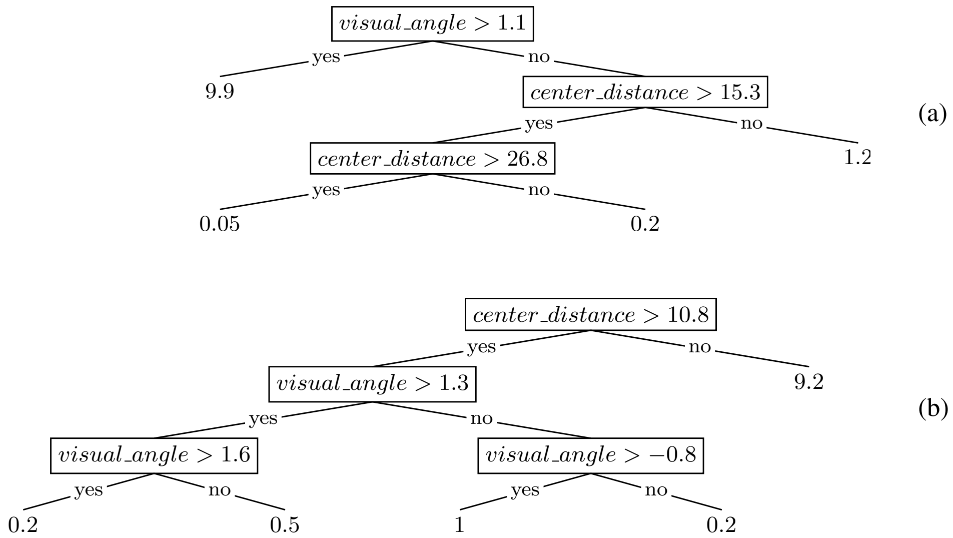

Figure 1: Two PCTs learned on one of the training folds of the MF dataset (a) The ordinary PCT learned by CLUS (b) The spatially-aware PCT learned by SCLUS.

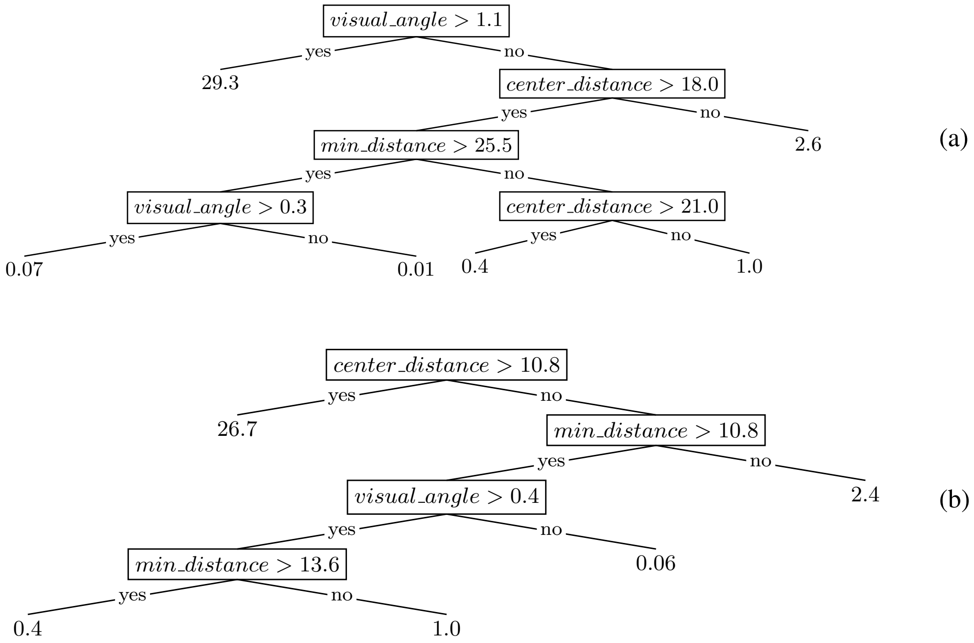

Figure 2: Two PCTs learned on one of the training folds of the MS dataset (a) The ordinary PCT learned by CLUS (b) The spatially-aware PCT learned by SCLUS.

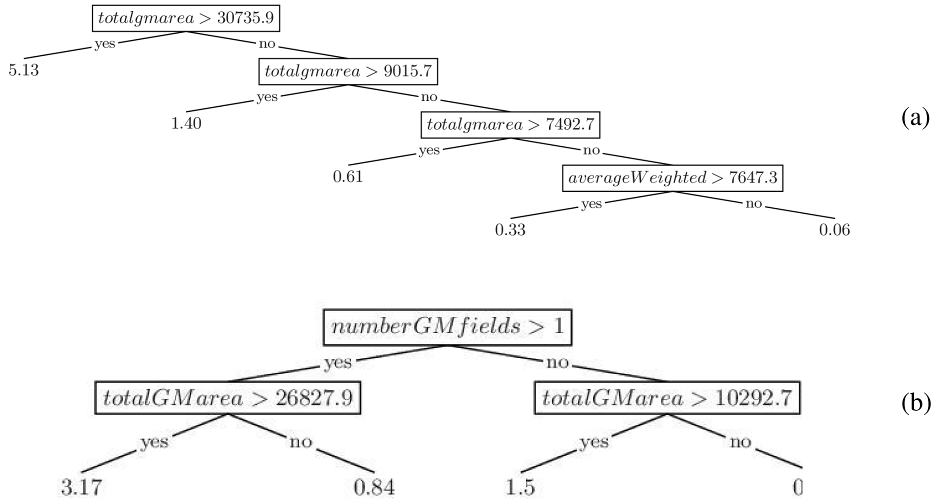

Figure 3: Two PCTs learned on one of the training folds of the Foixa dataset (a) The ordinary PCT learned by CLUS (b) The spatially-aware PCT learned by SCLUS.

of the MS/MF dataset in Figures 2(b)/1(b)). In the models obtained by CLUS, the distance to the donor field is used only to specify the target values at the bottom levels of tree, whereas the visual angle between the sampling plot and the donor field is chosen at the top level(s) of the tree. The choice of the distance attribute in SCLUS seems to be the one that better captures a global trend in the data, i.e., the concentric distribution of the pollen dispersal (crossover) rates (see Figure 4(a)). This makes the spatially-aware PCT more interpretable and understandable than the corresponding ordinary PCT.

Further on, in Figure 3(a) and Figure 3(b), we show the regression trees for the FOIXA dataset, obtained by using CLUS and SCLUS (with Global Moran *I* and *b*=20%), respectively. This dataset contains measurements of the rates of outcrossing at sampling points located within a conventional field, due to pollen inflow from the surrounding genetically modified (GM) fields. The studied fields are situated within the Foixa region in Spain.

Similarly as for the MF and MS datasets, CLUS and SCLUS learn differently structured trees. The models have different splits at the root node. The tree obtained by CLUS has the size of the surrounding GM fields (totalgmarea) at the root of the tree, as well as in most of the nodes of the tree, which means that the predictions in the leaves are based only on the values of one predictor variables and the other variables are practically of no importance. In contrast, the spatially-aware PCT has the number of surrounding GM fields (numberGMfields) at the root of the tree and the size of the surrounding GM fields (totalgmarea) in the other nodes of the tree. In this case, the predictions in the leaves are based on the values of two predictor variables and according to domain expertise, as in the case of MF/MS dataset, the spatially-aware PCT is more interpretable and understandable than the corresponding ordinary PCT.

A more detailed analysis of the training examples falling in the leaves of both the ordinary PCT and the spatially-aware PCT revealed that the leaves of both trees cluster examples with similar response values (this is due to the variance reduction). However, the training examples

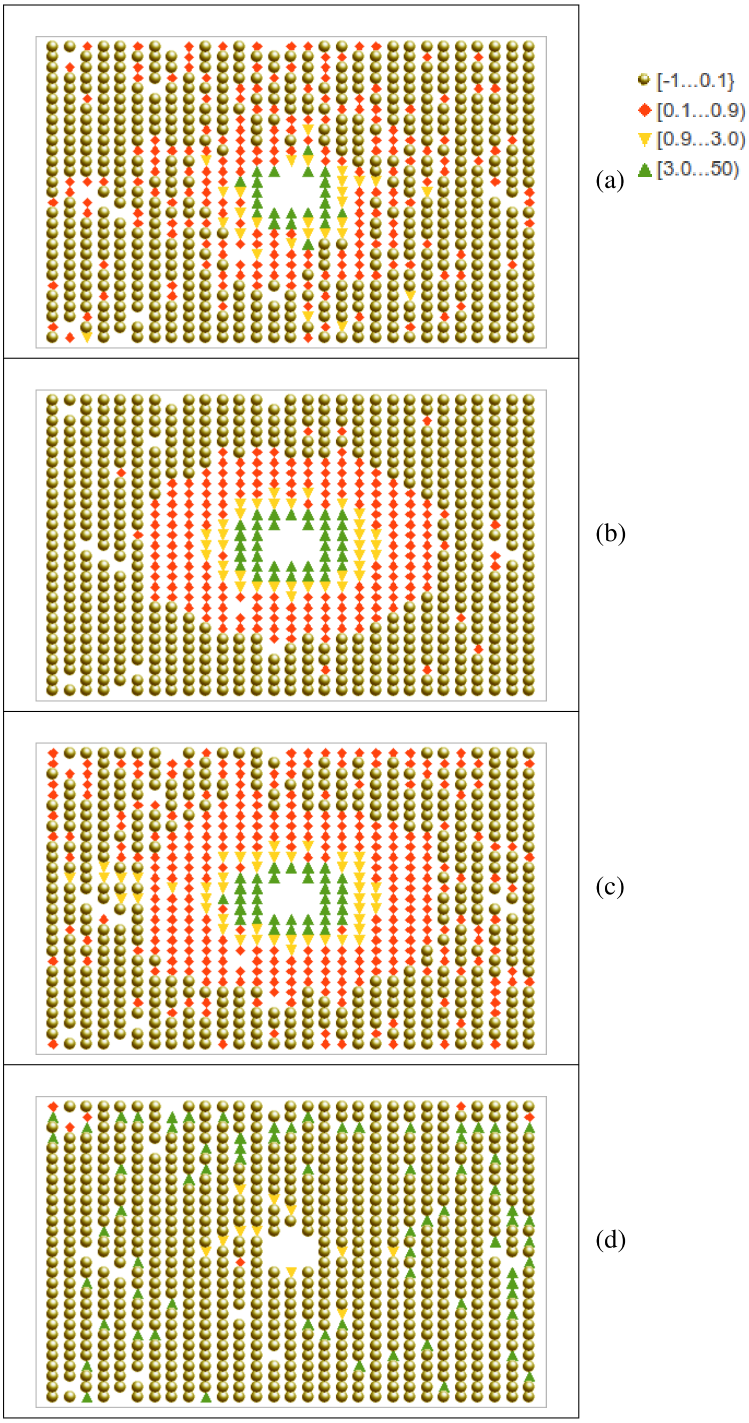

Figure 4: The pollen dispersal (crossover) rates of the MF dataset: (a) measured values (b) values predicted by the CLUS model, (c) values predicted by the SCLUS model and (d) values predicted by the GWR model.

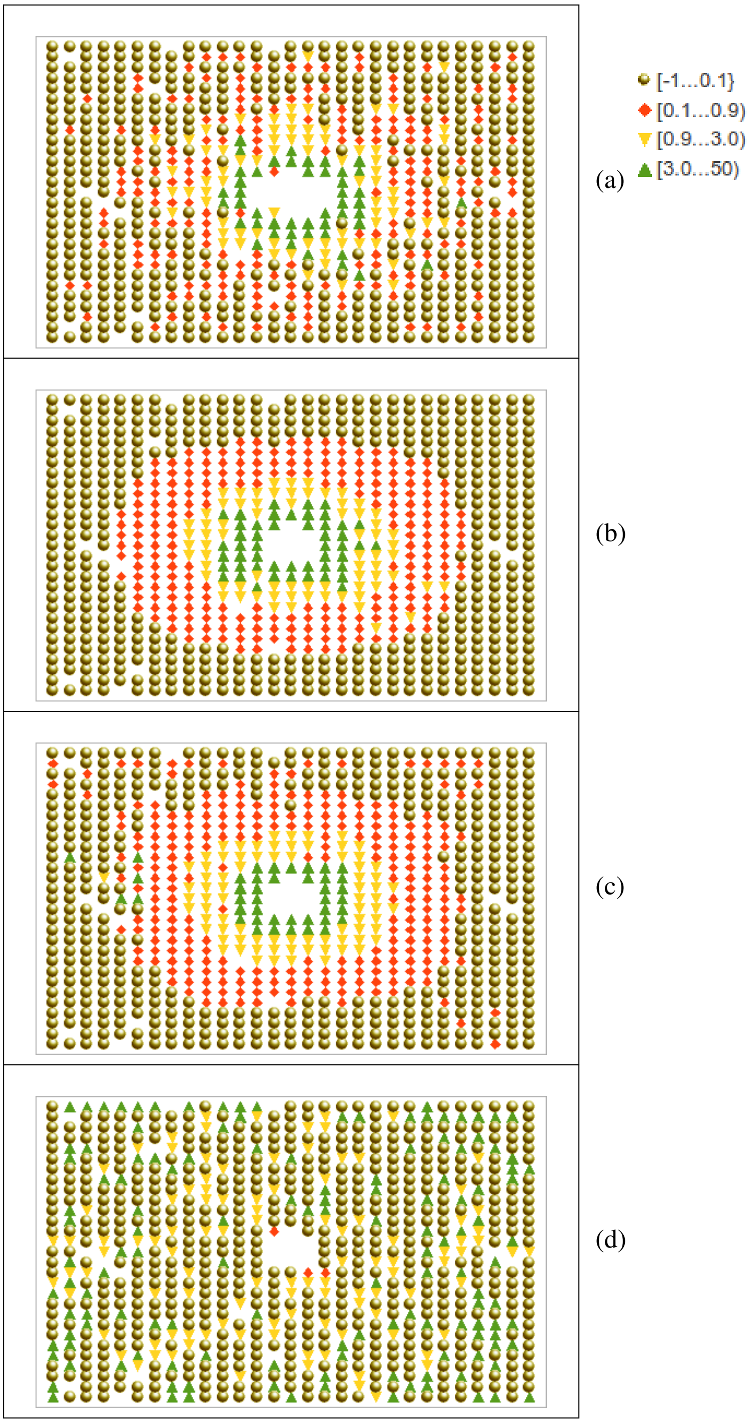

Figure 5: The pollen dispersal (crossover) rates of the MS dataset: (a) measured values (b) values predicted by the CLUS model, (c) values predicted by the SCLUS model and (d) values predicted by the GWR model.

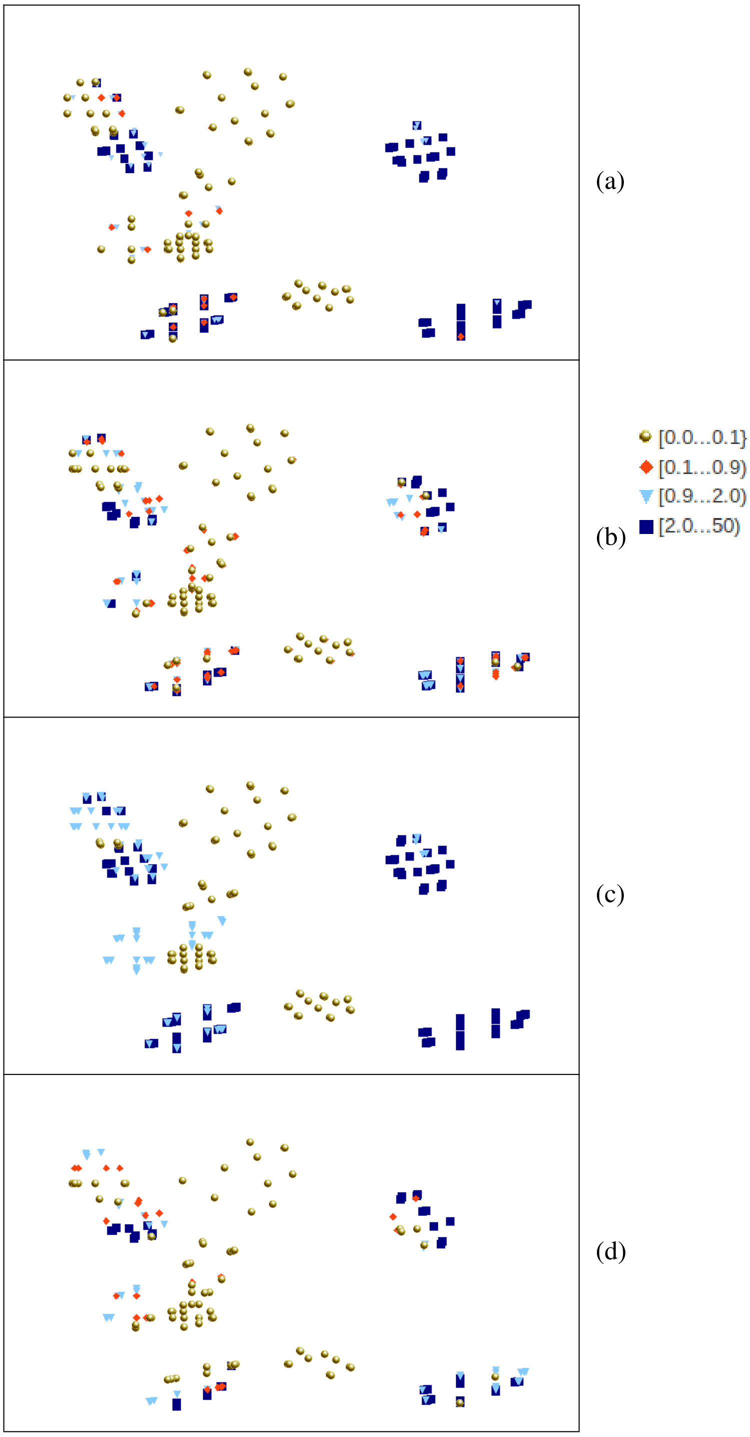

Figure 6: The outcrossing rate for selected sampling points of the FOIXA dataset: (a) measured values (b) values predicted by the CLUS model, (c) values predicted by the SCLUS model and (d) values predicted by the GWR model.

falling in the leaves of the spatially-aware PCTs are also close in space. This guarantees spatially smoothed predictions, where the predictions that are close to each other in space tend to have similar values.

### *5.5. Visual differences in the predictions made by the models*

In order to visualize the differences between the predictions of the SCLUS models and the models obtained by the other approaches, we use the MS, MF and Foixa datasets. The maps given in Figures 4 and 5 represent the predictions for the outcrossing rates for the testing (unseen) examples for the MF and MS datasets, respectively.

For the MF dataset, the real (measured) target are given in Figure 4(a), whereas the predictions obtained by the models learned with CLUS, SCLUS and GWR are given in Figures 4(b), 4(c) and Figure 4(d), respectively. The predictions made by CLUS (Figure 4(b)) tend to overgeneralize the training data in the attempt of maximizing the variance reduction. This results in forming ideal concentric circles (Figure 4(a)), which roughly approximate the spatial distribution of the target variable. However, by simultaneously analyzing Figure 1(a) and Figure 4(b), it is possible to see that this spatial awareness is simply due to the presence of the attribute *center_distance* in the tree. On the other hand, the predictions made by SCLUS (Figure 4(c)) are spatially-aware and, at the same time, follow the spatial distribution of the real target values much more closely. GWR (Figure 4(d)) has poor performance and fails to capture the global spatial distribution of the target. This is due to the fact that GWR accounts only for local properties of data.

For the MS dataset, the predictions obtained of the models learned with CLUS, SCLUS and GWR are given in Figure 5(b), 5(c) and Figure 5(d), respectively. These figures suggest the same conclusions we have drawn from the MF dataset. However, the predictions made by CLUS and SCLUS appear more similar for the MS than for the MF dataset.

Finally, the geographical maps given in Figure 6 present the predictions for the outcrossing rates for the testing (unseen) examples for the Foixa dataset. The real target is given in Figure 6(a), whereas the predictions obtained by models learned with CLUS, SCLUS, and GWR are shown in Figures 6(b), 6(c) and 6(d), respectively. The map of predictions obtained by using the model learned by SCLUS for the Foixa dataset (Figure 6(c)) shows that the predictions are smoother than the ones of the CLUS model. This means that SCLUS predictions that are close to each other in space tend to have similar values (very high/ low outcrossing rates are very close to each other). When plotted on a map, they form a nice smooth continuous surface without sharp edges and discontinuities. In contrast, there are sharp and discontinuousness in the CLUS predictions, so that the corresponding map looks like a mixture of “salt and pepper”. In addition, the SCLUS models are more accurate (in terms of the obtained errors, Table 3) and more interpretable than the competitors’ models (Figure 3(b) and Figure 6(c)). The map of predictions for the Foixa dataset obtained by using the model learned by GWR (Figure 6(d)) also shows sharp edges and discontinuities. This is due to the fact that GWR builds local models at each point: these are independent of the models built at the neighboring points. While the GWR models exploit the positive autocorrelation between neighboring points and in this way accommodate stationary autocorrelation, their predictions are still less accurate than those obtained with the SCLUS models. In sum, the SCLUS models are more useful than those of the (both spatial and a-spatial) competitors because both the tree models and the predictions are more realistic, easier to interpret, and more accurate.

## 6. Conclusions

In this paper, we proposed an approach that builds Predictive Clustering Trees (PCTs) and explicitly considers non-stationary spatial autocorrelation. The novelty of our approach is that it approaches clustering by maximizing both variance reduction and cluster homogeneity (in terms of autocorrelation), when the addition of a new node to the tree is considered. The effects of autocorrelation are identified and taken into account, separately at each node of the tree. The resulting models adapt to local properties of the data, providing, at the same time, spatially smoothed predictions. Due to the generality of PCTs, our approach works for different predictive modeling tasks, including classification and regression, as well as some clustering tasks.

The approach can consider different weighting schemes (degrees of smoothing) when calculating spatial autocorrelation as well as different sizes of neighborhoods (bandwidth). A procedure for the automated determination of the appropriate bandwidth is also explored in our study. A novelty of our approach is in the use of well known measures of spatial autocorrelation, such as Moran's $I$ and Geary's $C$. Previous related work on using autocorrelation in decision trees was based on special purpose measures of spatial autocorrelation, such as spatial entropy, and were limited to classification.

An extensive experimental evaluation has been performed to empirically prove the effectiveness of the proposed approach. Nine geo-referenced data sets are used for regression tasks while four geo-referenced data sets are employed for classification tasks. The experimental results show that the proposed approach performs better than standard spatial statistics techniques such as geographically weighted regression, which considers spatial autocorrelation but can capture global regularities only. SCLUS can identify autocorrelation, when present in data, and thus generate predictions that exhibit smaller autocorrelation in the errors than other methods: It can also generate clusterings that are more compact and trees that are smaller in size. Furthermore, the spatial maps of the predictions made by SCLUS trees are smother and do not require further post-smoothing for successful use in practice.

Several directions for further work remain to be explored. The automated determination of the parameter $\alpha$ that sets the relative importance of variance reduction and autocorrelation during tree construction deserves immediate attention. In a similar fashion, one might consider selecting an appropriate spatial autocorrelation measure. Finally, we plan to focus on multi-target problems, explicitly taking into account autocorrelation on the combination of several target variables.

## Acknowledgment

This work is in partial fulfillment of the research objectives of the project "EMP3: Efficiency Monitoring of Photovoltaic Power Plants" funded by Fondazione Cassa di Risparmio di Puglia and the PRIN 2009 project "Learning Techniques in Relational Domains and Their Applications" funded by the Italian Ministry of University and Research (MIUR). Džeroski is supported by the Slovenian Research Agency (grants P2-0103, J2-0734, and J2-2285), the European Commission (grants HEALTH-2008-223451, ICT-2010-266722, and ICT-2011-287713). He is also supported by Operation no. OP13.1.1.2.02.0005 financed by the European Regional Development Fund (85%) and the Slovenian Ministry of Higher Education, Science and Technology (15%). Stojanova is supported by the Slovenian Research Agency (grant J2-2285). The authors thank Luis Torgo for kindly providing the Portuguese rivers dataset.

## References

Anselin, L., Bera, A., 1998. Spatial dependence in linear regression models with an application to spatial econometrics. In: Ullah, A., Giles, D. (Eds.), Handbook of Applied Economics Statistics. Springer, pp. 21–74.

Appice, A., Ceci, M., Malerba, D., 2010. Transductive learning for spatial regression with co-training. In: Proc. ACM Symposium on Applied Computing. ACM, pp. 1065–1070.

Arthur, G., 2008. A history of the concept of spatial autocorrelation: A geographer's perspective. Geographical Analysis 40 (3), 297–309.

Bel, L. Allard, D., Laurent, J., Cheddadi, R., Bar-Hen, A., 2009. CART algorithm for spatial data: application to environmental and ecological data. Computational Statistics and Data Analysis 53, 3082–3093.

Besag, J., 1974. Spatial Interaction and the Statistical Analysis of Lattice Systems. Journal of the Royal Statistical Society, Series B (Methodological) 36 (2), 192–236.

Blockeel, H., De Raedt, L., Ramon, J., 1998. Top-down induction of clustering trees. In: Proc. 15th Intl. Conf. on Machine Learning. Morgan Kaufmann, pp. 55–63.

Bogorny, V., Valiati, J. F., da Silva Camargo, S., Engel, P. M., Kuijpers, B., Alvares, L. O., 2006. Mining maximal generalized frequent geographic patterns with knowledge constraints. In: Proc. 6th IEEE Intl. Conf. on Data Mining. IEEE Computer Society, pp. 813–817.

Breiman, L., Friedman, J., Olshen, R., Stone, J., 1984. Classification and Regression trees. Wadsworth & Brooks.

Brent, R., 1973. Algorithms for Minimization without Derivatives. Prentice-Hall.

Ceci, M., Appice, A., 2006. Spatial associative classification: propositional vs structural approach. Journal of Intelligent Information Systems 27 (3), 191–213.

Cortez, P., Morais, A., 2007. A Data Mining Approach to Predict Forest Fires using Meteorological Data. In: Proc. 13th Portuguese Conf. on Artificial Intelligence. APPIA, pp. 512–523.

Cressie, N., 1990. The origins of kriging. Mathematical Geology 22, 239–252.

Davis, J., 1986. Statistics and data analysis in geology, 2nd Edition. Wiley and Sons.

Debeljak, M., Trajanov, A., Stojanova, D., Leprince, F., Džeroski, S., 2012. Using relational decision trees to model out-crossing rates in a multi-field setting. Ecological Modelling In press.

Demšar, D., Debeljak, M., Lavigne, C., Džeroski, S., 2005. Modelling pollen dispersal of genetically modified oilseed rape within the field. In: Abstracts of the 90th ESA Annual Meeting. The Ecological Society of America, p. 152.

Dubin, R. A., 1998. Spatial autocorrelation: A primer. Journal of Housing Economics 7, 304–327.

Džeroski, S., Gjorgjioski, V., Slavkov, I., Struyf, J., 2007. Analysis of time series data with predictive clustering trees. In: Proc. 5th Intl. Wshp. on Knowledge Discovery in Inductive Databases. Springer, pp. 63–80.

Ester, M., Kriegel, H., Sander, J., 1997. Spatial data mining: A database approach. In: Proc. 5th Intl. Symp. on Spatial Databases. Vol. 1262. Springer, pp. 47–66.

Fotheringham, A. S., Brunsdon, C., Charlton, M., 2002. Geographically Weighted Regression: The Analysis of Spatially Varying Relationships. Wiley.

Glotsos, D., Tohka, J., Soukka, J., Ruotsalainen, U., 2004. A new approach to robust clustering by density estimation in an autocorrelation derived feature space. In: Proc. 6th Nordic Symposium on Signal Processing. IEEE Computer Society, pp. 296–299.

Goodchild, M., 1986. Spatial autocorrelation. Geo Books.

Gora, G., Wojna, A., 2002. RIONA: A classifier combining rule induction and k-NN method with automated selection of optimal neighbourhood. In: Proc. 13th European Conf. on Machine Learning. Springer, pp. 111–123.

Griffith, D., 2003. Spatial autocorrelation and spatial filtering: gaining understanding through theory and scientific visualization. Advances in spatial science. Springer.

Huang, Y., Shekhar, S., Xiong, H., 2004. Discovering colocation patterns from spatial data sets: A general approach. IEEE Transactions on Knowledge and Data Engineering 16 (12), 1472–1485.

Jahani, S., Bagherpour, M., 2011. A clustering algorithm for mobile ad hoc networks based on spatial auto-correlation. In: Proc. Intl. Symposium on Computer Networks and Distributed Systems. IEEE Computer Society, pp. 136–141.

Jensen, D., Neville, J., 2002. Linkage and autocorrelation cause feature selection bias in relational learning. In: Proc. 9th Intl. Conf. on Machine Learning. Morgan Kaufmann, pp. 259–266.

Legendre, P., 1993. Spatial autocorrelation: Trouble or new paradigm? Ecology 74 (6), 1659–1673.

Legendre, P., Dale, M. R. T., Fortin, M.-J., Gurevitch, J., Hohn, M., Myers, D., 2002. The consequences of spatial structure for the design and analysis of ecological field surveys. Ecography 25 (5), 601–615.

LeSage, J. H., Pace, K., 2001. Spatial dependence in data mining. In: Grossman, R., Kamath, C., Kegelmeyer, P., Kumar, V., Namburu, R. (Eds.), Data Mining for Scientific and Engineering Applications. Kluwer Academic, pp. 439–460.

Li, H., Calder, C. A., Cressie, N., 2007. Beyond moran's i: Testing for spatial dependence based on the spatial autoregressive model. Geographical Analysis 39 (4), 357–375.

Li, X., Claramunt, C., 2006. A spatial entropy-based decision tree for classification of geographical information. Transactions in GIS 10, 451–467.

Macchia, L., Ceci, M., Malerba, D., 2011. Learning to rank by transferring knowledge across different time windows. In: Whsp. Proc. of 15th Conf. on ECML/PKDD. Springer.

Malerba, D., Appice, A., Varlaro, A., Lanza, A., 2005a. Spatial clustering of structured objects. In: Proc. 15th Intl. Conf. on Inductive Logic Programming. Springer, pp. 227–245.

Malerba, D., Ceci, M., Appice, A., 2005b. Mining model trees from spatial data. In: Proc. 9th European Conf. on Principles of Knowledge Discovery and Databases. Springer, pp. 169–180.

Mehta, M., Agrawal, R., Rissanen, J., 1996. SLIQ: A fast scalable classifier for data mining. In: Proc. 5th Intl. Conf. on Extending Database Technology. Springer, pp. 18–32.

Michalski, R. S., Stepp, R. E., 1983. Learning from observation: Conceptual clustering. In: Michalski, R. S., Carbonell, J. G., Mitchell, T. M. (Eds.), Machine Learning: An Artificial Intelligence Approach. Tioga, pp. 331–364.

Moran, P. A. P. Notes on Continuous Stochastic Phenomena. *Biometrika* **37** 17–23 (1950).

Ohashi, O., Torgo, L., Ribeiro, R. P., 2010. Interval forecast of water quality parameters. In: Proc. 19th European Conf. on Artificial Intelligence. Vol. 215. IOS Press, pp. 283–288.

Pace, P., Barry, R., 1997. Quick computation of regression with a spatially autoregressive dependent variable. Geographical Analysis 29 (3), 232–247.

Quinlan, R. J., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann.

Rinzivillo, S., Turini, F., 2004. Classification in geographical information systems. In: Proc. 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases. Springer, pp. 374–385.

Rinzivillo, S., Turini, F., 2007. Knowledge discovery from spatial transactions. Jornal of Intellegence and Information Systems 28 (1), 1–22.

Robinson, W. S., 1950. Ecological correlations and the behavior of individuals. American Sociological Review 15, 351–357.

Sampson, P. D., Guttorp, P., 1992. Nonparametric Estimation of Nonstationary Spatial Covariance Structure. Journal of the American Statistical Association 87, 108–119.

Scrucca, L., 2005. Clustering multivariate spatial data based on local measures of spatial autocorrelation. Tech. Rep. 20, Università di Puglia.

Shepard, D., 1968. A two-dimensional interpolation function for irregularly-spaced data. In: Proc. 23rd ACM National Conference. ACM, pp. 517–524.

Stojanova, D., Ceci, M., Appice, A., Malerba, D., Džeroski, S., 2011. Global and local spatial autocorrelation in predictive clustering trees. In: Proc. 14th Intl. Conf. on Discovery Science. Springer, pp. 307–322.

Stojanova, D., Kobler, A., Ogrinc, P., Ženko, B., Džeroski, S., 2012. Estimating the risk of fire outbreaks in the natural environment. Data Mining and Knowledge Discovery 24 (2), 411–442.

Tobler, W., 1970. A computer movie simulating urban growth in the Detroit region. Economic Geography 46 (2), 234–240.

Wang, Y., Witten, I., 1997. Induction of model trees for predicting continuous classes. In: Proc. Poster Papers of the European Conference on Machine Learning. Faculty of Informatics and Statistics, University of Economics, Prague, pp. 128–137.

Witten, I., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition. Morgan Kaufmann.

Zhang, P., Huang, Y., Shekhar, S., Kumar, V., 2003. Exploiting spatial autocorrelation to efficiently process correlation-based similarity queries. In: Proc. 8th Symp. on Advances in Spatial and Temporal Databases. Springer, pp. 449–468.

Zhao, M., Li, X., 2011. An application of spatial decision tree for classification of air pollution index. In: 19th Intl. Conf. on Geoinformatics. IEEE Computer Society, pp. 1–6.