SAPIENZA UNIVERSITA' DI ROMA

Scuola di Dottorato in Economia
Dottorato di Ricerca in STATISTICA ECONOMICA XXVI ciclo

SPATIAL REGRESSION IN LARGE DATASETS: PROBLEM SET SOLUTION

Thesis submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in
economic Statistics

by

Myriam Tabasso

Program Coordinator
Prof. Giuseppe Espa

Thesis Supervisor
Prof. Giuseppe Arbia

"Knowledge is that possession that no misfortune can destroy, no authority can revoke, and no enemy can control. This makes knowledge the greatest of all freedoms."

Bryant Harrison McGill

Abstract

In this dissertation we investigate a possible attempt to combine the Data Mining methods and traditional Spatial Autoregressive models, in the context of large spatial datasets.
We start to considere the numerical difficulties to handle massive datasets by the usual approach based on Maximum Likelihood estimation for spatial models and Spatial Two-Stage Least Squares.
So, we conduct an experiment by Monte Carlo simulations to compare the accuracy and computational complexity for decomposition and approximation techniques to solve the problem of computing the Jacobian in spatial models, for various regular lattice structures. In particular, we consider one of the most common spatial econometric models: spatial lag (or SAR, spatial autoregressive model).
Also, we provide new evidences in the literature, by examining the double effect on computational complexity of these methods: the influence of "size effect" and "sparsity effect".
To overcome this computational problem, we propose a data mining methodology as CART (Classification and Regression Tree) that explicitly considers the phenomenon of spatial autocorrelation on pseudo-residuals, in order to remove this effect and to improve the accuracy, with significant saving in computational complexity in wide range of spatial datasets: real and simulated data.

Acknowledgements

The work presented in this thesis would not have been possible without the support of many outstanding people. I take this opportunity to extend my sincere appreciation to all those who made this PhD thesis possible.
First, I would express my sincere gratitude to my supervisor, Prof. Giuseppe Arbia for his continuous encouragement, enthusiasm, and support along the way of making this thesis. I would like to thank him for advising me during all phases of my PhD, but also for giving me the freedom to find my own way in this path. His ideas and suggestions have guided my work and improved the quality of my research.
I have been inspired to his passion in his work, his brilliance and his dedication and I have benefited greatly from his example and guidance.
I would like to thank Prof. Giovanni Felici, researcher of CNR-IASI (National Research Council-Institute for System Analysis and Computer Science "Antonio Ruberti") for his collaboration and for providing me the informatic support for my experiments to allow me to carry out my research.
I would also like to thank Prof. Giuseppe Espa and Prof. Guido Pellegrini, coordinators of PhD in Economic Statistics.
Many thanks also goes to all my colleagues for sharing my joys and tears during my PhD studies.
Finally, I would like to express my deepest gratitude to my family for their love and understanding and especially for supporting me in the difficult moments over these years.

Rome, 4 April 2014 *Myriam Tabasso*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent technology has increased the ability to analyze data, but has simultaneously increased the amount of data available for analysis. Spatial data technologies such as global positioning systems (GPS), geographic information systems (GIS), and address geocoding have created an explosion in the size of these data sets. Extracting useful and interesting patterns from massive geo-spatial datasets is important for many application domains, such as regional economics, ecology and environmental management, public safety, transportation, public health, business and travel, because *space is everywhere*. In the spatial statistics and econometric literature one challenge arises from the increasing size of georeferenced datasets, especially in the massive datasets (Griffith, 2012).

A commonly employed linear regression specification that incorporates spatial autocorrelation contains a spatial autoregressive term of the form $\rho\mathbf{W}y$ (spatial lag model) or $\rho\mathbf{W}\epsilon$ (spatial error model) where ρ is a spatial autoregressive parameter, $\mathbf{W}$ is a $n \times n$ matrix of spatial weights and y, ϵ are $n \times 1$ vectors of observations on the dependent variable or unobservable error terms. The usual approach is to apply Maximum Likelihood estimation procedures (ML), originally suggested by Ord (1975). When using ML estimation for spatial model, well-known problem is the computation of the logarithm of the determinant of the Jacobian term $|I - \rho\mathbf{W}|$ (or log-Jacobian $\ln|I - \rho\mathbf{W}|$), especially in very large data sets. The maximization of log-likelihood involves a nonlinear optimization that requires the evaluation of the Jacobian for each new value of the parameter ρ.

To account the numerical difficulties to handle massive data sets we survey various decomposition and approximation techniques. Ord (1975) suggested to calculate the determinant by mean of eigenvalues, but as argued by Smirnov and Anselin (2001) this method becomes numerically unstable for matrices of more than 1000 observations and it does not consider the high degree of sparsity of the spatial weights matrix. In addition, it requires huge amounts of memory.

To minimize the computational burden, recent suggestions take to account the sparsity of $\mathbf{W}$ such as Cholesky factorizations or LU decomposition (Pace and Barry, 1997, Pace and Barry, 1997a,b), characteristic polynomial method (Smirnov and Anselin, 2001), trace-based (Smirnov and Anselin, 2009), Monte Carlo (Barry and Pace, 1999) and Chebyshev approxi-

mations (Pace and LeSage, 2004).
In alternative way to overcome this computational problem is suggested by Kelejian and Prucha (1999), the Generalized Method of Moments (GMM) approach. In this context, data mining and knowledge discovery techniques become essential tools for successful analysis of very large datasets.
So, in this thesis we would investigate approximate solutions for scaling the spatial econometric models for large spatial data analysis problem by spatial data mining techniques.
Spatial data mining, a subfield of data mining, is concerned with the discovery of interesting and useful but implicit knowledge in spatial databases. Common patterns discovered by data mining algorithms include descriptive patterns (e.g., clustering), explanatory patterns (e.g., association rules) and predictive patterns (e.g., classification rules and decision trees). The foundations of data mining algorithms are in statistics and machine learning. One of the goals of data mining algorithms is to scale up to analyze very large datasets which may not fit in the main memory. The requirements for mining geospatial data are different from classical data mining. The reason for this are the special properties of spatial data (spatial autocorrelation, heterogenity, complexity, high dimensionality), therefore extracting knowledge from spatial data requires special approach by modelling spatial properties combined with classical data mining algorithms.

1.1 Motivations and Problem Statement

The huge amount of spatial data and the complexity of spatial data types and spatial access methods make the efficiency of spatial data mining algorithm an important research challenge.
Recent progress in data mining techniques poses challenges and creates a chance for the advancement of mining knowledge from spatial data and bring opportunities to extend these methods in the spatial econometrics framework.
The requirements of mining geospatial databases differ from those of mining classical relational databases. Geospatial data are described by geographic space and feature space. Computational representations of geospatial information require an implied topological and geometric measurement framework which affects the patterns that can be extracted. Geospatial data are also spatially dependent, meaning that similar things cluster in space.
These properties make classical data mining algorithms, which assume that data are independently generated and identically distributed over space, inappropriate for geospatial data. In particular, we deal tree issues that cannot leave out of consideration:

1. *Heterogeneity of spatial objects*: each unit of analysis involves spatial objects of different types, such as a town and a highway. In spatial databases, objects of different types are organized in separate layers, each of which has a distinct set of attributes and possibly a geometry attribute represented in the vector mode.

2. *The implicit definition of spatial relationships among*: spatial objects have a locational

property which implicitly defines several spatial relationships between objects, such as topological, distance-based and directional.

3. *Spatial autocorrelation.* Formally, spatial autocorrelation (or spatial dependence, as it is typically called in statistics) is defined as the 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 randomly associated pairs of observations . Informally, spatial positive (negative) autocorrelation occurs when the values of a given property are highly uniform (different) among similar spatial objects in the neighborhood.

Extracting knowledge from geospatial data therefore requires special approaches.
There are several ways to do that. The first one is to invent new, spatially aware data mining algorithms. This is the spatial data mining approach.
The second approach is to explicitly model spatial properties and relationships in the preprocessing step and then apply classical data mining algorithms.
The spatial statistics and econometrics framework outlines that the estimation of the parameters of the spatial autoregressive model using Maximum Likelihood (ML) theory is computationally very expensive because of the need to compute the logarithm of the determinant of a large matrix.
Major contributions of this study include spatial econometrics modeling for large geospatial data analysis, characterization of exact and approximate solutions of the SAR model, and experimental comparison of the proposed solution on real and simulated large geodatasets.
In this thesis, we focus on several approaches to analyze the numerical difficulties to handle large georeferenced datasets in traditional spatial econometric models, in terms of computational complexity.
To address this problem, the **main research goal** of this thesis is to suggest some directions along which spatial econometric modeling could benefit from the cross-fertilization with spatial data mining techniques.
Our study shows how spatial autoregressive models can be efficiently implemented without loss of accuracy. So that large geospatial datasets which are spatially auto-correlated can be analyzed in a reasonable amount of time. The thesis attempts to find answers to the following questions:

- *What are the main aspects that affect the computational difficulties by using the traditional approach in the large spatial datasets?*
- *What is the improving on computational complexity by including the data mining techniques within the framework of traditional spatial statistics and econometrics?*
- *What are the possible extensions based on current achievements?*

The corresponding findings of our research are published in several conference and journal publications in the areas of machine learning and data mining, ecological modeling, ecological informatics and bioinformatics, spatial econometrics and spatial statistics.

Finally, the **main contributions** are:

- the analysis of two different effects: influence of sample size on computational complexity and influence of sparsity spatial weights matrix on computational complexity in traditional spatial econometric approach;
- the extension some data mining algorithms on spatial datasets to consider the phenomenon of spatial autocorrelation on pseudo-residuals (prediction erros) and to reduce the computational complexity w.r.t. traditional approach.

1.2 Organization of the Thesis

Previously conducted research related to the topic is described in the following Chapter 2. It gives a broad overview to the theorethical framework of spatial econometrics models and spatial data mining. It contains the definition of Spatial Data Mining and introduces some of the main techniques and algorithms. In particular describes the Classification and Regression Trees (CART).
In Chapter 3 we compare the accuracy and computational complexity of decomposition and approximation techniques, to solve the problem of computing the Jacobian in spatial models in a series of experiments for various regular lattice structures. Also, we show new evidences in terms of computational complexity by considering the effect sparsity of spatial weights matrix and the effect of size sample.
In Chapter 4, we propose a data mining methodology that explicitly considers the phenomenon of spatial autocorrelation in an important data mining methodology called Classification and Regression Trees (CART). It represents an attempt to explore the combination of data mining methods and traditional spatial autoregressive models. Also, we discuss further developments of data mining in this context.
Finally, we draw general conclusions to summarize our research work.

Chapter 2

Spatial Linear Regression Models and data mining

This chapter discusses popular specifications of spatial econometric models. Section 2.1 presents the spatial econometric models, section 2.2 introduces the main concepts of data mining and spatial data mining.

2.1 Spatial regression models

In this section we present a general framework to incorporate spatial correlation structures into a linear regression model and the implication of this for estimation and specification testing. Early interest in the statistical implications of estimating spatial regression models dates back to the pioneering results of statistician Whittle(1954), followed by other important contributions, such as Besag(1974), Barlett (1963;1975), Ord (1975), and book by Ripley (1981). It was introduced in quantitative geography through the works of Cliff and Ord (1973, 1981) and Upton and Fingleton (1985). Paralleling this was the development of the field of spatial econometrics, started by regional scientists who were concerned with spatial correlation in multiregional econometrics models (Paelinck and klassen, 1979; Anselin, 1980). By the late 1980s and early 1990s, several works had appeared and the field had been given a clearer and more formal definition, for example, as outlined in the seminal book by Anselin (1988a), Griffith (1988) and Haining (1990), Cressie (1993). The book published by Luc Anselin in the late Eighties (Anselin, 1988a), certainly constitutes an important step forward in the historical development of the discipline. He defines the spatial econometrics as the collection of techniques that deal with the peculiarities caused by space in the statistical analysis of regional science models (Anselin, 1988a, p. 7). During the 1990s, the field also matured. Interest started to center on more rigorous formal proofs of the properties of estimators and test statistics (e.g., specialized laws of large numbers and central limit theorems were developed), new approaches were introduced (e.g., LM statistics, GMM estimation, Bayesian techniques), panel data and discrete choice models were considered, more attention was paid to computational aspects, and accessible software had become available. Spatial problems

also began to attract the attention of mainstream theoretical econometricians (such as Bera, Case, Conley, Kelejian, Pinkse, Prucha, Slade, among others) and papers started to appear in the leading econometric and field journals. In the 1992, Regional Science and Urban Economics published the first special issue of a journal totally devoted to spatial econometrics and by the end of the 1990s, important theoretical contributions had also appeared in the Journal of Econometrics and the International Economic Review, and spatial techniques had become so prevalent in real estate economics that a special issue of the Journal of Real Estate Finance and Economics was devoted to the topic (Pace et al. 1998). After around 2000, there was a seaside change with a tremendous increase in the number of both theoretical and applied papers dealing with spatial econometrics. Several journal special issues were devoted to the topic and articles appeared in all the major econometric and field journals (for a more detailed review, see Anselin, Florax and Rey, 2004). Spatial regression analysis is a core aspect of the spatial methodology toolbook and several text covering the state of art have appeared, such as Haining (2003), Waller and Gotway (2004), Banerjee et al. (2004), Fortin and Dale (2005), Schabengerger and Gotway (2005), and Arbia (2006). As argued in Arbia (2006), the integration of spatial methods with econometrics nevertheless remains in early phase and still lags far behind that experienced by time series methods in the Seventies after the path-breaking book by Box and Jenkins (1970). In fact, a discussion on spatial methods is absent in various introductory textbooks such as Baltagi (1999), Berndt (1991), Davidson (2000), Davidson and MacKinnon (1993), Goldberg (1998), Gourieroux and Montfort (1995), Greene (2003), Stock and Watson (2003), Thomas (1997), Verbeek (2000) and Woolridge (2002b). Exceptions are the books by Johnston (1991), Kmenta (1997), Maddala (2001), Baltagi (2001), Woolridge (2002a), Gujarati (2003) and Kennedy (2003).
Anselin(1988a) distinguishes the spatial effects in two broad classes: *spatial dependence* and *spatial heterogeneity*. Spatial dependence or spatial autocorrelation is best known and acknowledge most often, in particulary following the pathbreaking work of Cliff and Ord (1973). This dependence is related to the core of the disciplines of regional science and geography, as expressed in Tobler's (1970) first law of geography, in which "everything is related to everything else, but near things are more related than distant things". In general terms, spatial dependence can be considered to be the existence of a functional relationship between what happens at one point in space and what happens elsewhere. Spatial dependence can be caused by a variety of measurement problems such as the arbitrary delineation of spatial unit of observation, spatial aggregation and the presence of spatial externalities or spill-over effects. Spatial dependence is different with respect to the most common dependence of time series, indeed the multidirectional nature of dependence in space which is opposed to a clear one-directional situation in time, precludes the application of standard econometric methodology. Spatial heterogeneity is related to the structural instability in the form of non-constant error variances (heteroskedasticity) or model coefficients (variable coefficient, spatial regimes). There are many reasons why it is important to consider spatial heterogeneity explicitly. First, the "structure" behind the instability is spatial in the sense that the location of the observation is crucial in determining the form of the instability. Secondly, because the structure is spatial, heterogeneity often occurs jointly with spatial

autocorrelation and standard econometric techniques are no longer appropriate (see Anselin and Griffith, 1988). Thirdly in a single cross-section, spatial autocorrelation and spatial heterogeneity may be observationally equivalent: this requires that either aspects of the problem be structured very carefully to obtain identifiability of the model parameters. The focus is how one can to incorporate spatial effects into a linear regression models and the implication of this for estimation and specification test by considering the following general specification (Anselin,1988a):

$$y = \rho W_1 + X\beta + \epsilon \tag{2.1}$$
$$\epsilon = \lambda W_2 \epsilon + \mu \tag{2.2}$$

with $\mu \sim N(0, \Omega)$ and the diagonal elements of the error covariance matrix Ω as:
$\Omega_{ii} = h_i(z\alpha)$, $h_i > 0$.
In this specification, β is $K \times 1$ vector of parameters associated with exogenous variable X ($N \times K$ matrix), ρ is the coefficient of the spatially lagged dependent variable, and λ is the coefficient in a spatial autoregressive structure for the disturbance ϵ. The disturbance μ is normally distributed with a general diagonal covariance matrix Ω. The diagonal elements allow for heteroschedasticity as a function of $P + 1$ exogenous variables z, which include a constant term. The P parameters α are associated with nonconstant terms, such that, for $\alpha = 0$, it follows that $h = \sigma^2$ (the classic homoskedasticity situation). The two $N \times N$ matrices W_1, W_2 are standardized spatial weight matrices, associated with a spatial autoregressive process in the dependent variable and the disturbance term respectively. In total, the model has $3 + k + p$ unknown parameters, in vector form:

$$\theta = [\rho, \beta', \lambda, \sigma^2, \alpha']' \tag{2.3}$$

When subvectors of the parameter vector (3.2) are set to zero, specifically we have the following situations which correspond to four traditional spatial autoregressive models commonly discussed in the literature (see e.g. Hordijk, 1979; Anselin 1980, 1988a; Bivand 1984):
$1^o Case$: $\rho = 0$, $\lambda = 0$, $\alpha = 0$ (P+2 constraints):

$$y = X\beta + \epsilon \tag{2.4}$$

that is the classical linear regression model.
$2^o Case$: $\lambda = 0$, $\alpha = 0$ (P+1 constraints):

$$y = \rho W_1 y + X\beta + \epsilon \tag{2.5}$$

that is the mixed regressive spatial-autoregressive model: (**SAR** or Spatial Lag Model) (which includes the common factor specifications, i.e. with WX, as special case).
$3^o Case$: $\rho = 0, \alpha = 0$ (P+1 constraints):

$$y = X\beta + (I - \lambda W_2)^{-1}\mu \tag{2.6}$$

that is the linear regression model with a spatial autoregressive disturbance: **Spatial Error Model (SEM)**.
4^oCase: $\alpha = 0$ (P constraints):

$$y = \rho W_1 y + X\beta + (I - \lambda W_2)^{-1}\mu \tag{2.7}$$

that is the mixed regressive-spatial autoregressive model with a spatial autoregressive disturbance.
A variant of the spatial lag model that include spatially lagged independent variables is known as the **Spatial Durbin Model** (SDM, LeSage and Pace 2009):

$$y = \rho W y + X\beta + WX\lambda + \epsilon \tag{2.8}$$

where λ is the vector of coefficients for spatially lagged independent variables WX. The use of this model instead of the spatial lag model in (2.5) can potentially remove omitted variable bias, as discussed in detail in LeSage and Pace (2009). An alternative model with respect to SEM is the spatial moving average (SMA) model (Fingleton, 2008):

$$y = \rho W y + X\beta + (I + \lambda W)\mu \tag{2.9}$$

As can be observed by comparing (2.6) and (2.9), the spatial multiplier is not present in the SMA model. The SMA model is used to model localized effects. By its specification, spatial effects will affect only the first-order neighbors as defined by the weights matrix. In particular, this can been seen by considering the expanded form of $(I - \lambda W_2)^{-1}$.
A Leontief expression of the last matrix, under the assumption that $|\lambda| < 1$ is given by

$$(I - \lambda W_2)^{-1} = I + \lambda W + \lambda^2 W^2 + ... \tag{2.10}$$

As argued by Anselin (2003), the complete structure of the variance-covariance matrix then follows as the product of the (2.10) with its transpose, yielding a sum of terms containing matrix powers and products of W, scaled by powers of λ. Specifically the lowest order term is I, followed by λW and $\lambda W'$, $\lambda^2(W^2 + WW' + W^2)$ and so on. For a spatial weights matrix corresponding to first-order contiguity, each of the powers involves a higher order of contiguity, in effect creating band of every larger reach around each location, relating every location to every other one. Moreover the powers of the autoregressive parameter (with $|\lambda| < 1$) ensure that the covariance decreases with higher orders of contiguity.
Instead, the only diagonal non zero elements in the variance-covariance matrix are those corresponding to nonzero elements in W elements in W (or, equivalently, W') and W W'.
For W defined as first-order contiguity, such elements consist of location pairs that are first- and second-order neighbors, but no higher orders of contiguity. Consequently, the range of the effect of the spatial multiplier is much smaller than for a corresponding SAR model.
Several authors have suggested to combine spatial lag with spatial error dependence. The most general form is the spatial autoregressive, moving-average (SARMA) processes outlined by Huang (1984). Formally, a SARMA (p,q) process can be expressed as (Anselin and Bera, 1998)

$$y = \rho_1 W_1 y + \rho_2 W_2 y + ... + \rho_p W_p y + \epsilon \tag{2.11}$$
$$\epsilon = \lambda_1 W_1 \mu + \lambda_2 W_2 \mu + ... + \lambda_q W_q \mu + \epsilon \tag{2.12}$$

A different specification that combines spatial-autoregressive model with spatial-autoregressive disturbances is often referred to as a SARAR(p,q) model, see (Anselin and Florax, 1995). In modeling the outcome for each unit as dependent on a weighted average of the outcomes of other units, SARAR models determine outcomes simultaneously. Formally a SARAR (1,1) process can be expressed in (3.1). These various specifications are the most important to analyse global and local externalities in spatial econometric models (see Anselin, 2003).

2.1.1 Tests for spatial error and spatial lag dependence

This subsection present different specification of tests for spatial dependence in regression models: Moran's I and some Lagrange Multiplier (LM) tests: spatial error dependence; spatial lag dependence, second order spatial error dependence and for a first order spatial autoregressive moving average or SARMA process (Anselin and Florax, 1994). The most commonly used specification test for spatial autocorrelation is Moran's I, which is defined as:

$$I = [N/S](\mathbf{e'W_1 e}/\mathbf{e'e}) \tag{2.13}$$

where **e** is a R by 1 vector of OLS residuals, W is a spatial weight matrix, N is the number of observations and S is a standardization factor equal to the sum all elements in weight matrix. For a weight matrix that is normalized such that the row elements sum to one, expression (2.13) simplifies to:

$$I = (\mathbf{e'W_1 e}/\mathbf{e'e}) \tag{2.14}$$

The detailed moments are derived and discuss in Cliff and Ord (1972) and Anselin(1988a). It is important to note that, in contrast to the test based on the Lagrange Multiplier principle, Moran'I test does not have a direct correspondence with a particular alternative hypothesis. The second test, LM-ERR, is based on the Lagrange Multiplier principle and was originally suggest in Burridge (1980). The test is identical for spatial autoregressive and spatial moving average errors. It is defined as:

$$LM - ERR = (\mathbf{e'W_1 e}/\mathbf{s^2})^2/T_1 \tag{2.15}$$

where $s^2 = \mathbf{e'e}/R$, and $T_1 = tr(W_1'W_1 + W_1^2)$. This statistic is distributed as χ^2 with one degree of freedom.
The third test, K-R, is the robust large sample test suggested by Kelejian and Robinson (1992). This test does not assume normality, non linearity and is derived from an auxiliary regression using cross products of residuals of observations that are potentially spatially

correlated and cross product of the corresponding explanatory variable. In the auxiliary regression the dependent variable is:

$$C_h = e_i e_j \tag{2.16}$$

where h is an index for each cross product, e is a residual term and i, j are contiguous observations. The explanatory variables in the auxiliary regression, Z_h are formed as cross products of X_i and X_j. With γ as the coefficient vector obtained from OLS estimation in a regression of $\mathbf{C}$ on $\mathbf{Z}$, and α as the associated vector of residuals, the K-R statistic results as:

$$K - R = (\gamma' \mathbf{Z}' \mathbf{Z} \gamma)/(\alpha' \alpha/(h_R) \tag{2.17}$$

where h_R is the number of observations in the auxiliary vector (2.16). The statistic is distributed as χ^2 with K degrees of freedom, where K is the number of explanatory variables in $\mathbf{Z}$.
The corresponding robust to local misspecification in the form of spatial lag error is the fourth test LM-EL (Bera and Yoon, 1992), which is computed as:

$$LM - EL = [\mathbf{e'W_1 e}/\mathbf{s^2} - \mathbf{T_1}(R\tilde{J}_{p\beta})^{-1}(\mathbf{e'W_1 y}/\mathbf{s^2})]^2/[T_1 - {T_1}^2(R\tilde{J}_{p\beta})^{-1}] \tag{2.18}$$

$$(R\tilde{J}_{p\beta})^{-1} = [T_1 + (\mathbf{W_1 X}\beta)'\mathbf{M}(\mathbf{W_1 X}\beta)/s^2]^{-1} \tag{2.19}$$

where $\mathbf{W_1 X}\beta$ is a spatial lag of the predicted values from an OLS regression,
$\mathbf{M} = \mathbf{I} - \mathbf{X}(\mathbf{X'X})^{-1}\mathbf{X'}$ is the projection matrix. This statistic is also distributed as $\chi^2(1)$.
The fifth test is LM-ERR(2), which is implied to test a second order spatial dependence. In general as shown in Anselin(1994), test for higher order error dependence are simply the sum of the corresponding one-directional tests, distributed as χ^2 with degrees of freedom equal to the number of terms in the sum. The LM-ERR(2)can be expressed as:

$$LM - ERR(2) = (\mathbf{e'W_1 e}/\mathbf{s^2})^2/T_1 + (\mathbf{e'W_2 e}/\mathbf{s^2})^2/T_2 \tag{2.20}$$

with $T_2 = tr(\mathbf{W_2'W_2} + \mathbf{W_2^2})$.
To test the spatial lag dependence, we review the principal tests: LM-LAG, LM-LE, SARMA (Anselin and Florax, 1994).
LM-LAG is the Lagrange Multiplier test for spatial lag dependence of Anselin(1988b):

$$LM - LAG = (\mathbf{e'W_1 y}/\mathbf{s^2})^2/(R\tilde{J}_{p\beta}) \tag{2.21}$$

distributed as χ^2 with one degrees of freedom.
LM-LE is the counterpart of LM-EL, which can be used as a test for a spatial lag robust to local misspecification in the form of a spatial moving average error process by Bera and Yoon (1992). It is defined as:

$$LM - LE = (\mathbf{e'W_1 y}/\mathbf{s^2} - \mathbf{e'W_1 e}/\mathbf{s^2})^2/[R\tilde{J}_{p\beta} - T_1] \tag{2.22}$$

distributed as χ^2 with one degrees of freedom.
The final test is SARMA, is a Lagrange Multiplier test for a joint spatial lag and spatial moving average error and it is expressed as:

$$SARMA = (\mathbf{e'W_1y/s^2} - \mathbf{e'W_1e/s^2})^2/[R\tilde{J}_{p\beta} - T_1] + (\mathbf{e'W_1e/s^2})^2/T_1 \tag{2.23}$$

We note that the SARMA test is the sum of LM-LE (2.22) and LM-ERR (2.15).

2.2 Data mining and KDD

Several authors have observed that the term "data mining" has had a varied history (Fayyad, Piatetsky-Shapiro, and Smyth 1996; Smyth, 2000). It can be considered as a single step in the multi-step process of Knowledge Discovery in databases (KDD), where KDD is defined as the *"non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"* (Fayyad, Piatetsky-Shapiro, Smyth and Uthurusamy, 1996). The term process implies that KDD comprises many steps, which involve data preparation, search for patterns, knowledge evaluation, all repeated in multiple iterations. In alternatively, data mining is the *"process of extracting valid, previously unknown, comprehensible and actionable information from large database and using it to make crucial business decisions"*(Simoudis, 1996). In this case data mining, not KDD, is viewed as the overall process of extracting high-level knowledge from low-level data. Some authors underline the difficulty to isolate a core set of fundamental techniques that clearly distinguish data mining from any single component discipline: in some way it is a uniquely powerful combination of individual techniques from each discipline associated with analyzing massive data sets. In particular, it is a multidisciplinary field and it includes: *machine learning, statistics, database technology, high performance computing, data visualization, image processing* (Behnke and Dobinson, 2000). According to Weiss and Davison (2010) data mining can be considered a possible response to many problems like the *scalability* of traditional statistical techniques, which often cannot handle data sets with milion or billions of records and hundreds or thousands of variables; *highly unstructured* (non-numeric) data: text, audio, video, images. This data cannot easily be analyzed using traditional statistical techniques and the number of data analysts has not matched the exponential growth in the amount of data, which has caused much of this data to remain unanalyzed in a *"data tomb"* (Fayyad, 2003). In data mining the analyst does not need to make specific assumptions about the data nor formulate a specific hypothesis to test. The data mining process is typically **data-driven** and inductive rathen than hypothesis-driven or deductive process used by statisticians.
The data mining tasks can be categorized in **predictive** tasks and **descriptive** tasks (Weiss and Davison, 2010). The predictive tasks allow to predict the value of a variable based on other existing information, while the descriptive tasks summarize the data in some manner. We briefly describe the principal predictive and descriptive data mining tasks. **Classification and regression** tasks are predictive tasks that involve building a model to predict a target, or dependent variable, from a set of explanatory or independent variables. **Associ-**

ation rule analysis is a descriptive data mining task that involves discovering patterns, or associations, between elements in a data set. The associations are represented in the form of rules, or implications. The most common association rule task is *market basket analysis*. **Cluster analysis** is a descriptive data mining task where the goal is to group similar objects in the same cluster and dissimilar objects in different clusters. **Text mining**: the unstructured nature of text require special consideration. Example applications of text mining includes the identification of specific noun phrases such as people, products and companies, which can then be used in more sophisticated co-occurrence analysis to find nonobvious relationships among people or organizations. A second application area that is growing in importance is sentiment analysis, in which blogs, discussion boards, and reviews are analyzed for opinions about products or brands. **Link Analysis**: is a form of network analysis that examines associations between objects. For example, given a graph showing relationships between objects, link analysis can find particularly important or well-connected objects and show where networks may be weak (e.g., in which all paths go through one or a small number of objects).
According to Mitra et al. (2002) the main challenges in the data mining procedure are: *massive data sets and high dimensionality* (huge data sets increases the size of the space of patterns); *user interaction and prior knowledge*: data mining is inherently an interactive and iterative process; *overfitting and assessing the statistical significance*: regularization and resampling methodologies need to be emphasized for model design; *understandability of patterns*: rule structuring, natural language representation, and the visualization of data and knowledge; *nonstandard and incomplete data*: the data can be missing and/or noisy; *mixed media data*: learning from data that is represented by a combination of various data (media, like numeric, symbolic, images and text); *management of changing data and knowledge*: rapidly changing data (nonstationary), in a database that is modified, deleted, augmented, may make previously discovered patterns invalid (incremental methods for updating the patterns); *integration*: data mining tools are often only a part of the entire decision making system.

2.2.1 CART: classification and regression trees

We briefly recall some general background on Classification and Regression Trees (CART). Classification and regression tree has been an important data mining methodology for the analysis of large data sets via binary partitioning procedure (Breiman et al., 1984). It consists in recursive division of N cases on which a response variable and a set of predictors are observed. Such a partitioning procedure is known as **regression tree** when the response variable is continuously valued and as a **classification tree** when the response variable is categorical. A classification tree procedure provides not only a classification rule for new cases of unknown class, but also an analysis of the dependence structure in large data sets. Figure 1 depicts a simple tree structure with tree layers of nodes.

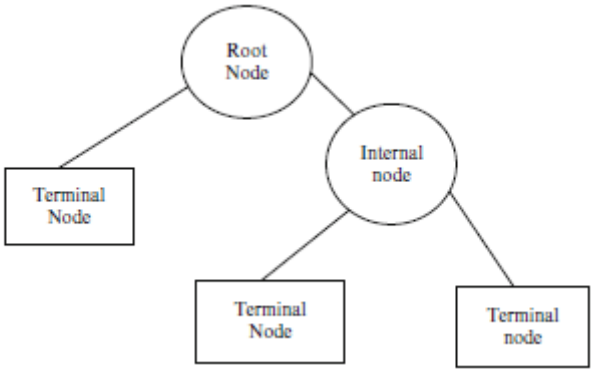


Figure 2.1: A simple tree structure (Y. Leung, 2010).

The root node contains the entire learning sample, and the other nodes correspond to subgroups of the learning sample. The two subgroups in the left and right offspring nodes are disjoint, and their union comprises the subgroups for the parent node. A critical step of the tree-based technique is to determine the split from one parent node to two offspring nodes. Let (X,Y) be a multivariate random variable where X is the predictor vector $(X_1 , \ldots, X_m, \ldots X_M)$ where $X_1 , \ldots, X_m, \ldots X_M$ can be a mixture of ordered and categorical variable: and Y is the criterion variable taking values in the set of prior classes $\Gamma = \{ 1, \ldots, j, \ldots, J \}$.
Four elements are needed in the classification tree growing procedure (Leung, 2010):

1. A set of binary questions of the form $\{ is\ X \in A? \}$.

2. A goodness of split criterion $\triangle i\ (s \mid t)$ that can be evaluated for any split s of any node t.

3. A splitting termination rule.

4. A rule for assigning every terminal node to a class.

For each ordered variable X_m, all questions in a set of binary questions are of the form {is $X_m \leq$ c?} for c ranging over (-∞, ∞).
If X_m is categorical taking values, say, in $\{b_1 , b_2, \ldots b_u \}$, then all questions in a set of binary questions are questions of the form $\{is\ X \in s?\}$, as s ranges over all nontrivial subset of $\{b_1 , b_2, \ldots b_u \}$.
The set of binary questions generates a set Q of splits s of every node t. For those cases in t answering "yes" to a question will go to the left descendant node $\{t_L\}$ and those answering "no" will go to the right descendant node $\{t_R\}$. The goodness of split is measured by an impurity function defined for each node. Intuitively, we want each leaf node to be "pure", that is, one class dominates.

Definition 1. *(Breiman et al., 1984) : An* ***impurity function*** *is a function ϕ defined on the set all J-tuples of numbers (p_1,..., p_J) satisfying p_j =0, j=1, . .., J, $\sum_j p_j = 1$ with the properties*

1. *ϕ is a maximum only at the point ($\frac{1}{j}$, $\frac{1}{j}$, ..., $\frac{1}{j}$)*
2. *ϕ achieves its minimum only at the point (1,0 , ..., 0), (0,1,...,0), ..., (0,0,..,0,1)*
3. *ϕ is a symmetric function of p_1,..., p_J.*

Definition 2. *(Breiman et al., 1984): Given an impurity function ϕ, define the* ***impurity measure*** *$i(t)$ of any node t as*

$$i(t) = \phi\, p(1 \mid t),\ p(2 \mid t),\ \ldots,\ p(J|t)$$

if a split s of a note t sends a proportion p_R of the data cases in t to $t_R$and proportion $p_L$to t_L , define the decrease in impurity to be

$$\triangle\ i\ (s,\ t) = i(t) - p_R\ i(t_R) - p_L\ i(t_L)$$

wherep_R andp_L are the proportions of the samples in node t that go to the right nodet_R and the left node t_L respectively.

Definition 3. *(Breiman et al., 1984): the* ***Tree impurity*** *I (T) is defined by*

$$I(T) = \sum_{t \in \tilde{T}} I(t) = \sum_{t \in \tilde{T}} i(t)\ \ p(t)$$

where $\tilde{T}$ denote the set of terminal node.

The most popular splitting rules are the entropy and the Gini index. The **entropy index** is

$$H(t) = \sum_j p(j \mid t) \log\{\ p(j \mid t)\}$$

The **Gini index** is

$$D(t) = \sum_{j=1}^{J} p(j \mid t)\, p(i \mid t) = 1 - \sum_{j=1}^{J} p^2(j \mid t)$$

Both indices are equal to 0 when there is only class present in leaf t and maximum when all classes are present equal probabilities.
Let be (X^1, ..., X^p, Y) an independent sample of random variables, where $X^k$is the explanatory variables and Y is categorical variable to explained. The final tree overfits the available data and the prediction error $R(T) = P\{\ T\ (X^1,\ \ldots,\ X^p) \neq Y\}$is typically large. In designing a classification tree, the ultimate goal is to produce from the data a tree T whose

probability of prediction error $R(T)$ is as small as possible. Thus, in second stage the tree T is "pruned" to produce a subtree $T^{'}$ whose expected performance is superior to $R(T)$. When the data are independent samples the proportion $p(j \mid t)$ is estimated by $\widehat{p}\ (j \mid t) = n_{jt}/n_t$, where n_{jt} is the number of samples in leaf t that are in class j, and n_t is the total number of samples in leaf t.
The mechanism of the regression tree is similat to the classification tree. In a tree structured predictor the space X is partitioned by a sequence of binary splits into terminal nodes. In each terminal node t, the predicted response value $y(t)$ is constant. Starting with a learning sample $\mathcal{L}$, three elements are necessary to determine a tree predictor:

1. a way to select a split at every intermediate node;
2. a rule for determining when a node is terminal;
3. a rule for assign a value $y(t)$ to every terminal node t.

It is therefore necessary first to define a criterion of accuracy of the rule prediction; to this end it is typically used the **Mean squared error R(d)** of the predictor **d** that can be estimated according to following criterion.

Definition 4. *(Breiman et al., 1984) Define the mean squared error $R^*(d)$ of the predictor d as*

$$R^*(d) = E(Y - d(\mathbf{X}))^2 \tag{2.24}$$

where: $R^*(d)$ is the expected squared error using $d(\mathbf{X})$, $d(\mathbf{X})$ is a predictor of Y, $\mathbf{X} = (\mathbf{x_1}, ..., \mathbf{x_N})$.
The optimal predictor has a simple form:

Proposition 1. *(Breiman et al., 1984) The predictor d_B which minimizes $R^*(d)$ is*

$$d_B(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x}) \tag{2.25}$$

$d_B(\mathbf{x})$ is the conditional expectation of the response, given that the measurement vector is $\mathbf{x}$.

Given a learning sample $\mathfrak{L}$ consisting of $(\mathbf{x_1}, y_1)..,(\mathbf{x_n}, y_n), ...(\mathbf{x_N}, y_N)$ to construct a predictor d($\mathbf{x}$) and to estimate its MSE $R^*(d)$, if we use as accuracy criterion the **resubstitution estimate** for $R^*(d)$ we have:

$$R(d) = \frac{1}{N}\sum_{n=1}^{N}(y_n - d(\mathbf{x}_n))^2 \tag{2.26}$$

as the optimal predictor $y(t)$ that minimizes $R(d)$.

Proposition 2. *(Breiman et al., 1984) The value of $y(t)$ that minimizes $R(d)$ is the average of y_n for all cases $(\mathbf{x}_n, y_n)$ falling into t; that is, the minimizing $y(t)$ is*

$$\bar{y}(t) = \frac{1}{N(t)}\sum_{\mathbf{x}_n \in t} y_n \tag{2.27}$$

where the sum is over all y_n such that $\mathbf{x}_n \in t$ and $N(t)$ is the total number of cases in t.

So the problem of assigning a value to each node is solved by replacing the values in the node with their arithmetic mean, which represents the best forecast if you choose to resubstitution estimate of $R(d)$ as a measure of the accuracy of predictor.
If the optimal $\bar{y}(t)$ (2.27) represents the prediction of Y for node t and by using the notation $R(T)$ instead $R(d)$, where T is a generic regression tree we define

$$R(t) = \frac{1}{N} \sum_{\mathbf{x}_n \in t} (y_n - \bar{y}(t))^2 \tag{2.28}$$

and

$$R(T) = \sum_{t \in \tilde{T}} R(t) \tag{2.29}$$

where $\tilde{T}$ is the set of terminal nodes of T.
So that

$$R(T) = \frac{1}{N} \sum_{t \in \tilde{T}} \sum_{\mathbf{x}_n \in t} (y_n - \bar{y}(t))^2 \tag{2.30}$$

where for every node t, $\sum_{\mathbf{x}_n \in t} (y_n - \bar{y}(t))^2$ is the *within node* of squares and it is the total squared deviations of the y_n in t from their average. By summing over t $\in \tilde{T}$ one obtains the total within node sum of squares, and dividing by N one provides the average. Given any set $\mathcal{S}$ of splits of a current terminal node t in $\tilde{T}$,

Definition 5. *(Breiman et al., 1984) The best split s^* of t is that split in $\mathcal{S}$ which produces the largest reduction of $R(T)$. More precisely, for any split s of node t into t_L and t_R, let*

$$\Delta R(s,t) = R(t) - R(t_L) - R(t_R) \tag{2.31}$$

Take the best split s to be a split such that

$$\Delta R(s^*,t) = \max_{s \in \mathcal{S}} \Delta R(s,t) \tag{2.32}$$

Thus, a regression tree is constructed iteratively dividing the nodes in order to produce the maximum decrease of R(T). This criterion identifies the breakdown threshold of the space of explanatory variables that most effectively separates the high response values from the low ones.
Let us defined the tree thus obtained as T_{max}. To select the optimal sequence we consider the cost-complexity *pruning*.

2.2.2 Pruning

The pruning techniques present in this subsection follow the approach used in system CART, by proceding in two separate stages, where initially a sequence of alternative pruned trees is generated and then a tree selection process is carried out to obtain the final model. There are two main strategies:

1. Pre-pruning stops growing the tree during the learning, before it reaches the point where it perfectly classifies the learning data set. The generic decision tree learning algorithm learner continues splitting the nodes as long as there is an attribute to split and/or the data is not classified perfectly. Adding to the general terminal conditions a node will not be split with the pruning if

   - the number of instances matching to the node is too small (N_t);
   - the impurity of the split on the node (I_t) is low enough;
   - the best test is not statistically significant (according to some statistical test). The main concern about the pre-pruning approach is that the optimum values of the parameters (N_t, I_t, significance level) are not only problem dependent but they can also differ for the different branches of the same tree.

2. Post-pruning allows the tree to over-fit and then prunes the tree later. This approach needs a test to be able to evaluate the generalization error. Unless there is test set available the learning data set L is split into two sets: training set L_T to build the tree and validation set L_V. After building a complete tree T_0 from the training set L_T a sequence of trees $T_1, T_2, ...$ are computed by removing some subsets of nodes from the initial tree T_0. In the sequence each tree T_i is obtained by removing some subtree from the previous tree T_{i1}. At the end the best tree T_i, with the minimum validation error on L_V is selected from the sequence. Post-pruning is the most common strategy: it is the process by which a large tree is grown and the reliable evalutation methods are used to select the "right-sized" pruned tree of this initial model.

CART (Breiman et al., 1984) prunes a large regression tree using two stage algortihm called **Error-complexity pruning** (or minimal cost-complexity for classification tree).

Definition 6. *(Breiman et al., 1984) For any subtree $T \leq T_{max}$, define its **complexity** as $|\tilde{T}|$, the number of terminal nodes in T. Let $\alpha \geq 0$ be a real number called the **complexity parameter** and define the cost-complexity measure $R_\alpha(T)$ as*

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}| \tag{2.33}$$

For each value of α, find that subtree $T(\alpha) \preceq T_{max}$ which minimizes $R_\alpha(T)$:

$$R_\alpha(T(\alpha)) = \min_{T \preceq T_{max}} R_\alpha(T) \tag{2.34}$$

The result is a decreasing sequence of trees $T_1 > T_2 > ...> \{t_1\}$ with $T_1 \preceq T_{max}$ and a corresponding increasing sequence of α values $0 = \alpha_1 < \alpha_2 < ...$ such that for $\alpha_k \leq \alpha < \alpha_{k+1}$, where $k = 1, ..., K$ and T_k is the smallest subtree of T_{max} minimizing $R_\alpha(T)$.
This sequence is obtained by minimizing the following function

$$g(t) = \frac{E_e}{N(|\tilde{T}_t|) - 1} \tag{2.35}$$

where N denote the total number of training instance and the pruned subtree has E_e more misclassified training instance than the starting tree. The heart of this process is to calculate each value of α, relative to each pruned subtree. The key to calculating each value of α is to understand that it works by *weakest-link cutting* (Breiman et al., 1984).

Weakest-link cutting works by considering the *weakest-link* $\bar{t}_1$ as the node such that $g(\bar{t}_1) = \min_{t \in T} g(t)$.
Then the value of $g(\bar{t}_1)$ is the value of complexity parameter for the pruned subtree $T_1 - T_{\bar{t}_1}$ and denoted as α_2. The pruned subtree $T_1 - T_{\bar{t}_1}$ is denoted T_2. Now, by using T_2 as a starting tree instead of T_1, the next step is to find $\bar{t}_2$ in T_2 by *weakest-link cutting* and calculate the corresponding value of α α_3. This process is similar to the previous calculation of $\bar{t}_1$. After this step, the second pruned subtree T_3, $T_2 - T_{\bar{t}_2}$ is obtained. This process is repeated recursively until the pruned subtree to be pruned only has the root node. Thus, a decreasing sequence of pruned subtrees $T_1 > T_2 > ... > T_n$ and an increasing sequence of their corresponding values $\alpha_1 < \alpha_2 < ...$ such that for α_n are formed. T_n stands for the pruned subtree from T_1 that only has the root node, α_n stands for the value corresponding to T_n. Because T_1 is an unpruned tree, its corresponding value of α_1 is 0. In summary, this is the relative algorithm:

MINIMAL COST-COMPLEXITY PRUNING($\mathbf{X}, k$)

1 **Input:**
$\mathbf{X}$: a set of N labeled instances
k : maximum number of trees
2 $T_{max} \leftarrow$ LEARNER($\mathbf{X}$, **attrs**, **0**)
3 $T_1 = T(\alpha_{min})$ where $\alpha_{min} = 0$:
$R(T_1) = R(T_{max})$
4 **for** $i \leftarrow 1$ **to** k
do
5 **for** $\forall t \in T_i$ $g_i(t) = \begin{cases} \frac{R(t)-R(T_i)}{|\tilde{T}_i|-1}, & \text{if } t \notin \tilde{T}_i \\ \infty, & \text{else} \end{cases}$
6 Choose the weakest link $\bar{t}_i$:
$\bar{t}_i = \arg\min_{t \in T_i} g_i(t)$
$g_i(\bar{t}_i) = \min_{t \in T_i} g_i(t)$
and the set of weakest links $\{\bar{t}_i\}$
$\{\bar{t}_i\} = \{\bar{t}_i' : g_i(\bar{t}_i) = g_i(\bar{t}_i')\}$
7 $\alpha_{i+1} \leftarrow g_i(\bar{t}_i)$
8 $T_{i+1} \leftarrow T_i - T_{\bar{t}_i}, \forall \bar{t}_i \in \{\bar{t}_i\}$
9 **return** *The Sequence of Pruned Trees and their Complexity parameters*
$T_1 \succ T_2 \succ \ldots \succ T_k$ and $\{\alpha_i\} : \alpha_i < \alpha_{i+1}, k \geq 1, \alpha_1 = 0$
$T(\alpha_i) = T_i$, for $i \geq 1$, $\alpha_i \leq \alpha < \alpha_{i+1}$.

Figure 2.2: Minimal cost-complexity pruning (Breiman et al.,1984)

The problem of minimal cost-complexity pruning is now reduced to selecting which pruned subtree is optimum from the sequence. This can be done by cross-validation.
To select the right sized tree from the sequence $T_1 > T_2 > ...$ estimates of $R(T_k)$ are needed. Let us randomly divided $\mathfrak{L}$ into V-fold cross validation $\mathfrak{L}_1, ..., \mathfrak{L}_V$ such that each sub sample $\mathfrak{L}_v$, $v = 1, ..., V$, has the same number of cases (as nearly as possible).
For each v, this produces the trees $T^{(v)}(\alpha)$ which are the minimal error-complexity trees for the parameter value α. Grow and prune using all of $\mathfrak{L}$, getting the sequence $\{T_k\}$ and $\{\alpha_k\}$. The **cross-validation estimates** are given by

$$R^{CV}(T_k) = \frac{1}{N}\sum_{v=1}^{V}\sum_{(x_n,y_n)\in\mathfrak{L}_V}(y_n - d_k^{(v)}(\mathbf{x_n}))^2 \tag{2.36}$$

and the corresponding relative error estimate

$$RE^{CV}(T_k) = R^{CV}(T_k)/R(\bar{y}) \tag{2.37}$$

$$R(\bar{y}) = \frac{1}{N}\sum_{n=1}(y_n - \bar{y})^2 \tag{2.38}$$

where $d_k^{(v)}(\mathbf{x})$ is the predictor corresponding to the tree $T^{(v)}(\alpha'_k)$ with $(\alpha'_k) = \sqrt{\alpha_k \alpha_{k+1}}$. The tree selected is T_K where K is the maximum k such that

$$R^{CV}(T_k) \leq R^{CV}(T_{k_0}) + SE \tag{2.39}$$

where

$$R^{CV}(T_{k_0}) = \min_k R^{CV}(T_k) \tag{2.40}$$

It is called **1-SE rule**.
In conclusion, the tree structured approach presents many advantages: it needs of only a few elements: the set of questions, a rule for selecting the best split at any node, a criterion for choosing the right-sized tree; it a powerful and flexible classification tool: it can be applied to any data structured and the final classification has a simple form which can be compactly stored and that efficiently classifies new data; it makes powerful use of conditional information in handling nonhomogeneous relationships; it does automatic stepwise variable selection and complexity reduction; it gives not only the predicted classification but also it estimates the misclassification probability for the object; it is invariant under all monotone transformations of individual ordered variables; it is extremely robust with respect to outliers and misclassified point in the sample; it provides easily understood and interpreted information regarding the predictive structure of the data.

2.2.3 Boosting, bagging and Random Forests

Instead of recursively partitioning smaller and smaller portions of the data set like CART, boosting considers the full data set at each potential partitioning node. The motivation for boosting was a procedure that combines the outputs of many weak classifiers to produce a powerful committee. Given $Y \in \{-1, 1\}$ and a vector of predictor variables X, a classifier $G(X)$produces a prediction taking on of the two values $\{-1, 1\}$. The aim of the boosting algorithm is sequentially apply the weak classification algorithm to repeatedly modified versions of the data, producing a sequence of weak classifiers $G_m(x), m = 1, 2, ..., M$..

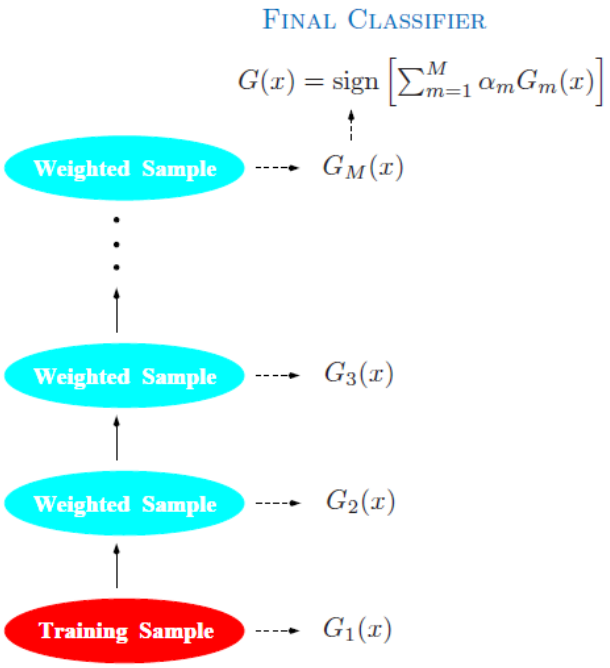


Figure 2.3: Schematic of AdaBoost. Classifiers are trained on weighted versions of the dataset, and the combined to produce final prediction (Hastie, T. et al.,2009)

The following figure shows the details of of the *AdaBoost.M1 algorithm*.

1. Initialize the observation weights, $w_i = 1/N, i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   a. Fit a classifier $G_m(x)$ on the training data using weights w_i.
   b. Compute $err_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x))}{\sum_{i=1}^{N} w_i}$.
   c. Compute $\alpha_m = log(1 - err_m/err_m)$.
   d. Set $w_i \leftarrow w_i \cdot exp[\alpha_m \cdot I(y_i \neq G_m(x_i))], i = 1, 2, \ldots, N$.
3. Output $G(x) = \texttt{sign}[\sum_{m=1}^{M} \alpha_m G_m(x)]$.

Figure 2.4: AdaBoost.M1 algorithm (Hastie, T. et al.,2009)

Boosting is used as "base learner" $G(x)$ in classification tree. Boosting also does not perform well when there is a large amount of classification noise. It results less attractive to for statisticians than bagging or Random Forests, in part, because, it lacks consistency and there is not imply convergency. Bagging, or bootstrap aggregation, is an ensemble method that uses a bootstrap sample of the hold out data to train predictors and then combines results from several fitting attempts, assigning a predicted class or value for each observation. **Bagging** is one of the earliest methods to combine"random tree" and provides a key step in

the development of Random Forests. Let Ω be an original dataset, divided into a training Ω_{train} and Ω_{test} and let B a series of bootstrap samples from Ω the bagging algorithm is the following:

1. From an original data set, Ω:
   a. Take B bootstrap samples from the training data, Ω_{train}
   b. Aggregate the collection of bootstrap samples: $\sum_i^B (B_i)$
2. Train predictors using these bootstrapped and aggregated (bagged) data by growing many trees without pruning and then counting the number of times (over trees) that each case is classified in each category.
3. Combine predictors using majority voting (for classification) or averaging (for regression) over the set of trees and assign cases accordingly.

Figure 2.5: Bagging algorithm, (Hastie, T. et al.,2009)

Random Forests is a substantial modification of bagging that provides a classifier consisting of a collection of tree-structured classifiers. So, the result is a forest constructed from randomly selected cases and randolmy selected predictors. In particular, the algorithm is given as follows:

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

Figure 2.6: Random Forest for Regression or Classification, (Hastie, T. et al., 2009)

The idea in Random Forests is to improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the variance too much. This is allowed by the random selection of the input variables in the tree-growing process. Furthermore, variables in out-of-bag samples are randomly permuted and then their impact on the test set error is measured as "variable importance"(for more details see Breiman, 2001).

2.3 Spatial data mining and spatial data structures

In this research the possible use of spatial data mining (SDM) methods is investigated for integrating by traditional spatial statistics and econometrics. The research is based on a literature survey which identifies the core concepts and methods of SDM. This background is a starting point for further theoretical and conceptual analysis. Spatial analyitical methods traditionally were developed for exploring geospatial data are focused on small datasets. To discover new unexpected patterns, trends and relationships that can be hidden in very large and heterogeneous geospatial datasets, data mining plays as an appropriate tool for extracting patterns from large geospatial databases. Let us start introducing the following definition:

Definition 7. *Spatial data mining and knowledge discovery (SDMKD) is the efficient extraction of hidden, implicit, interesting, previously unknown, potentially useful, ultimately understandable, spatial or non-spatial knowledge (rules, regularities, patterns, constraints) from incomplete, noisy, fuzzy, random and practical data in large spatial databases (Deren and Shuliang, 2005).*

A spatial pattern expresses a spatial relationship among spatial objects and to extract spatial patterns from spatial data sets it is important to identify the relevant spatial objects and the properties of, and relationships between, relevant spatial objects (Malerba, 2007). We observe three principal differences with respect to *classical data mining*. First, classical data mining treats each input as independent of other inputs, whereas spatial patterns often must satisfy the constraints of continuity and high autocorrelation among nearby features. This characteristics is called spatial autocorrelation in spatial statistics. It is diffent from the spatial heterogenity that refers to the non-stationarity of most geographic processes vary by location and that it is not possible to describe the phenomenon well at any location using a global estimate of parameters. Second, classical data mining deals with numbers and categories. In contrast, spatial data is more complex and includes extended objects such as points, lines, and polygons. Spatial objects have a geometry which need to be represented. In spatial data bases, object of the same type are organized in layers, each of which can have its own set of attributes and at most one geometry attribute. Third, classical data mining works with explicit inputs, whereas spatial predicates (e.g., overlap) and attributes (e.g., distance, spatial autocorrelation) are often implicit. Spatial objects have a locational property which implicitly defines spatial relationships between objects: topological, distance and direction relations.
SDM is a confluence of databases technology, artificial intelligence, machine learning, probabilistic statistics, visualization, information science, pattern recognition and other disciplines. The specificity of SDM lies in its interaction with space. In effect, a geographical database constitutes a spatio-temporal continuum in which properties concerning a particular place are generally linked and explained in terms of the properties of its neighborhood. We can thus see the great importance of spatial relationships in the analysis process. Temporal aspects for spatial data are also a central point but are rarely taken into account (Zeitouni, 2000).

It is necessary to develop new methods that consider the huge volume of data (e.g. enconding geometric location), the time consuming and the complexity of spatial relationships and spatial data handling. Basic tasks of spatial data mining are: *a) spatial classification*: finds a set of rules which determine the class of the classified object according to its attributes; *b) spatial regression or prediction model*: the response attribute depends on the attribute values of objects spatially-related to the object to be predicted; *c) spatial association rules*: find (spatially related) rules from the database. Association rules describe patterns, which are often in the database. The association rule has the following form: A $\longrightarrow$ B(s%; c%), where "s" is the *support* of the rule (the probability, that A and B hold together in all the possible cases) and "c" is the *confidence* (the conditional probability that B is true under the condition of A); *d) spatial clustering*: groups the object from database into clusters in such a way that object in one cluster are similar and objects from different clusters are dissimilar (*partitioning method, hierarchical method, density based method and grid-based method*); *e) spatial trend detection*: finds trends in database. A trend is a temporal pattern in some time series data. A spatial trend is defined as a pattern of change of a non-spatial attribute in the neighborhood of a spatial object.
Finally, to extract knowledge from spatial data therefore requires special approaches. A first possible approach is to invent new spatially aware algorithms that are particulary adapted to explore spatial data. A second approach is to explicitly model spatial properties and relationships in the pre-processing step and then use classical data mining algorithms.

2.3.1 A brief review some of the existing algorithms for clustering high-dimensionalty data: density-based algorithms for clusters discovering (DBSCAN) and spectral clustering

2.3.2 A Density based notion of clusters

The key idea of a density-based cluster is that for each point of a cluster its neighborhood for some given radius has to contain at least a minimum number of points, i.e. the "density" in the Eps-neighborhood of points has to exceed some threshold (Ester et al. 1996). This idea is illustrated by the sample sets of points depicted in Figure 6.

Figure 2.7: Simple databases (M. Ester, H.-P. Kriegel, J. Sander,1996)

In these examples, we can easily and unambiguously detect clusters of points and noise points not belonging to any of those clusters, mainly because we have a typical density of points inside the clusters which is considerably higher than outside of the clusters. Furthermore, the density within the areas of noise is lower than the density in any of the clusters.
In the following, we formalize the intuitive notion of "clusters" and "noise" in a database D of point of k-dimensional space S.
The shape of a neighborhood is determined by the choice of a distance function for two points p and q, denote by $dist(p, q)$. DBSCAN is a density based algorithm which discovers clusters with arbitrary shape and with minimal number of input parameters. The input parameters required for this algorithm is the radius of the cluster (Eps) and minimum points required inside the cluster (MinPts).
The following definitions are the key concepts of the DBSCAN algorithm.

Definition 8. *(M. Ester, H.-P. Kriegel, J. Sander,1996) : The* ***Eps-neighborhood*** *of a point p, denote by N_{Eps} (p) is defined by* $\mathrm{N_{Eps}}$ *(p)=* $\{q \in D | \, dist \ (p, q) \leq Eps\}$.

A naiive approach could require for each point in a cluster that there are at least a minimum number (*MinPts*) of points in an Eps-neighborhood of that point. This approach fails because there are two kinds of points in the cluster, the points which is inside the cluster (***core points***), and points on the border of the cluster (***border points***). In general, for every point p in a cluster C there is a point q in C so that p is inside of the Eps-neighborhood of q and N_{Eps} contains at least *MinPt*s points. This concept is elaborated in the following definition.

Definition 9. *(M. Ester, H.-P. Kriegel, J. Sander, 1996): A point p is* ***directly density-reachable*** *from a point q with respect to Eps, MinPts if*

1. $p \in$ $\mathrm{N_{Eps}}$ *(q) and*
2. [$\mathrm{N_{Eps}}$ *(q)] = Minpts (****core point condition****).*

Obviously, directly density-reachable is symmetric for pairs of core points. In general, however, it is not symmetric if one core point and one border point are involved.
Figure 7 shows the asymmetric case.

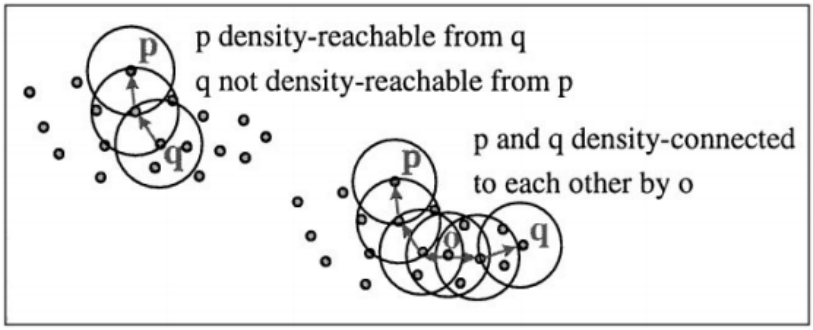


Figure 2.8: Core points and border points (M. Ester, H.-P. Kriegel, J. Sander,1996)

Definition 10. *(M. Ester, H.-P. Kriegel, J. Sander, 1996): A point p is* ***density-reachable*** *from a point q with respect to Eps and MinPts if there is a chain of points* $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ *such that* p_{i+1} *is directly density-reachable from* p_i.

Density-reachability is a canonical extension of direct density-reachability. This relation is transitive, but it is not symmetric.

Definition 11. *(M. Ester, H.-P. Kriegel, J. Sander, 1996): A point p is* ***density-connected*** *to a point q with respect to Eps and MinPts if there is a point o such that both, p and q are density-reachable from o with respect to Eps and MinPts.*

This relation is a symmetric relation and for density reachable points it is also reflexive.

Definition 12. *(M. Ester, H.-P. Kriegel, J. Sander, 1996): Let D be a database of points. A* ***cluster*** *C with respect to Eps and MinPts is a non-empty subset of D satisfying the following conditions:*

1. $\forall$ *p, q: if* $p \in C$ *and q is density-reachable from p wrt. Eps and MinPts, then* $q \in C$ *(****Maximality****);*

2. $\forall$ *p, q if* $p \in C$ *: p is density-connected to q wrt. Eps and MinPts (****Connectivity****).*

Definition 13. *(M. Ester, H.-P. Kriegel, J. Sander, 1996): Let* $C_1, \ldots, C_k$ *be the clusters of the database D with respect to parameters* Eps_i *and* $MinPts_i$, *i=1, . . . , k. Then we define the* ***noise*** *as the set of points in the database D not belonging to any cluster* C_i, *i.e. noise=*$\{p \in D \mid \forall\ i:\ p \notin C_i\ )$.

In this subsection, we describe the algorithm DBSCAN (Density Based Spatial Clustering of Applications with noise) that is designed to discover the spatial data clusters with noise. The steps involved in this algorithm are as follows: (i) select an arbitrary point; (ii) retrieve all points density-reachable from p with respect to *Eps* and *MinPts*; (iii) if p is a core point, a cluster is formed; (iv) if p is a border point, no points are density reachable from p and DBSCAN visits the next point of the database; (v) continue the process until all the points have been processed.
DBSCAN requires two input parameters (Minimum points and radius) and supports the user in finding an approximate value for it using *k-dist graph*.
Let d be the distance of a point p to its k-th nearest neighbor, then the d-neighborhood of p contains exactly k+1 points for almost all points p. The d-neighborhood of p contains more than k+1 points only if several points have exactly the same distance d from p which is quite unlikely.
The *k- dist* approach looks at the behavior of the distance from a point to its *kth* nearest neighbor. If *k* is not larger than the cluster size, the value of *k - dist* is small for points that belong to the same cluster. The *k- dist* for points not in the cluster is relatively large. The idea is to pick a value of *k* to be the *MinPts*. The following steps are performed to find the value of *k*:

1. Compute the *k- dist*, (distance to its *kth* nearest neighbor) for each of the data points.

2. Sort *k- dist* measures in increasing order.

3. Plot the sorted *k-dist* values: this graph is called the sorted *k-dist graph*. We expect to see a sharp change at the value of k-*dist (threshold point)* that corresponds to a suitable value of *Eps*. If we select this distance as the *Eps* parameter and take value of *k* as the *MinPts* parameter, then points for which k-dist is less than *Eps* will be labeled as core points, while other points will be labeled or border points.

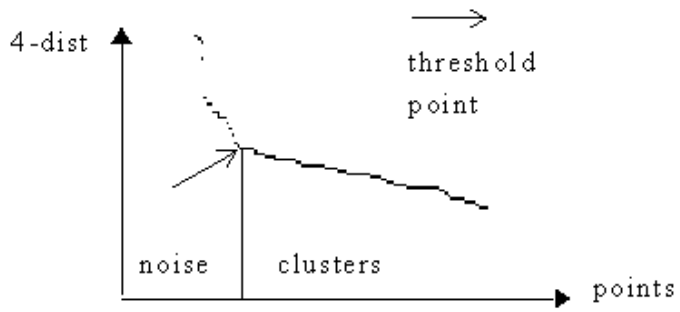


Figure 2.9: Sorted 4-dist grapgh for sample database 3 (M. Ester, H.-P. Kriegel, J. Sander,1996)

Finally we observe that DBSCAN is very robust to outliers but it very sensible to the choise of values of these parameters (Eps, MinPts) and it is highly affected by the distance measure used in finding the distance between two points.
The algorithm fails to identify clusters if density varies and if the data set is too sparse.
It is very important underline the improvement performance of DBSCAN when we use spatial structures as Kd-trees.
The use of the Kd-tree data structure enables efficient computation of the k-nearest neighbours (k-NN) of a pattern point, particularly for large data. (Sushmita Mitra, Jay Nandy, 2011).
The basic time complexity of the DBSCAN algorithm is *O(m*time to find points in the Eps-neighborhood)* where m is the number of points. In the worst case time complexity of DBSCAN algorithm is $O(m^2)$. However, in low dimensional data, this time complexity can be reduced to O(m log m) using Kd- trees, that allow efficient retrieval of all points within a given distance of a specific point. The space requirement of DBSCAN, even high-dimensional data, is *O(m)* because it is only necessary to keep a small amount of data for each point i.e. the cluster label and the identification of each point as a core, or noise point.

2.3.3 Spectral Clustering

In the current subsection, we briefly introduce spectral clustering methodololghy and we present the fast spectral clustering based on RP-Tree, used in this case as a local data reduction step.
Clustering is a fondamental problem in data mining, statistical machine learning and scientific discovery. In particular in spatial data mining, a wide variety of methods have been devoloped to solve spatial clustering problem (Han et al., 2001).
Some clustering methods are strongly lied to Euclidean geometry, making explicit or implicit assumptions that clusters form convex regions in Euclidean space, spectral methods are more flexible and capturing a wider range of geometry.
Given a set of n data points $\mathrm{x}_1, \ldots \mathrm{x}_\mathrm{n}$, with each $x_i \in R^d$, we define an *affinity graph* $\mathcal{G} = (V,\ E)$ as an undirected graph in which the i^{th} vertex corresponds to the data point x_i. For each edge $(i, j) \in E$, it is associated a weight a_{ij} that encodes the affinity (or similarity) of the data points x_i and x_j. The matrix $A = (a_{ij})_{i,j=1}^{n}$ is the *affinity matrix*.
The goal of the spectral clustering is to partition the data into m disjont classes such that each x_i belongs to one and only one class. Different spectral clustering formalize this partitioining problem in different way. We report the *normalized cuts* (N_{cut}) formulation (Donghui Yan, Ling Huang, Michael I. Jordan, 2009).
Define $W = (V_1,\ V_2) = \sum_{i \in V_1,\ j\ \in V_2}\ a_{ij}$ for two (possibly overlapping) subsets V_1 and V_2 of V.
Let $V = (V_1,\ ..,\ V_m)$ denote a partition of V, and consider the following optimization criterion:
$N_{cut} = \sum_{j=1}^{m} \frac{W\ (V_j,\ V) - W(V_j,\ \ V_j\ )}{W\ (V_j,\ V)}$, where the numerator in the j^{th} term is equal to the sum of the affinities on edges leaving the subset V_j and the denominator is equal to the total

degree of the subset V_j . Minimizing the sum of such terms thus aims at finding a partition in which edges with large affinities tend to stay within the individual subset V_j and in which the size of the V_j are balanced.
The equation above is intractable and it can be rewriting as normalized quadratic form involving indicator vectors that then replaced with real-valued vector, resulting in a generalized eigenvector problem. The problem is redefined in terms of the (normalized) *graph Laplacian* L of A as follows:

$$L = D^{-1/2}\ (D - A)\, D^{-1/2} = I - D^{-1/2} A\ D^{-1/2} = I - L^0$$

where $D = diag\ (d_1,\ \ldots d_n)$ with $d_i = \sum_{j=1}^{n} a_{ij}\ ,\ i = 1, \ldots,\ n$ and where the final equality defines L^0. N_{cut} is based on the eigenvectors of this normalized graph Laplacian.
A specific example of a spectral cluster algorithm based on Gaussian Kernel as the pairwise affinities is defined by:
Algorithm 1 Spectral Clustering (Donghui Yan, Ling Huang, Michael I. Jordan, 2009)

1. Compute the affinity matrix A with elements : $a_{ij} = \exp\left(-\frac{\left\|\mathrm{x_i}-\mathrm{x_j}\right\|^2}{2\sigma^2}\right)\ \ i, j = 1, \ldots,\ n$
2. Compute the diagonal degree matrix D with elements: $d_i = \sum_{j=1}^{n} a_{ij}$
3. Compute the normalized Laplacian matrix: $L = D^{-1/2}\ (D - A)\, D^{-1/2}$
4. Find the second eigenvector v_2 of L
5. Obtain the two partitions using v_2: $S = \{i\ : (\ \mathrm{v}_2)_\mathrm{i} > 0\}$, $\overline{\mathrm{S}} = \{i\ : (\ \mathrm{v}_2)_\mathrm{i} \leq 0\}$

We observe that we have:

1. **Input:** n data points $\{x_i\ \}_{i,j=1}^{n}\ \ ,\ x_i \in R^d$
2. **Ouput**: Bipartition S and $\overline{\mathrm{S}}$ of the input data

Vector quantitization is the problem of choosing a set of representative points that best represent a data set in sense of minimizing a distortion measure.
When we use the RP-tree as a local distortion-minimizing transformation, the algorithm is called "**RP-tree-based approximate spectral clustering**" (**RASP**) and it is obtained by:
Algorithm 2 RASP (Donghui Yan, Ling Huang, Michael I. Jordan, 2009)

1. Build an h-level random projection tree on $\mathrm{x}_1,\ \ldots, \mathrm{x}_\mathrm{n}$; compute the centers of mass $\mathrm{y}_1,\ \ldots \mathrm{y}_\mathrm{k}$ of the data points in the leaf cells as the k representative points.
2. Run a spectral clustering algorithm on $\mathrm{y}_1,\ \ldots \mathrm{y}_\mathrm{k}$ to obtain an m-way cluster membership for each of y_i.

3. Recover the cluster membership for each x_i by looking up the cluster membership of the corresponding centroid y_j.

In the algorithm above we observe that we have:

1. **Input:** n data points $\{x_i\}_{i,j=1}^{n}$, number of representative points k
2. **Ouput**: m-way partition of input data

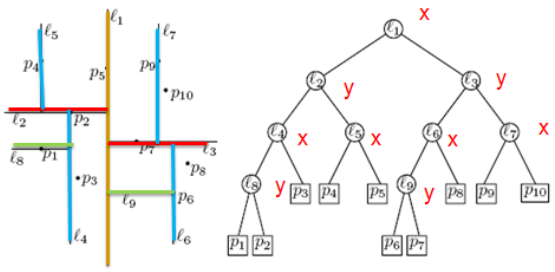
The total computational cost of this method is $O(k^3) + O(hn)$, where the $O(hn)$ term arises from the cost of the bulding the h-level random projection tree.
The vector quantization error is calculated by the average *squared* Euclidean distance between a vector in the set and the representative vector to which it is mapped. This error is closely related (in fact, proportional) to the *average diameter* of cells, that is, the average squared distance between pairs of points in a cell. We remind that in RP-Tree the diameter of the cells for a vector quantization construction method depends on the intrinsic dimension, rather than the extrinsic dimension of the data.
Finally, the vector quantization error of RP tree behaves as $e^{-O\ (\frac{h}{d})}$ with h the depth of the tree and d the intrinsic dimension of the data. Thus the quantization error can be made small as the tree depth as grows.

2.3.4 Spatial data structures: Spatial Partitioning Tree

In this subsection, we briefly review the particular spatial data structure to support SDM.
A *spatial partitioning tree* recursively divides space into increasingly fine partitions.
There are many types of spatial partitioning trees and these spatial structures depend on different split coordinate is choosen at each stage (Verma et al., 2009)

1. **Dyadic tree**: pick a coordinate direction and splits the data at the midpoint along that direction.
2. **k-d tree (k-dimensional tree)**: pick a coordinate direction and splits the data at the median along that direction.
3. **Random Projection (RP-tree)**: split the data at the median along a random direction chosen from the surface of the unit sphere.
4. **Principal Direction (PD or PCA) tree**: split at the median along the principal eigenvector of the covariance matrix.

The splitting rules differ only in the nature of the split; it corresponds to the subroutine called **ChooseRule**.
The core tree-building algorithm is called **MakeTree**.
Input: data set $S \subset R^d$



procedure MakeTree (S)
if |S|< *MinSize*[1] return (*Leaf*)
LeftTree← $MakeTree(\{x \in S : Rule\ (x) = true\})$
RightTree← $MakeTree(\{x \in S : Rule\ (x) = false\})$
return ([*Rule, LeftTree, RightTree]*)

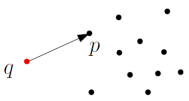
procedureChooseRule (S)
comment: **k-d tree version**
choose a coordinate direction i

$$Rule\,(x) := \ x_i \ \leq median\ (\{z_i : z \in \ S\})$$

return*(Rule)*
Split by x-coordinate (y-coordinate): split by a vertical (horizontal) line that has half the points left or on, and half right.

Uses of K-d trees: classification, regression, vector quantization, nearest neighbor, etc...
In summary, in the Approximation Nearest Neighbor Algorithm:

- **Problem Definition**:
  - Given: a set of n points P in R^2 (e.g. firms in the space)
  - Goal: given a query point q, finds the nearest neighbor p of q in P



- **Motivation**: nearest neighbor search arises in many applications: GIS, statistical classification, clustering, graphics and geometry processing.....

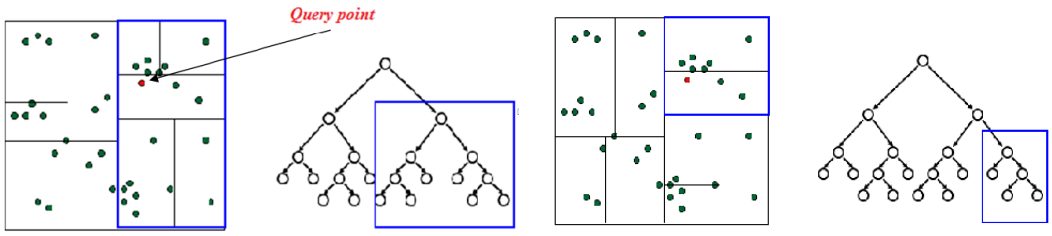
[1] $MinSize$, for the minimum node size,

- **High dimensionality**: the problem is redefined in terms of approximate searching: **(1+ϵ)- approximate nearest neighbor**.
  Given a set of points P in a d-dimensional space R^d, a query point q $\in$ P and $\epsilon > 0$, we say that p $\in$ P is (1+ϵ) nearest neighbor of q if dist(p,q) $\leq$ (1+ϵ)dist(p^*, q) where p^* is the true nearest neighbor to q.

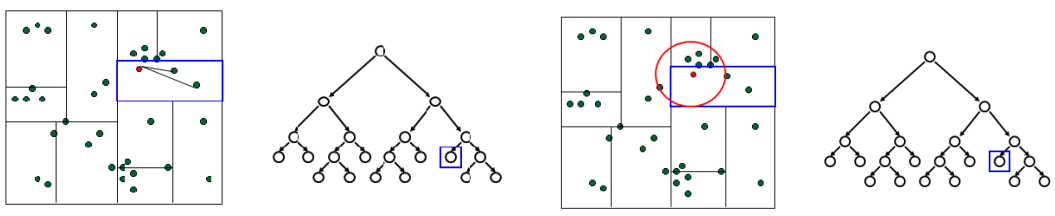
  Low dimensions: kd-tree.
  High dimensions: the performance of kd-tree rapidly degrades and the RP-tree is implied as preprocessing step: project the data point to a subspace of lower dimension and then do apply hybrid sp-tree search (for more details see *Liu T. et al., 2005*).

In particual the *Nearest Neighbor with Kd- Tree* is described as follows:

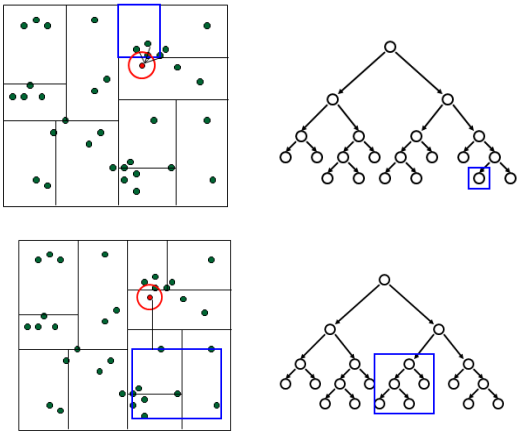
1. Explore the branch of the tree that is closest to the query point first.



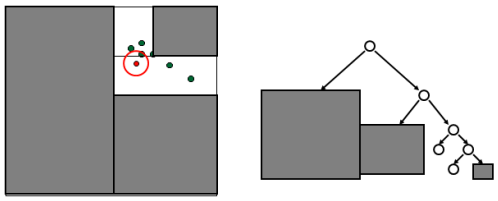
2. When we reach a leaf node, compute the distance to each point in the node



3. Then, backtrack and try the other branch at each node visited and each time a new closest node is found, we can update the distance bounds.



Using the distance bounds and the bounds of the data below each node, we can prune parts of the tree that could not include the nearest neighbor.



Chapter 3

Likelihood estimation of large spatial autoregressive models: a survey of the literature and new evidences

3.1 Introduction

In this section we discuss on computational complexity of exact and approximation solutions to estimate the parameters of spatial autoregressive (SAR) model for large spatial datasets. The ML estimation is the usual approach to estimate the commons spatial econometric models, as suggested by the original solution of Ord (1975). This approach presents the well-known problem of the computation of the logarithm of the determinant of the Jacobian term $|I - \rho\mathbf{W}|$ (or log-Jacobian $\ln|I - \rho\mathbf{W}|$), especially in very large data sets.
So, the ML estimation can be computationally infleasible for large datasets, requiring $O(n^3)$ for a data set of size n and huge amounts of memory making these methods computationally intractable for large n.
In the literature, to reduce the numerical difficulties, recents works has suggested various decomposition and approximation techiniques such as Cholesky factorizations or LU decomposition (Pace and Barry, 1997, Pace and Barry, 1997a,b), Characteristic polynomial method (Smirnov and Anselin, 2001), Trace-based (Smirnov and Anselin 2009), Monte Carlo (Barry and Pace,1999) and Chebyshev approximations (Pace and LeSage, 2004).

An alternative way to overcome this computational problem is suggested by Kelejian and Prucha (1999), the Generalized Method of Moments (GMM) approach.

The main **contributions** of this work are listed as follows:

- we review various approaches proposed in the recent literature to account the numerical difficulties when using Maximum Likelihood estimation for spatial models in very large data sets;
- we provide an overview of well known problem, the computation of the logarithm of the determinant of the Jacobian term;
- to minimize the computational burden, we evaluate, by a Monte Carlo study, the performance of different decomposition and approximation techniques in terms of accuracy and computational complexity, when using various regular grids and large simulated data set, in some ranges of values of spatial coefficient;
- we analyze the double effect on computational complexity: the influence of *"size effect"* and the influence of *"sparsity effect"*.

3.2 A survey of literature

In the section, we discuss the estimation of spatial autoregressive process in dependent variable. Formally, spatial lag model is presented as

$$y = \rho \mathbf{W} y + X\beta + \epsilon \tag{3.1}$$

where $\mathbf{W}$ is a spatial weights matrix, $\epsilon \sim N(0, \sigma^2 I)$ is a vector of random error terms.
The log-likelihood function for the model (3.1) is (Anselin, 1988a):

$$L(\rho, \beta, \sigma^2) = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}(Ay - \mathbf{X}\beta)'(Ay - \mathbf{X}\beta)' + \ln|A| \tag{3.2}$$

where $\mathbf{A} = \mathbf{I} - \rho\mathbf{W}$, $|\cdot|$ denotes the determinant.
From the first-order conditions to this model yields b as estimator of β (Anselin,1988a):

$$\begin{aligned} & b = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\mathbf{A}\mathbf{y} \\ & or \\ & b = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} - \rho(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y} = b_0 - \rho b_L \end{aligned} \tag{3.3}$$

The OLS estimators b_0 and b_L are obtained from a regression of the $\mathbf{X}$ on $\mathbf{y}$ and on $\mathbf{W}\mathbf{y}$ respectively. The ML estimates for β is a function of these auxiliary regression coefficients as well as of ρ. The estimation of ρ cannot be expressed analitically, so the estimate for β can be found conditional upon ρ.
Furthermore, we have:

$e_0 = y - \mathbf{X}b_0$
$e_L = Wy - \mathbf{X}b_L$
The estimate for the error variance σ^2, conditional upon ρ can be expressed as:

$$\sigma_\rho^2 = \frac{1}{N}(e_0 - \rho e_L)'(e_0 - \rho e_L) \tag{3.4}$$

Substitution of the estimates for β and σ^2 into the likelihood results, it yields the concentrated likelihood function in the following form:

$$L_c = C - \frac{N}{2}\ln\left[\left(\frac{1}{N}\right)(e_0 - \rho e_L)'(e_0 - \rho e_L)\right] + \ln|I - \rho\mathbf{W}| \tag{3.5}$$

where C is the usual constant. This expression is a **non-linear function** in the parameter ρ so, the maximation of it needs to apply an appropriate nonlinear optimization routine.
The information matrix for this model can be expressed as (Smirnov, 2005):

$$I(\rho, \beta, \sigma^2) = \begin{bmatrix} tr(AA) + tr(A'A) + \frac{\beta' X' A' A X \beta}{\sigma^2} & \frac{\beta' X' A' A X}{\sigma^2} & \frac{tr(A)}{\sigma^2} \\ \frac{X' A X \beta}{\sigma^2} & \frac{X'X}{\sigma^2} & 0 \\ \frac{tr(A)}{\sigma^2} & 0 & \frac{N}{2\sigma^2} \end{bmatrix} \tag{3.6}$$

where A =W(I-ρW)$^{-1}$.
A distinguishing feature of the likelihood for spatial lag model is the presence of a *Jacobian term* of the form $\ln|\mathbf{I} - \rho\mathbf{W}|$. In computational terms, the maximization of the log-likelihood involves a nonlinear optimization that requires the evalutation of the Jacobian for values of the parameter ρ (Smirnov and Anselin, 2001). In addition, according to Bivand et al. (2013), with large samples, the computation of the information matrix involves computation of a large sparse matrix. It may be approximated by a numerical Hessian, the computation of which also involves the values of the Jacobian.
Several solutions to overcome these issues have been offered in the literature and we will treat here to give a brief overview.
The original solution as proposed by Ord (1975) consists of exploiting the decomposition of the jacobian in terms of the eigenvalues ω_i (i=1, ...,n) of the spatial weighting matrix $\mathbf{W}$:

$$\ln|\mathbf{I} - \rho\mathbf{W}| = \sum_{i=1}^{n} \ln(1 - \rho\omega_i) \tag{3.7}$$

This approach involves two steps: in the first, the eigenvalues of spatial weights matrix are computed, and in the second the log-likelihood or their derivatives are evaluated at each iteration, substituting a value for ρ in (3.7). The computation complexity of routines on typically dense matrices is $O(n^3)$ operations and $O(n^2)$ memory (Saad, 1992).
An alternative methods for computing the Jacobian proposed by Pace and Barry (1997a,b) regards sparse matrix techniques that provide powerful methods to quickly evaluate the Jacobian in spatial autoregressive models. We have the Cholesky factorization of a sparse, symmetric, positive-definite matrix and the LU factorization if simmetry requirements on

the matrix need to be relaxed. As observed by Bivand et al. (2013) for the same symmetric, positive-definite matrix, the computation of Jacobian by Cholesky or LU factorizations is identical within machine precision (Higham, 2002).
The Cholesky factorization consists of solving $(\mathbf{I}-\rho\mathbf{W}) = \mathbf{LL'}$, where $\mathbf{L}$ is a lower triangular matrix. The determinant of the Jacobian is thus $|\mathbf{I}-\rho\mathbf{W}| = |\mathbf{L}||\mathbf{L'}| = |L^2|$. The log-Jacobian can be expressed as

$$\ln|\mathbf{I}-\rho\mathbf{W}| = 2\sum_{i=1}^{n}\ln(l_{ii}) \tag{3.8}$$

where l_{ii} (i=1,...,n) as the diagonal elements of $\mathbf{L}$.
In the LU case for a nonsingular matrix, the log-determinant is

$$\ln|\mathbf{I}-\rho\mathbf{W}| = \sum_{i=1}^{n}\ln|u_{ii}| \tag{3.9}$$

where u_{ii} (i=1,...,n) as the diagonal elements of $\mathbf{U}$, from $(\mathbf{I}-\rho\mathbf{W}) = \mathbf{LU}$ with $\mathbf{U}$ is an upper triangular matrix.
As argued in Smirnov and Anselin (2001) a potential problem in the Cholesky decomposition is that the parameter values for ρ is unknown. The valid interval for parameter values is $(1/\omega_{min}, 1/\omega_{max})$, where ω_{min} and ω_{max} are respectively, the smallest and largest eigenvalues of $\mathbf{W}$ (Anselin,1988a). For row-standardized spatial weights, the largest eigenvalue is always $+1$ and the lower bound is typically less than -1. In the literature the interval suggested is $(-1,+1)$. The complexity of the factorization for a typical spatial weigths matrix is reduced from $O(n^3)$ to $O(n^2)$ and this approch also requires less memory to compute the Jacobian term than the eingenvale approch (Pace and Barry, 1997a,b). If we replace in the equation 3.8 the **LL'** factorization with **LDL'**, where $\mathbf{D}$ is a diagonal matrix, and re-express the log-determinat, changing the sign of ρ we have:

$$\ln|\mathbf{I}+\rho\mathbf{W}| = n\ln(\rho) + 2\ln\left(\left|\mathbf{W}+\frac{1}{\rho}\mathbf{I}\right|\right) \tag{3.10}$$

This updating procedure uses the Cholesky factorizations of $\mathbf{W}$ and **-W**.
A number of alternative approaches have been suggested in the literature by Griffith (1992) and Griffith and Sone (1995) that describing analytical ways of calculating the eigenvalues for a regual square surface. In particular, in our analysis to compute *analytical eigenvalues* we follow the approach in Griffith (2000 pag.99, corollary 2.2):

$$\lambda_{kl} = \left[\cos\left(\frac{k2\pi}{P}\right) + \cos\left(\frac{l2\pi}{Q}\right)\right]/2, k = 1,2...P, l = 1,2,...Q \tag{3.11}$$

Walde et al. (2008) examine some approximations of the Jacobian such as Monte Carlo approach or Chebyshev approximation. Chebyshev approximation is proposed by Pace and LeSage (2004) in the following form:

$$\ln|\mathbf{I}-\rho\mathbf{W}| \simeq \sum_{j=1}^{q+1} c_j(\rho) tr(\mathbf{T}_{j-1}(\mathbf{W})) - \frac{n}{2}c_1(\rho) \tag{3.12}$$

where:

- tr denotes the matrix trace operator;
- $\mathbf{T}_0(\mathbf{W}) = \mathbf{I}$;
- $\mathbf{T}_1(\mathbf{W}) = \mathbf{W}$;
- $\mathbf{T}_2(\mathbf{W}) = 2\mathbf{W}^2 - \mathbf{I}$, $\mathbf{T}_{k+1}(\mathbf{W}) = 2\mathbf{W}\mathbf{T}_k(\mathbf{W}) - \mathbf{T}_{k-1}(\mathbf{W})$;
- q represents the highest power of the approximating polynomial which thus has $q+1$ coefficients $c_j(\rho)$. The $c_j(\rho)$ are given by

$$c_j(\rho) = \left(\tfrac{2}{q+1}\right) \sum_{k=1}^{q+1} \ln\left[1-\rho\cos\left(\tfrac{\pi(k-0.5)}{q+1}\right]\right) \cos\left(\tfrac{\pi(j-1)(k-0.5)}{q+1}\right)$$

  where π is the constant pi-value. As shown by Pace and LeSage (2004), the quadratic approximation leads to a sufficient accuracy of the log-Jacobian.

The feature of the *Monte Carlo* approach (Barry and Pace, 1999) is to estimate the log-determinat by p independent random variables V_i in the following form:

$$V_i = -n \sum_{k=1}^{m} \frac{\mathbf{r}_i'\mathbf{W}\mathbf{r}_i}{\mathbf{r}_i'\mathbf{r}_i}\frac{\rho^k}{k} \tag{3.13}$$

where $i = 1, .., p$ and $\mathbf{r}_i \sim N_n(0, I)$, $\mathbf{r}_i$ independent of $\mathbf{r}_j$, if $i \neq j$.
They proved the bound for the term $|E[\overline{V}] - \ln|\mathbf{I}-\rho\mathbf{W}|| \leq \frac{n\rho^{m+1}}{(m+1)(1-\rho)}$.
For a given value of ρ, the mean of the generated V_i is used as an estimate for the log Jacobian. The precision of this estimate can be manipulated by means of the tuning parameters p (the number of random variables generated) and m (the number of elements in the sum of ratios of quadratic forms), in Barry and Pace (1999) they suggested $p = 16$ and $m = 30$.
Barry and Pace (1999) shown that the order of complexity is $O(n \ln n)$ and it allows to estimate models with more than 1 million observations, but the parameter space employed in setting the range of values for ρ may be inappropriate (Barry and Pace, 1999 only use positive values for spatial parameter).
Finally, another method based on the characteristic polynomial of the spatial weights matrix suggested by Smirnov and Anselin (2001). In practice, the determinant of $(\mathbf{W}-\lambda\mathbf{I})$ is expressed by a polynomial in the coefficient λ, the roots of which correspond to the eigenvalues of $\mathbf{W}$:

$$|\mathbf{W}-\lambda\mathbf{I}| = (-1)^n(q_0\lambda^n + q_1\lambda^{n-1} + ... + q_{n-1}\lambda + q_n) \tag{3.14}$$

where $q_i(i = 0, 1, ..., n)$ are the coefficients of the characteristic polynomial. Setting $\lambda = 1/\rho$ and multiplying both sides by $(-\rho)^n$ yields the Jacobian determinant as a function of the coefficient of the characteristic polynomial of $\mathbf{W}$:

$$|\mathbf{I} - \rho\mathbf{W}| = q_0 + q_1\rho + ... + q_{n-1}\rho^{n-1} + q_n\rho^n \tag{3.15}$$

The computational complexity associated with (3.14) and (3.15) is $O(n)$ and the characteristic coefficients only need to be computed once, similar to the eigenvalue approach.
The coefficients of the characteristic polynomial are computed by means of a divide-and-conquer algorithm, the original problem is recursively split into smaller problems until the solution for the smaller problem becomes computationally feasible. Griffith (2004) provides approximation to terms required in the characteristic polynomial approach of Smirnov and Anselin (2001). In particular for a *row-standardized rook contiguity spatial weights* for a P-by-Q regular grid, we have:

$$\hat{q}_2 = 0.11735 + 0.10091\left(\frac{1}{P^{5/4}} + \frac{1}{Q^{5/4}}\right) + \frac{0.42844}{PQ} \tag{3.16}$$

$$\hat{q}_4 = 0.07421 + 0.05730\left(\frac{1}{P^{2/3}} + \frac{1}{Q^{2/3}}\right) + \frac{0.66001}{PQ} \tag{3.17}$$

$$\hat{q}_{20} = 0.05521 + 0.52467\left(\frac{1}{P^7/4} + \frac{1}{Q^7/4}\right) + \frac{2.48015}{PQ} \tag{3.18}$$

$$\ln(|\mathbf{I} - \rho\mathbf{W}|) = -\ln(1 + \hat{q}_2(\rho^2) + \hat{q}_4(\rho^4) + \hat{q}_{20}(\rho^{20}))PQ \tag{3.19}$$

All these techniques are based on two step: 1) determining the logarithm of the Jacobian, conditional upon ρ; 2) evalutation of log-likelihood function at each iteration, substituting a value for ρ. In addition, a computation method for calculating lower moments of the actual distribution of eigenvalues of the spatial weights and applying those for the efficient calculation of the log-determinant has been proposed by Smirnov and Anselin (2009). Consider the j-th non-central moment of the set of eigenvalues of the matrix$\mathbf{W}$:

$$\Omega^j = \sum_{i=1}^{n} \omega^j \tag{3.20}$$

where ω^j is the $i - th$ eigenvalue of $\mathbf{W}$ The log-determinant of a positive definite matrix $\mathbf{I} - \rho\mathbf{W}$ is given by

$$\ln(|\mathbf{I} - \rho\mathbf{W}|) = -\lim_{n\to\infty} \sum_{j=1}^{n} \frac{1}{j}\rho^j\Omega^j \tag{3.21}$$

It derives expanding the logarithms around zero with respect to $\rho\omega_i$ using the Taylor series

$$\ln(|\mathbf{I} - \rho\mathbf{W}|) = -\sum_{j=1}^{\infty} \tfrac{1}{j}\rho^j tr(\mathbf{W}^j) = -\sum_{j=1}^{\infty} \tfrac{1}{j}\rho^j\Omega^j$$

The operational use of the equation (3.21) involves the computation of the limit using the asymptotic properties of the power series

$$\ln(|\mathbf{I} - \rho\mathbf{W}|) = -\sum_{j=1}^{m} \frac{1}{j}\rho^j\Omega^j - R_m(\rho) \tag{3.22}$$

where $R_m(\rho) = \lim_{n\to\infty} \sum_{j=m+1}^{n} \frac{1}{j}\rho^j\Omega^j$ is a correction term.
The computational solution involves three components: the computation of the exact lower moments of eigenvalues of matrix $\mathbf{W}$, and their use in the formula (3.21) for any ρ; approximating R_m (ρ) using asymptotic properties of the converging power series of the parameter ρ and combining finite and asymptotic properties of the eigenvalue moments to calculate the log-determinant.
The first component is based on

$$\Omega^j = tr\mathbf{W}^j = \sum_{i=1}^{n} \eta_i'\mathbf{W}^j\eta_i \tag{3.23}$$

where η_i is a vector of canonical base, of length n of zeros, except for the $i-th$ element equal to one.
The second component regards the asymptotic properties of the moments of the eigenvalues: the moments with even indices converge to $q_1 + q_2$ as indicate by

$$\lim_{n\to\infty} \Omega^{2j} = q_1 + q_2$$

and the moments with odd indices converge to $q_1 - q_2$

$$\lim_{n\to\infty} \Omega^{2j+1} = q_1 - q_2$$

Combining finite and asymptotic properties of moments of eigenvalues, Smirnov and Anselin shown the interpolating scheme to provide an approximation of $R_m(\rho)$, by four higest exact moments $m-3, ..., m$. We have:

$$\Omega^{m+2j} = \Omega^m \times \left(\frac{\Omega^m}{\Omega^{m-2}}\right)^j \tag{3.24}$$

and

$$\Omega^{m+2j-1} = \Omega^{m-1} \times \left(\frac{\Omega^{m-1}}{\Omega^{m-3}}\right)^j \tag{3.25}$$

where $m$as the number of exactly computed moments Ω^j.
As argued by Bivand et al. (2013) the first and third component is two interesting innovations.

Consider the following cross sectional spatial model

$$y_N = \alpha e_N + X_N\beta + \lambda W_N y_N + \epsilon_N = Z_N\gamma + \epsilon_N \tag{3.26}$$

where $Z_N = (e_N, X_N, W_N y_N)$ and $\gamma = (\alpha, \beta', \lambda)'$

Kelejan and Prucha (1998) suggests the **2SLS estimation** for γ.
Let $H_N = (e_N, X_N, G_N)$ be the matrix of instruments, where G_N is a $N \times r$ matrix of nonstochastic variables, $r \geq 1$ and it could be taken to be the linearly independent columns of $(\mathbf{W}_N\ \mathbf{X}_N, \mathbf{W}_N^2\ \mathbf{X}_N, ..., \mathbf{W}_N^q\ \mathbf{X}_N)$.
Let $P_{H_N} = H_N(H_N'H_N)^{-1}H_N$ and note that since H_N contains e_N and X_N, $P_{H_N e_N} = e_N$ and $P_{H_N X_N}$=X_N.
Let $\hat{Z}_N$=$P_{H_N} Z_N = (e_N, X_N, W_N y_N)$.

$$\hat{\gamma}_{2SLS,N} = (\hat{Z_N}'\hat{Z_N})^{-1}\hat{Z_N}'y_N \tag{3.27}$$

Finally, the algorithm proposed by Kazar et al. (2004) provides better approximation when the data is strongly correlated (i.e., spatial dependency is high) and problem size gets high. The key idea of the proposed algorithm is to find only some of the eigenvalues of a large matrix, instead of finding all the eigenvalues, by reducing the size of large matrix dramatically using Gauss-Lanczos algorithm. In their paper, by experimental results show that the proposed algorithm saves computation time for the large problem sizes with respect to ML-based approximate SAR model solutions, namely Taylor's series, and Chebyshev polynomials.

3.2.1 Related works and our contribution

Walde et al. (2008) in their work, gives the results for the only accuracy spatial autoregressive disturbance (SEM model) for Ord method (1975), various decomposition techniques (Cholesky, LU), Chebyshev and MC approximation, Characteristic polynomial approach and GMM.
With respect to this work:

- we consider the performance in terms of **timing** and **accuracy** of approximation and decomposition techinques for **different spatial (spatial lag) econometric model** and for **different regular grids** ;
- the Characteristic polynomial (Smirnov and Anselin, 2001) approach is substituted by the **approximation of Griffith (2004)** and **Spatial 2SLS approach** is considered instead of GMM;
- we consider different size of regular grids and we analyze the influence of **"size effect"** and **" sparsity effect"** in terms of computational complexity.

Bivand et al. (2013) in their paper focuses only on accuracy of sparse matrix and approximate approaches to computing the Jacobian log-determinant term.
In our work we consider:

- the **timing** and **accuracy** of spatial lag and the influence of **"size effect"** and **"sparsity effect"** for various regular grids, in terms of computational complexity;

- analytical eigenvalues approach for row-standardized spatial weights matrix *(Griffith 2000, Corollary 2.2 page 99)* instead of analytical eigenvalues approach for binary spatial weights matrix *(Griffith and Sone, 1995)*.

3.3 Experimental design

We conduct Monte Carlo experiments, by considering *timing* and *accuracy* of the various techniques to solve the problem of computing the Jacobian in spatial models, for various regular lattice structures. The regular lattices range from 15×15 ($n = 225$) to 1024×1024 ($n = 1048576$) to investigate the double effect on computational complexity for spatial autoregressive model:

1. "**sample size effect**": the influence of different sample size, by fixing a specific degree of sparsity of spatial weights matrix;

2. "**sparsity effect**": the influence of different degrees of sparsity[1] of spatial weights matrix for each n.

We proceed as described in following step.
In first step we generate the explanatory variable x_1 from the uniform distribution U (0,1). The vector β consists of two coefficients: intercept $\beta_1 = 1$ and $\beta_2 = 1$. Further, we generate the error terms ϵ which are assumed to be normally i.i.d. distributed, with mean 0 and variance 1.
In second step, we compute the dependent variable $\mathbf{y}$ for the spatial lag model
$\mathbf{y} = \rho\mathbf{W} + x_1\beta + \epsilon$, using $\mathbf{W}$, x_1 and the errors obtained in the previous step. In particular we consider different values of spatial parameter $\rho = \{-0.7, -0.5, 0.3, 0.5, 0.7\}$.
For each Monte Carlo trial (number of iterations is 100), we compare the accuracy with respect to the true parameter values and timing for decomposition and approximation techniques.
All the experiments were conducted on a CNR-IASI Server with Intel(R) Core(TM) i7-3770 CPU with 3.40GHz (4 core, 8 thread) and 32.00 GB memory. The development tool was R 3.0.1. [2]
In the Table 3.1 we start to compare the timing for only computing Jacobian values for ρ in the range $[-0.9, 0.99]$, in steps of 0.01, totalling to 190 values for row-standardized spatial weights matrices, by using different approaches.
As can we note, the original Ord solution and matrix decompositions present a high degree of computational complexity when N increases from 225 to 1048576. The approximation

[1] The sparsity of $\mathbf{W}$ measured by the percentage of zero elements in the off-diagonal elements.
[2] The associated source code is available from the author upon request.

Table 3.1: Timings for computing 190 Jacobian values for $\rho[-0.9, 0.99]$ for regular grids, rook contiguity, using different sparse matrix decomposition and approximation

Regular grid	15X15	31X31	63X63	127X127	255X255	320X320	400X500	500X1000	800X1000	1024X1024
Eigen setup	0.035	1.505	61.858	3401.181						
Eigen	0.002	0.006	0.023	0.088						
Cholesky decomposition	0.189	2.162	15.205	127.766	1190.130	3988.987	12888.730	50932.271	130815.065	304685.741
LU decomposition	0.641	1.458	12.116	115.495	1143.210	3910.670	12733.132	50594.358	130573.415	305281.423
MC setup	0.013	0.020	0.057	0.196	0.850	1.328	6.646	16.420	27.333	39.986
MC	0.025	0.025	0.026	0.026	0.026	0.026	0.027	0.027	0.026	0.026
Chebyshev q=2 setup	0.007	0.007	0.010	0.018	0.057	0.086	0.164	1.377	2.691	5.045
Chebyshev q=4 setup	0.007	0.010	0.02	0.065	0.831	0.692	2.679	6.163	8.974	14.345
Chebyshev q=2	0.015	0.014	0.015	0.014	0.014	0.014	0.014	0.015	0.015	0.014
Chebyshev q=4	0.032	0.032	0.032	0.031	0.031	0.032	0.033	0.032	0.032	0.031
Analytical eigenvalues	0.003	0.012	0.055	0.22	0.903	1.431	2.724	6.646	10.245	13.196
Approximation polynomial approach	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
N	225	961	3969	16129	65025	1 02400	200 000	500 000	800 000	1 048 576
Sparsity (%)	1.7777	0.4162	0.1007	0.0248	0.0061	0.0039	0.002	0.0008	0.0005	0.0003

Notes: Results are in seconds, to compute the approximation of polynomial approach we follow Griffith (2004). The sparsity indicates the percentage of nonzero weights.

techniques (MC and Chebyshev) show a less degree of computational complexity w.r.t. previous approaches. A significant decrease of computational complexity appears in the case of analytical eigenvalues and in the approximation of polynomial approach. By following the experimental design described above, we proceed examining the order of computational complexity to estimate the parameters of spatial lag model. As suggested in Smirnov and Anselin (2001), we assess the time involved in the computations as a function of sample size, for several methods analyzed in the previous section, by considering first order rook contiguity matrix. In this way we consider the "Guess Ratio" rule defined by the ratio of observed running time (t(n)) and the underlying algortihm's running time function (theoretical order of computational complexity, g(n)). If the ratio grows as the input size increases, then g(n) understimates the running time; if the ratio converges to 0 as the input size increases, then the g(n) is an overestimate. In the case that the ratio converges to some constat greater then 0, then g(n) is a good estimation for the growth rate of t(n).
The correspondent results are shown in the following tables.

Table 3.2: Order of computational complexity (timing) for regular grids, rook **first order** contiguity, to estimate one specific spatial autocorrelation parameter using different sparse matrix decompositions and Ord solution

Grid	Ord			Cholesky			LU		
	T/n	T/n^2	T/n^3	T/n	T/n^2	T/n^3	T/n	T/n^2	T/n^3
15X15	64.71	2.88	1.28	76.13	3.384	1.5038683	83.01	4.880	1.63968
31X31	221.12	2.30	18.66	209.76	2.183	0.2271337	217.28	2.261	0.23527
63X63	3,925.49	9.89	0.25	74.82	0.188	0.0047493	105.72	0.266	0.00671
127X127	145,730.04	90.53	0.56	24.74	0.015	0.0000951	303.12	0.188	0.00117
255X255				23.37	0.004	0.0000055	238.77	0.037	0.00006
320X320				23.92	0.002	0.0000023	568.97	0.056	0.00005
400X500				87.04	0.002	0.0000022	1017.08	0.051	0.00003
500X1000				74.58	0.001	0.0000003	1394.77	0.028	0.00001
800X1000									
1024X1024									

Notes: Results are computed in 0.001 seconds for 100 iterations (elapsed time) and scaled by 10 for Cholesky and 1000 for Ord and LU.

Table 3.3: Order of computational complexity (timing) for regular grids, rook **first order** contiguity, to estimate one specific spatial autocorrelation parameter using Chebyshev and MC Approximations

Grid	Chebyshev		MC	
	T/n	T/nlogn	T/n	T/nlogn
15X15	72.5733	30.8536	70.5422	29.9901
31X31	208.3985	69.5685	214.8814	72.0420
63X63	69.9186	19.4290	71.0617	19.7466
127X127	18.0985	4.3014	18.5584	4.4107
255X255	5.2612	1.0931	5.7181	1.1880
320X320	3.7371	0.7459	4.2510	0.8485
400X500	2.4951	0.4707	1.7971	0.3390
500X1000	1.8695	0.3280	2.4214	0.4249
800X1000				
1024X1024				

Notes: Results are computed in 0.001 seconds for 100 iterations (elapsed time).

Table 3.4: Order of computational complexity (timing) for regular grids, rook **first order** contiguity, to estimate one specific spatial autocorrelation parameter using Analytical Eigenvalues, Griffith approximation and Spatial 2SLS estimation

Grid	Analytical eigen		Griffith		2SLS	
	T/n	T/nlogn	T/n	T/nlogn	T/n	T/nlogn
15X15	23.0133	9.7838	7.6267	3.2424	7.6444	3.2499
31X31	9.6004	3.2187	1.8658	0.6255	2.0926	0.7016
63X63	6.6133	1.8377	0.5709	0.1586	0.9607	0.2670
127X127	6.6135	1.5718	0.3417	0.0812	0.4855	0.1154
255X255	6.3595	1.3213	0.2785	0.0579	0.0004	0.1095
320X320	6.4416	1.2857	0.2592	0.0517	0.4287	0.0856
400X500	6.0711	1.1453	0.1441	0.0272	0.1945	0.0367
500X1000	6.4378	1.1296	0.1649	0.0289	0.2243	0.0394
800X1000	6.3747	1.0799	0.1794	0.0304	0.2432	0.0412
1024X1024	6.3149	1.0489	0.1866	0.0310	0.2630	0.0437

Notes: Results are computed in 0.001 seconds for 100 iterations (elapsed time).

In summary, by evaluating the timing in function of sample size over the range of n, from the comparison of our analytic results, the asymptotic performance is summarized as:

- Ord's solution: $O(n^3)$
- Cholesky and LU decomposition: $O(n^2)$
- Chebychev and MC Approximation: $O(nlogn)$
- Analytical eigenvalues for regular grids (Griffith, 2000): $O(n)$
- Griffith's Approximation Characteristic polynomial: $O(n)$
- Spatial 2SLS estimation: $O(nlogn)$

In order to evaluate the sparsity effect, we consider three different order contiguity matrix: first order (W1), second order (W2) and seven order (W7) and to analyze the size effect, we consider a specific degree of sparsity of spatial matrix (Ws). The main results are reported in following tables.[3]

Table 3.5: Computational complexity (timing) for regular grids, rook **first second, seven, order** contiguity and **fixed sparsity degree**, to estimate one specific spatial parameter using different sparse matrix decomposition, approximation and two stage least squares

Grid	Ord				Cholesky				LU			
	W1	W2	W7	Ws	W1	W2	W7	Ws	W1	W2	W7	Ws
15X15	1.2782	1.2696	21.0031	1.2782	3.3837	3.0035	47.6674	3.3837	4.8800	4.9728	49.9639	4.8800
31X31	18.6553	50.4492	0.4571	17.5942	2.1828	6.3373	4.9515	2.3320	2.2610	6.4941	8.9165	3.0241
63X63	0.2492	0.2158	0.2499	0.4501	0.1885	0.1887	1.3366	9.0021	0.2664	0.3152	10.0189	38.4155
127X127	0.5602	0.4855	0.5625	1.1204	0.0153	0.0198	1.1822	3.6382	0.1879	0.1043	8.3515	26.9562
255X255					0.0036	0.0053	0.2768	0.8665	0.0367	0.0795	6.9617	18.9152
320X320					0.0023	0.0041	0.1801	0.2064	0.0556	0.0310	5.8031	13.2728
400X500					0.0044	0.0032	0.3354	0.0491	0.0509	0.0382	4.8373	9.3136
500X1000					0.0015	0.0034	0.1150	0.0117	0.0279	0.0379	4.0323	6.5353
800X1000												
1024X1024												

Notes: Results are computed in 0.001 seconds, by considering the order of computational complexity, for 100 iterations (elapsed time) and scaled by 10 for Cholesky and 1000 for Ord and LU.

[3]As argued in Smirnov and Anselin (2001), we consider only the relative magnitudes of timing over size sample, because the absolute values of timing are not of interest.

Table 3.6: Computational complexity (timing) for regular grids, rook **first second, seven, order** contiguity and **fixed sparsity degree**, to estimate one specific spatial parameter using different sparse matrix decompositions, approximations and two stage least squares

Grid	Chebyshev				MC			
	W1	W2	W7	Ws	W1	W2	W7	Ws
15X15	30.8536	37.6426	454.4394	30.8536	29.9901	37.1135	448.3382	29.9901
31X31	69.5685	202.3083	160.4575	78.4239	72.0420	203.0246	156.3901	75.0318
63X63	19.4290	19.0748	35.1678	173.1902	19.7466	18.7925	21.5284	28.8433
127X127	4.3014	4.6026	14.9122	414.0678	4.4107	4.3417	4.0949	23.3004
255X255	1.0931	1.4379	13.2348	89.7848	1.1880	1.2340	1.1600	251.7991
320X320	0.7459	1.2022	10.3512	19.4686	0.8485	0.9622	0.9045	460.2351
400X500	0.4707	0.9241	11.3462	14.2215	0.3390	0.5962	0.5605	837.6279
500X1000	0.3280	0.7627	11.1373	10.9154	0.4249	0.4409	0.4144	1524.4828
800X1000								
1024X1024								

Notes: Results are computed in 0.001 seconds, by considering the order of computational complexity, for 100 iterations (elapsed time).

Table 3.7: Computational complexity (timing) for regular grids, rook **first second, seven, order** contiguity and **fixed sparsity degree**, to estimate one specific spatial parameter using different sparse matrix decomposition, approximation and two stage least squares

Grid	Analytical eigen				Griffith				2SLS			
	W1	W2	W7	Ws	W1	W2	W7	Ws	W1	W2	W7	Ws
15X15	23.0133	23.1778	13.5200	23.0133	7.6267	10.1289	10.5867	7.6267	3.2499	3.1101	3.3217	3.2499
31X31	9.6004	11.9407	7.8980	7.2581	1.8658	2.0791	1.7419	1.8169	0.7016	0.6147	0.6677	0.6255
63X63	6.6133	6.6982	6.4843	6.3270	0.5709	0.9997	0.6168	0.8327	0.2670	0.1659	0.2172	0.3037
127X127	6.6135	6.3692	6.2246	7.0219	0.3417	0.5599	0.3482	1.7035	0.1154	0.0015	0.1216	0.6011
255X255	6.3595	6.1390	6.3358	10.9262	0.2785	0.3550	0.2805	5.5973	0.1095	0.0545	0.0892	0.1903
320X320	6.4416	6.0564	6.5260	14.3784	0.2592	0.3486	0.2868	9.7459	0.0856	0.0528	0.0863	3.0642
400X500	6.0711	6.0370	6.2850	28.6739	0.1441	0.3001	0.2540	16.9694	0.0367	0.0485	0.0793	6.3824
500X1000	6.4378	5.9996	6.2471	74.9080	0.1649	0.3070	0.2487	29.5470	0.0394	0.0469	0.0757	12.4457
800X1000	6.3747	6.0679	6.2621	119.4902	0.1794	0.2791	0.2643	51.4468	0.0412	0.0451	0.0726	24.2691
1024X1024	6.3149	6.3149	6.2200	155.9921	0.1866	0.2049	0.3048	89.5786	0.0437	0.0452	0.0786	47.3247

Notes: Results are computed in 0.001 seconds, by considering the order of computational complexity, for 100 iterations (elapsed time).

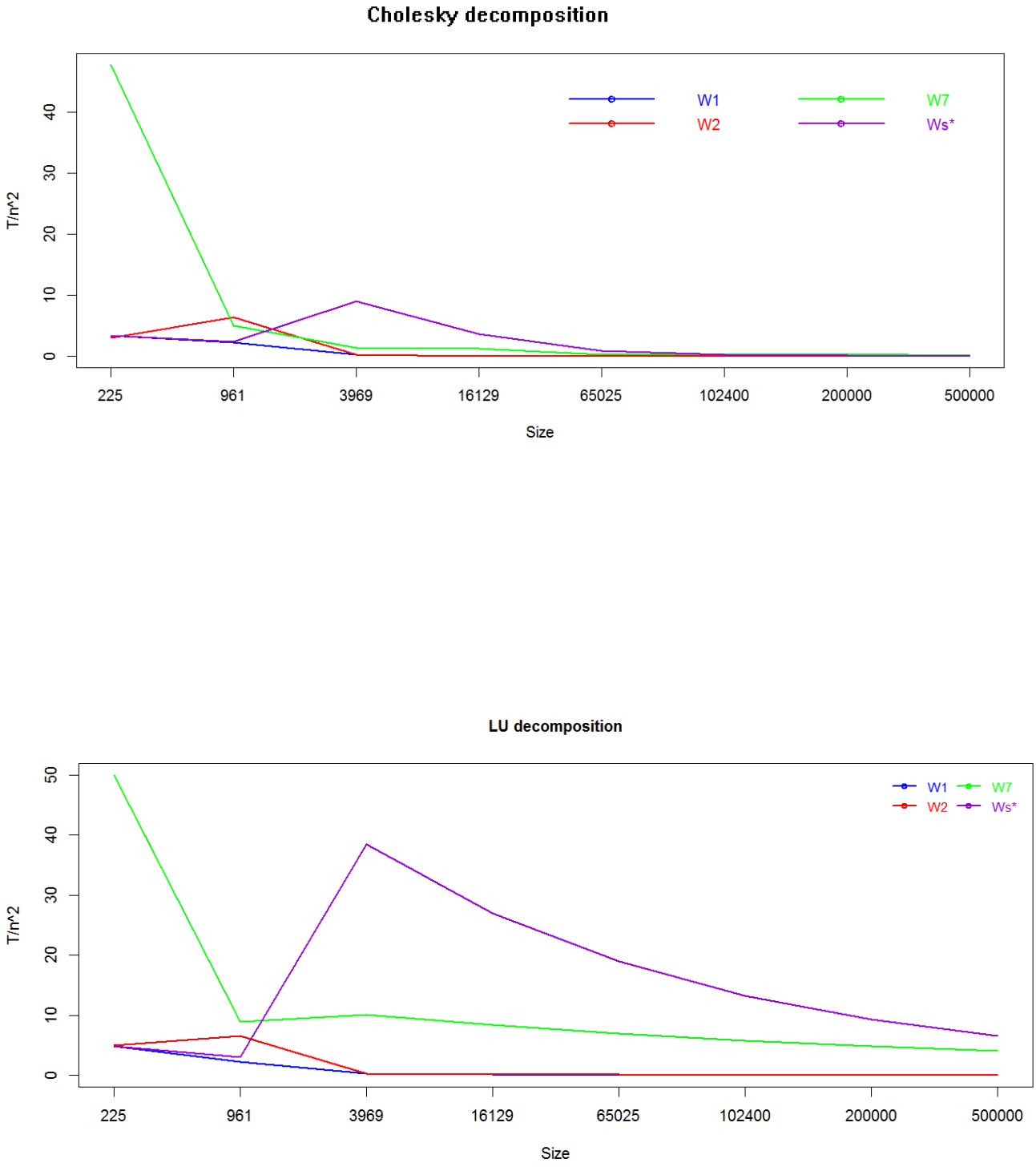


Figure 3.1: Size and sparsity effect on sparse matrix decompositions

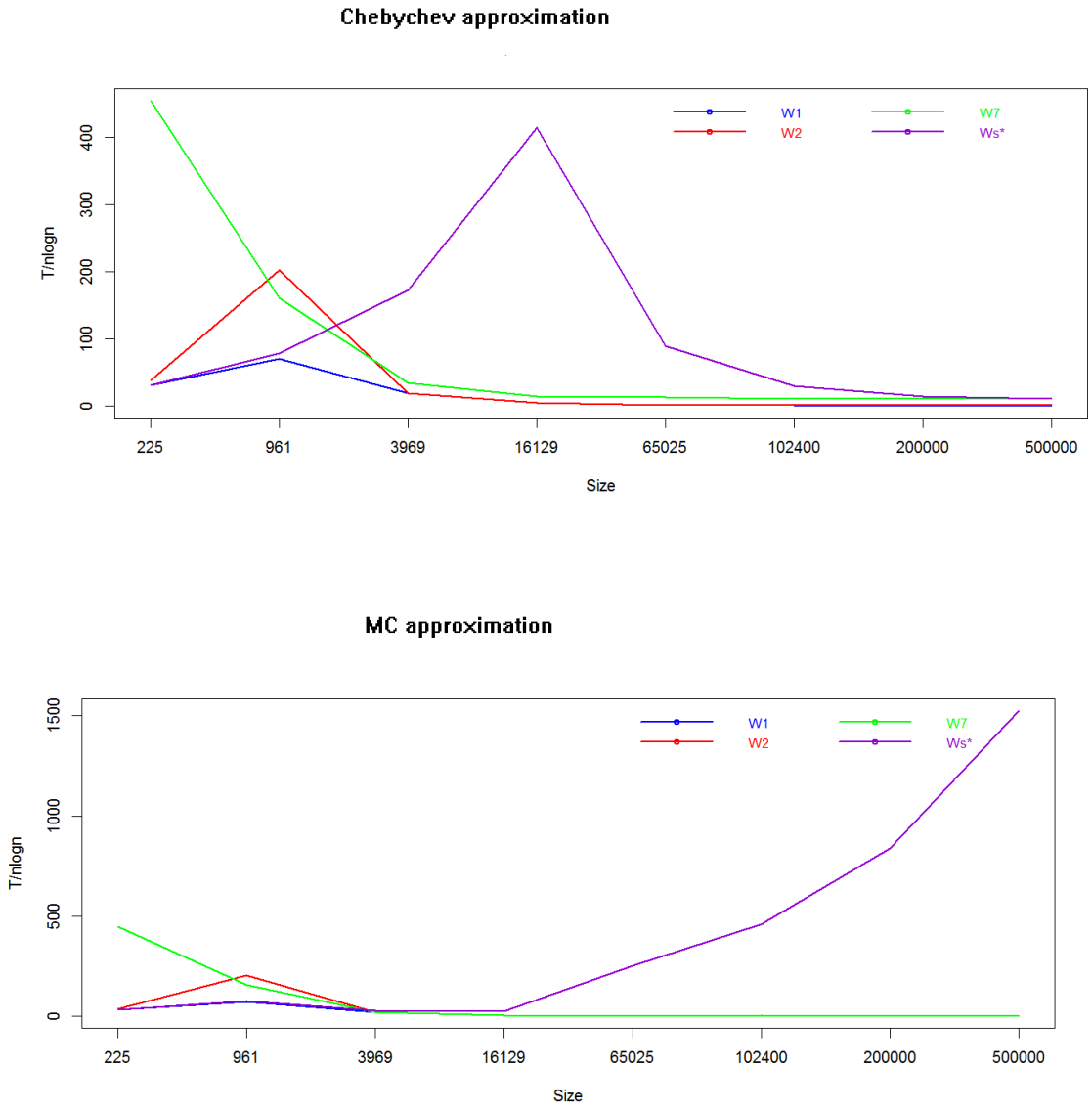


Figure 3.2: Size and sparsity effect on Chebyshev and MC approximations

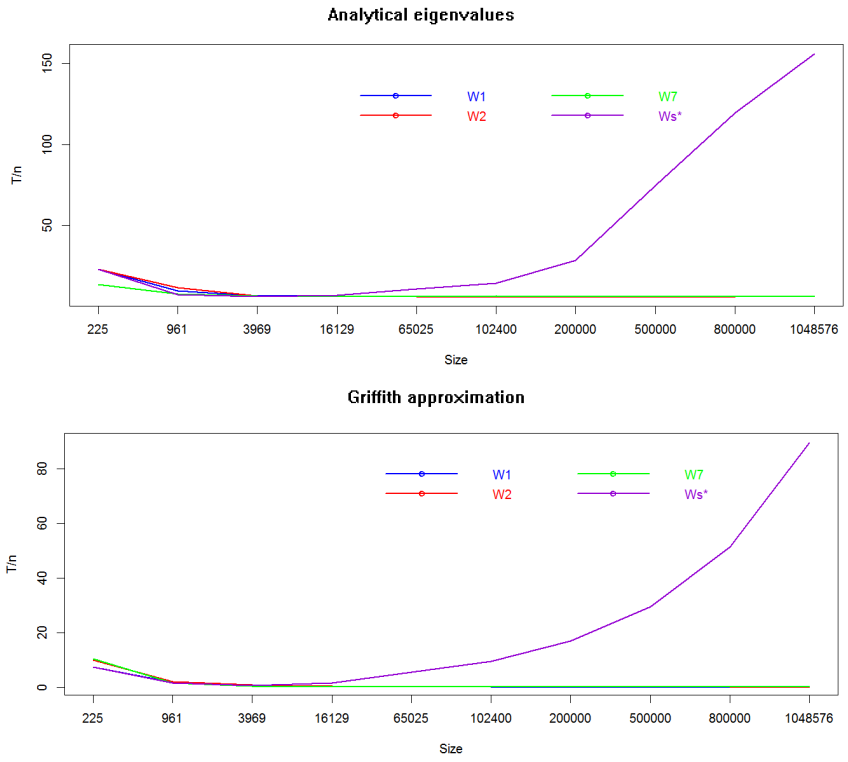


Figure 3.3: Size and sparsity effect on Analytical eigenvalues and Griffith approximations

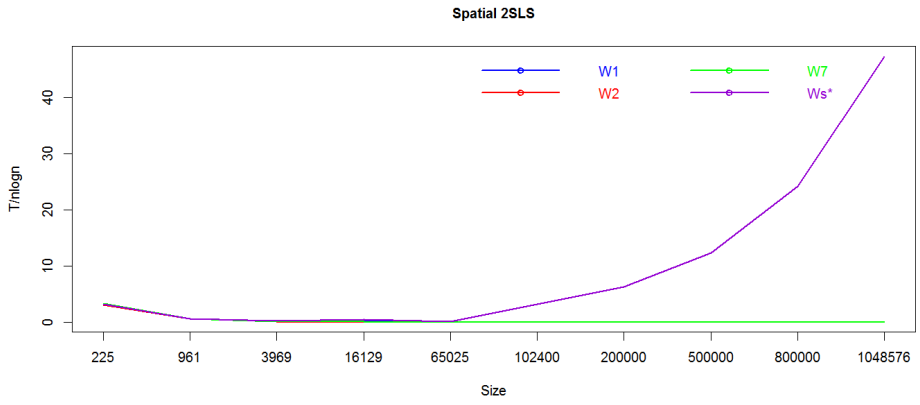


Figure 3.4: Size and sparsity effect on Spatial 2SLS

The following tables (Tables 3.8-3.11) show the summary accuracy estimates of spatial autocorrelation parameters for regular grids, rook contiguity, using different approaches.

Table 3.8: Summary accuracy estimates of spatial autocorrelation parameters for regular grids, rook contiguity, using different sparse matrix decomposition, approximation and two stage least squares, N=225

	Regular grid N=225	Median	Mean	Standard error	RMSE
Ord	ρ_1=-0.7	-0.6197	-0.6109	0.0062	0.128794
	ρ_2=-0.5	-0.4697	-0.4674	0.0060	0.095722
	ρ_3=0.3	0.2810	0.2772	0.0066	0.101592
	ρ_3=0.5	0.4559	0.4533	0.0062	0.104067
	ρ_3=0.7	0.6133	0.6139	0.0065	0.130075
Cholesky	ρ_1=-0.7	-0.6197	-0.6109	0.0061	0.127715
	ρ_2=-0.5	-0.4697	-0.4674	0.0060	0.095722
	ρ_3=0.3	0.2810	0.2772	0.0066	0.101592
	ρ_3=0.5	0.4559	0.4533	0.0062	0.104067
	ρ_3=0.7	0.6133	0.6139	0.0065	0.130075
LU	ρ_1=-0.7	-0.6197	-0.6109	0.0061	0.127715
	ρ_1=-0.5	-0.4697	-0.4674	0.0060	0.095722
	ρ_1=0.3	0.2810	0.2772	0.0066	0.101592
	ρ_1=0.5	0.4559	0.4533	0.0062	0.104067
	ρ_1=0.7	0.6133	0.6139	0.0065	0.130075
Chebyshev	ρ_1=-0.7	-0.6189	-0.6100	0.0061	0.128344
	ρ_1=-0.5	-0.4696	-0.4671	0.0060	0.095825
	ρ_1=0.3	0.2810	0.2772	0.0066	0.101592
	ρ_1=0.5	0.4557	0.4531	0.0062	0.104157
	ρ_1=0.7	0.6126	0.6129	0.0064	0.129624
MC	ρ_1=-0.7	-0.6225	-0.6093	0.0062	0.129906
	ρ_1=-0.5	-0.4681	-0.4663	0.0060	0.096102
	ρ_1=0.3	0.2793	0.2758	0.0066	0.101915
	ρ_1=0.5	0.4556	0.4533	0.0062	0.104067
	ρ_1=0.7	0.6129	0.6125	0.0065	0.131006
Analyt eigen	ρ_1=-0.7	-0.6018	-0.6095	0.0054	0.121455
	ρ_1=-0.5	-0.4714	-0.4635	0.0057	0.092965
	ρ_1=0.3	0.2680	0.2808	0.0063	0.096431
	ρ_1=0.5	0.4518	0.4566	0.0047	0.082788
	ρ_1=0.7	0.5662	0.5713	0.0059	0.156192
Griffith	ρ_1=-0.7	-0.6018	-0.6095	0.0054	0.121455
	ρ_1=-0.5	-0.4714	-0.4635	0.0056	0.091587
	ρ_1=0.3	0.2680	0.2808	0.0064	0.097901
	ρ_1=0.5	0.4519	0.4566	0.0059	0.098569
	ρ_1=0.7	0.5662	0.5713	0.0066	0.162372
2SLS	ρ_1=-0.7	-0.6572	-0.6389	0.0250	0.379407
	ρ_1=-0.5	-0.4984	-0.4508	0.0262	0.396068
	ρ_1=0.3	0.3912	0.3117	0.0262	0.393174
	ρ_1=0.5	0.5855	0.4905	0.0252	0.378119
	ρ_1=0.7	0.7490	0.6696	0.0224	0.337372

Table 3.9: Summary accuracy estimates of spatial autocorrelation parameters for regular grids, rook contiguity, using different sparse matrix decomposition, approximation and two stage least squares, N=961

	Regular grid N=961	Median	Mean	Standard error	RMSE
Ord	ρ_1=0.7	-0.6301	-0.6216	0.0020	0.068147
	ρ_1=0.5	-0.4820	-0.4747	0.0014	0.044040
	ρ_1=0.3	0.2922	0.2880	0.0014	0.043543
	ρ_1=0.5	0.4690	0.4676	0.0017	0.053750
	ρ_1=0.7	0.6495	0.6296	0.0026	0.085556
Cholesky	ρ_1=-0.7	-0.6301	-0.6216	0.0020	0.068147
	ρ_1=-0.5	-0.4820	-0.4747	0.0014	0.044040
	ρ_1=0.3	0.2922	0.2880	0.0014	0.043544
	ρ_1=0.5	0.4690	0.4676	0.0017	0.053750
	ρ_1=0.7	0.6495	0.6296	0.0026	0.085556
LU	ρ_1=-0.7	-0.6301	-0.6216	0.0020	0.068147
	ρ_1=-0.5	-0.4820	-0.4747	0.0014	0.044040
	ρ_1=0.3	0.2922	0.2880	0.0014	0.043544
	ρ_1=0.5	0.4690	0.4676	0.0017	0.053750
	ρ_1=0.7	0.6495	0.6296	0.0026	0.085556
Chebyshev	ρ_1=-0.7	-0.6292	-0.6207	0.0020	0.068288
	ρ_1=-0.5	-0.4818	-0.4745	0.0014	0.044050
	ρ_1=0.3	0.2922	0.2880	0.0014	0.043544
	ρ_1=0.5	0.4688	0.4674	0.0017	0.053763
	ρ_1=0.7	0.6486	0.6286	0.0026	0.085698
MC	ρ_1=-0.7	-0.6292	-0.6216	0.0020	0.068147
	ρ_1=-0.5	-0.4820	-0.4747	0.0014	0.044040
	ρ_1=0.3	0.2902	0.2879	0.0014	0.043546
	ρ_1=0.5	0.4684	0.4673	0.0017	0.053769
	ρ_1=0.7	0.6498	0.6291	0.0026	0.085627
Analyt eigen	ρ_1=-0.7	-0.6116	-0.6116	0.0019	0.066715
	ρ_1=-0.5	-0.4673	-0.4653	0.0014	0.044604
	ρ_1=0.3	0.2941	0.2917	0.0013	0.040369
	ρ_1=0.5	0.4704	0.4688	0.0015	0.047473
	ρ_1=0.7	0.5769	0.5844	0.0022	0.081563
Griffith	ρ_1=-0.7	-0.6061	-0.6073	0.0020	0.070593
	ρ_1=-0.5	-0.4635	-0.4616	0.0014	0.044875
	ρ_1=0.3	0.2961	0.2936	0.0013	0.040341
	ρ_1=0.5	0.4663	0.4650	0.0014	0.044625
	ρ_1=0.7	0.5701	0.5790	0.0023	0.085941
2SLS	ρ_1=-0.7	-0.6554	-0.6404	0.0054	0.170952
	ρ_1=-0.5	-0.4770	-0.4683	0.0064	0.199405
	ρ_1=0.3	0.3201	0.2971	0.0064	0.198408
	ρ_1=0.5	0.5151	0.4926	0.0056	0.173655
	ρ_1=0.7	0.7126	0.6916	0.0041	0.127171

Table 3.10: Summary accuracy estimates of spatial autocorrelation parameters for regular grids, rook contiguity, using different sparse matrix decomposition, approximation and two stage least squares, N=3969

	Regular grid N=3969	Median	Mean	Standard error	RMSE
Ord	ρ_1=-0.7	-0.5846	-0.6103	0.0009	0.034940
	ρ_2=-0.5	-0.4659	-0.4667	0.0005	0.016612
	ρ_3= 0.3	0.2907	0.2896	0.0003	0.009408
	ρ_4=0.5	0.4590	0.4657	0.0005	0.016676
	ρ_5=0.7	0.5796	0.6224	0.0011	0.040122
Cholesky	ρ_1=-0.7	-0.5846	-0.6104	0.0009	0.034929
	ρ_1=-0.5	-0.4659	-0.4666	0.0005	0.016616
	ρ_1=0.3	0.2907	0.2896	0.0003	0.009408
	ρ_1=0.5	0.4590	0.4657	0.0005	0.016676
	ρ_1=0.7	0.5796	0.6224	0.0011	0.040122
LU	ρ_1=-0.7	-0.5846	-0.6103	0.0009	0.034940
	ρ_1=-0.5	-0.4659	-0.4666	0.0005	0.016616
	ρ_1=0.3	0.2907	0.2896	0.0003	0.009408
	ρ_1=0.5	0.4590	0.4657	0.0005	0.016676
	ρ_1=0.7	0.5796	0.6224	0.0011	0.040122
Chebyshev	ρ_1=-0.7	-0.5840	-0.6095	0.0009	0.035084
	ρ_1=-0.5	-0.4657	-0.4657	0.0005	0.016676
	ρ_1=0.3	0.2907	0.2896	0.0003	0.009408
	ρ_1=0.5	0.4589	0.4655	0.0005	0.016690
	ρ_1=0.7	0.5790	0.6215	0.0011	0.040262
MC	ρ_1=-0.7	-0.5838	-0.6103	0.0009	0.034940
	ρ_1=-0.5	-0.4658	-0.4663	0.0005	0.016636
	ρ_1=0.3	0.2905	0.2896	0.0003	0.009408
	ρ_1=0.5	0.4585	0.4654	0.0005	0.016697
	ρ_1=0.7	0.5800	0.6223	0.0011	0.040137
Analyt eigen	ρ_1=-0.7	-0.6116	-0.6116	0.0009	0.035715
	ρ_1=-0.5	-0.4673	-0.4653	0.0005	0.016704
	ρ_1=0.3	0.2941	0.2917	0.0004	0.012469
	ρ_1=0.5	0.4704	0.4688	0.0005	0.016473
	ρ_1=0.7	0.5769	0.5844	0.0009	0.041263
Griffith	ρ_1=-0.7	-0.5807	-0.6093	0.0009	0.036126
	ρ_1=-0.5	-0.4601	-0.4640	0.0005	0.016796
	ρ_1=0.3	0.2962	0.2941	0.0004	0.012435
	ρ_1=0.5	0.4624	0.4660	0.0005	0.016656
	ρ_1=0.7	0.5566	0.5797	0.0010	0.045472
2SLS	ρ_1=-0.7	-0.6693	-0.6615	0.0013	0.041782
	ρ_1=-0.5	-0.4962	-0.4870	0.0016	0.049769
	ρ_1=0.3	0.3016	0.2982	0.0017	0.052703
	ρ_1=0.5	0.5019	0.4966	0.0015	0.046512
	ρ_1=-0.7	0.6383	0.6962	0.0012	0.037214

Table 3.11: Summary accuracy estimates of spatial autocorrelation parameters for regular grids, rook contiguity, using different sparse matrix decomposition, approximation and two stage least squares, N=65025

	Regular grid N=65025	Median	Mean	Standard error	RMSE
Cholesky	ρ_1=-0.7	-0.6147	-0.6148	0.00020	0.058259
	ρ_1=-0.5	-0.4672	-0.4679	0.00010	0.026530
	ρ_1=0.3	0.2944	0.2935	0.00003	0.007692
	ρ_1=0.5	0.4737	0.4719	0.00010	0.026290
	ρ_1=0.7	0.6341	0.6327	0.00030	0.081029
LU	ρ_1=-0.7	-0.6147	-0.6148	0.00020	0.058259
	ρ_1=-0.5	-0.4672	-0.4679	0.00010	0.026530
	ρ_1=0.3	0.2944	0.2935	0.00003	0.008202
	ρ_1=0.5	0.4737	0.4719	0.00010	0.026290
	ρ_1=0.7	0.6341	0.6327	0.00030	0.081029
Chebyshev	ρ_1=-0.7	-0.6139	-0.6140	0.00020	0.058396
	ρ_1=-0.5	-0.4671	-0.4678	0.00010	0.026537
	ρ_1=0.3	0.2943	0.2935	0.00003	0.008202
	ρ_1=0.5	0.4735	0.4717	0.00010	0.026301
	ρ_1=0.7	0.6331	0.6317	0.00030	0.081165
MC	ρ_1=-0.7	-0.6151	-0.6149	0.00020	0.058242
	ρ_1=-0.5	-0.4672	-0.4679	0.00010	0.026530
	ρ_1=0.3	0.2944	0.2936	0.00005	0.013811
	ρ_1=0.5	0.4738	0.4720	0.00010	0.026284
	ρ_1=0.7	0.6338	0.6327	0.00030	0.081029
Analyt eigen	ρ_1=-0.7	-0.5695	0.6117	0.00020	0.058797
	ρ_1=-0.5	-0.4513	-0.4665	0.00010	0.026367
	ρ_1=0.3	0.2916	0.2926	0.00004	0.009490
	ρ_1=0.5	0.4502	0.4695	0.00010	0.026430
	ρ_1=0.7	0.5370	0.5848	0.00020	0.064271
Griffith	ρ_1=-0.7	-0.5650	-0.6087	0.00020	0.059336
	ρ_1=-0.5	-0.4489	-0.4360	0.00010	0.028576
	ρ_1=0.3	0.2943	0.2952	0.00004	0.009203
	ρ_1=0.5	0.4479	0.4666	0.00010	0.026616
	ρ_1=0.7	0.5320	0.5806	0.00020	0.065256
2SLS	ρ_1=-0.7	-0.6670	-0.6673	0.00007	0.018919
	ρ_1=-0.5	-0.4901	-0.4903	0.00030	0.076594
	ρ_1=0.3	0.3032	0.3027	0.00009	0.023977
	ρ_1=0.5	0.5032	0.5020	0.00008	0.020659
	ρ_1=0.7	0.7027	0.7012	0.00006	0.015811

Table 3.12: Summary accuracy estimates of spatial autocorrelation parameters for regular grids, rook contiguity, using different sparse matrix decomposition, approximation and two stage least square, N=102,400

	Regular grid N=102,400	Median	Mean	Standard error	RMSE
Cholesky	ρ_1=-0.7	-0.6612	-0.6155	0.0001674309	0.100054186
	ρ_2=-0.5	-0.4842	-0.4680	0.0000793029	0.040841015
	ρ_3=0.3	0.2947	0.2938	0.0000254687	0.010240227
	ρ_4=0.5	0.4928	0.4725	0.0000906807	0.039978546
	ρ_5=0.7	0.6945	0.6338	0.0002134825	0.095127794
LU	ρ_1=-0.7	-0.6612	-0.6155	0.0001674312	0.100054237
	ρ_2=-0.5	-0.4842	-0.4680	0.0000793029	0.040841015
	ρ_3=0.3	0.2947	0.2938	0.0000254691	0.010240328
	ρ_4=0.5	0.4928	0.4725	0.0000906807	0.039978546
	ρ_5=0.7	0.6945	0.6338	0.0002134825	0.095127794
Chebyshev	ρ_1=-0.7	-0.6601	-0.6147	0.0001662833	0.100535884
	ρ_2=-0.5	-0.4841	-0.4678	0.0000791441	0.029903544
	ρ_3=0.3	0.2947	0.2938	0.0000254618	0.009719890
	ρ_4=0.5	0.4926	0.4723	0.0000904975	0.029889718
	ρ_5=0.7	0.6932	0.6329	0.0002119393	0.068160623
MC	ρ_1=-0.7	-0.6613	-0.6155	0.0001673678	0.100043375
	ρ_2=-0.5	-0.4843	-0.4679	0.0000793564	0.040930034
	ρ_3=0.3	0.2947	0.2938	0.0000254870	0.010244888
	ρ_4=0.5	0.4923	0.4725	0.0000906561	0.039972832
	ρ_5=0.7	0.6940	0.6339	0.0002135098	0.095064509
Analyt eigen	ρ_1=-0.7	-0.6655	-0.6242	0.0001658677	0.092527385
	ρ_2=-0.5	-0.4896	-0.4724	0.0000796106	0.037562737
	ρ_3=0.3	0.2949	0.2939	0.0000232032	0.009622509
	ρ_4=0.5	0.4954	0.4761	0.0000874078	0.036802964
	ρ_5=0.7	0.6426	0.5984	0.0001793835	0.116678417
Griffith	ρ_1=-0.7	-0.6640	0.6216	0.0001707822	0.095584021
	ρ_2=-0.5	-0.4860	-0.4694	0.0000771880	0.039349537
	ρ_3=0.3	0.2974	0.2965	0.0000226794	0.008072217
	ρ_4=0.5	0.4917	0.4730	0.0000848924	0.038302006
	ρ_5=0.7	0.6389	0.5944	0.0001818421	0.120565518
2SLS	ρ_1=-0.7	-0.6692	-0.6692	0.0000467531	0.034241368
	ρ_2=-0.5	-0.4945	-0.4932	0.0000578091	0.019709128
	ρ_3=0.3	0.2991	0.2983	0.0000644449	0.020692319
	ρ_4=0.5	0.4990	0.4981	0.0000555274	0.017870062
	ρ_5=0.7	0.6991	0.6982	0.0000420444	0.013574082

3.4 Conclusions and Final Remarks

In this chapter, we first analyzed the scientific contributions of this work. In particular, we focus on computational issues, in large datasets with traditional implementations. We compared the proposed methods in terms of computational complexity and accuracy on a wide range of simulated datasets by regular grids to evaluate the sensitivity of these methodologies with respect to the *sparsity* and *size effect*.
We summarize our empirical results as follows:

- **Accuracy of ML-based approach** :
  sparse matrix decompositions, Chebyshev and MC approximation can be considered roughly equivalent in terms of accuracy.
  The Griffith's approximations present values of RMSE slightly greater than other methods, for large values of ρ .

- **Accuracy of Spatial 2SLS estimation**: this method presents values of RMSE slightly less than other methods.

- **Sensitivity of the accuracy for** ρ: RMSE is greater for the extreme values of spatial autocorrelation coefficient.

- **LU decomposition**: the "sparsity effect" dominates the"size effect" *for small datasets*, while *for moderately large datasets* the complexity computational results affected by"size effect".

- **Cholesky decomposition**: the timing increases with greater sparsity but much less significantly w.r.t. "size effect" *for moderately large datasets* (as argued in Smirnov and Anselin, 2001) and *for small datasets* it appears more sensitive to "sparsity effect" .

- **MC Approximation** results influenced for *moderately large data sets* by "sparsity effect". For *large datasets* the "size effect" is greater than "sparsity effect".

- **Chebyshev Approximation** exhibites the sensitivity w.r.t."size effect" *for moderately large datasets*.

- **Griffith approximation (2004), Analytical eigenvalues and Spatial 2SLS estimation** appear to be affected by the "size effect" when increasing the sample size datasets.

- **Computational difficulties**: Ord approach fails for $n > 16129$.
  For $n > 500,000$ sparse matrix decomposition and approximation techniques are not feasible computation.

Griffith approximation (2004), Analytical eigenvalues and Spatial 2SLS estimation are feasible computed from $n = 225$ *to* $n = 1048576$.

Chapter 4

Spatial Econometric Modelling of massive datasets: the contribution of Data Mining

4.1 Introduction

In this chapter we present the contribution of the spatial data mining on spatial econometric models of massive datasets.[1] We propose a data mining methodology that explicitly considers the phenomenon of spatial autocorrelation on prediction errors.
We suggest some directions along which spatial econometric modeling could benefit from the cross-fertilization spatial data mining techniques such as Classification and Regression Trees (CART). We use the CART algorithm to fit empirical data and produce a tree with optimal tree size for different specifications of spatial econometric models.
We also examine some diagnostic measures to evaluate the spatial autocorrelation of the pseudo-residuals obtained from the regression tree analysis and we compare the accuracy and performance of different versions of CART that take into account the effects of spatial dependence.
To address this issue, we start examining a non-spatial regression tree, then we include the geographical coordinates of data in the covariate set and finally, we consider one of the most common spatial econometric models: Spatial Lag combined with two versions of regression trees: non-spatial regression tree and geographical coordinates-based regression tree.
This allows us to determine the strength and the possible role of spatial arrangement on the variables in the predictive model and reduce the effect of spatial autocorrelation on prediction errors. In particular, we test the sensibility of various regression trees with different spatial weights matrix specifications such that to remove the spatial autocorrelation on pseudo-residuals and to improve the accuracy of spatial predictive models, saving in terms

[1] Part of the results of this work have been presented at *53rd European Regional Science Association Congress*, **http://ideas.repec.org/p/wiw/wiwrsa/ersa13p1004.html**

of computational complexity. This chapter is organized as follows. Section 2 describes the motivation for this work by reporting the related work from research lines. The focus of Section 3 is the analysis of different versions of CART to compare the performance and evaluate the spatial autocorrelation of prediction errors of the regression trees (pseudo-residuals). Finally, Section 4 reports some concluding remarks and future works.

4.2 Related works and our contribution

In the literature, several approaches have been proposed that face the treatment of spatial data in classification and regression data mining methods. In the following section, we report several related works that deal this problem in different data mining tasks.

Spatial association rule mining. Many existing co-location mining algorithms were developed for spatial data. Seminal work on spatial association rules discovery is that of Koperski and Han (1995) for extraction of multi-level spatial association rules by a progressive deepening of the levels. Shekhar et al. (2001) discuss several interesting approaches to mine co-location patterns, which are subsets of Boolean spatial features whose instances are frequently located together in close proximity. Huang et al. (2004) define the spatial co-location rule and propose an algorithm to find it for spatial application domains.
Xiao et al. (2008) proposed a density based approach to searching for co-location instances. Zhang et al. (2004) enhanced searching for co-location patterns proposed in Shekhar et al. (2001) by presented an approach to find spatial star, clique, and generic patterns. Morimoto (2001) discovers frequent patterns in spatial database by grouping neighborhood class sets using a support count measure as a means to determine frequency.

Spatial classification data mining. In spatial classification data mining tasks, most of the approaches focus on the phenomenon of spatial autocorrelation classification process.
Zhao and Li (2011) have adapted classification tree for handling geographical data by proposing a spatial entropy-based decision tree which captures 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. The notion of spatial entropy provides an integration of spatial dimension in classification tree algorithm.
The *Spatial Entropy* is defined as follows (Li and Claramunt, 2006):

$$Entropy_s(A) = -\sum_i \frac{d_i^{int}}{d_i^{ext}} P_i log_2(P_i) \tag{4.1}$$

where C is the set of spatial entities of a given dataset, C_i denotes the subset of C whose entities belong to the i-th category of the classification, d_i^{int} is the "intra-distance", the average distance between the entities of C_i, d_i^{ext} is the "exta-distance", the average distance between the entities of C_i and P_i is the proportion of entities labeled with value i over the

total number of entities.
At each level of such a spatial form of a decision tree, the supporting attribute that gives the maximum spatial information gain is selected as a node. This guarantees that the spatial entities of a same category are preferably aggregated.
Rinzivillo and Turini (2007) also redefined the classical information gain used in a standard decision tree induction procedure.
The *Spatial information gain* is defined by:

$$Gain = -\sum_{l} \frac{mes(S|c_i)}{mes(S)} log_2 \frac{mes(S|c_l)}{mes(S)} - \sum_{j} \frac{mes(S|v_j)}{mes(S)} H(S|v_j) \tag{4.2}$$

where $mes(t)$ is the aggregated spatial measure of the spatial transaction t, S is a set of spatial transactions whose class label attribute has l distinct classes $(i.e., c_1, c_2, ..., c_l)$ and $S|c_i$ is the set of transactions labeled by c_i; V is an attribute that has q distinct values $(i.e., v_1, v_2, ..., v_q)$.
This measure is then used 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) adapted the Breiman's classification trees (Breiman et al., 1984) to the case of spatially dependent samples, focusing on environmental and ecological applications. They modified the algorithm to take into account the irregularity of sampling by weighting the data according to their spatial pattern. Two approaches were considered: the first one considers the irregularity of the sampling by weighting the data according their spatial pattern. The idea is to "decluster" the data based on the Kriging method.
The second approach that uses spatial estimates of the quantities involved in the construction of the discriminant rule.
Let be $\{ X^1 (s_\alpha), \ldots, X^p(s_\alpha), Y(s_\alpha)\}, \alpha = 1, \ldots, n$ the samples that are originated from random fields $\{X^1 (), \ldots, X^p(), Y()\}$ on some domain $\mathcal{D} \in \mathbb{R}^2$ and explicity take into account the dependence structure on these fields.
The basic idea of the first approach is to weight the samples such that clustered data have less weight than sparse data. In particular we have:

$$\widehat{p}\ (j \mid t) = \frac{1}{\sum_{\alpha \in t} w_\alpha} \sum_{\alpha \in t}^{n} w_\alpha \mathbb{I}\{ Y(s_\alpha) = i\}$$

and
$\hat{R}(T) = \sum_{\alpha=1}^{n} w_\alpha \mathbb{I}\{ T (X^1 (s_\alpha), \ldots, X^p(s_\alpha)) \neq Y(s_\alpha)\}$, s.t. $\sum_{\alpha=1}^{n} w_\alpha = 1$
For determining these wieghts they consider the method related to geostatistics (L. Bel, D. Allard, J.M. Laurent, R. Cheddadi, A. Bar-Hen, 2009).
If the covariance function $C(\cdot)$ of a random field $Z(\cdot)$ is know, the best linear unbiased predictor of a regional average on a 2d domain $\mathcal{D}$ is the so called **Kriging of a regional**

average (or kriging of the mean if $\mathcal{D} \rightarrow \mathbb{R}^2$) . Let us denote $Z_{\mathcal{D}}$ the average $Z(\cdot)$ over $\mathcal{D}$. The kriging of $Z_{\mathcal{D}}$ is the quantity $\widehat{Z_{\mathcal{D}}} = \sum_\alpha w_\alpha \, Z\ (s_\alpha\ )$, such that $\mathbb{E}\ \left(\widehat{Z_{\mathcal{D}}} - Z_{\mathcal{D}}\right) = 0$ and $var\ \left(\widehat{Z_{\mathcal{D}}} - Z_{\mathcal{D}}\right)$ is minimum. The vector $W = (w_1,\ \ldots,\ w_n)^T$ is the solution of the system

$$\begin{pmatrix} \mathbf{C} & \mathbf{1} \\ \mathbf{1^T} & 0 \end{pmatrix} = \begin{pmatrix} W \\ v \end{pmatrix} \begin{pmatrix} C_{\mathcal{D}} \\ 1 \end{pmatrix}$$

where $\mathbf{C}$ is the matrix whose α, β element is $C(s_\alpha,\ s_\beta)$, $C_{\mathcal{D}}$ is the vector with elements

$$C(s_\alpha, \mathcal{D}) = \frac{\mathbf{1}}{|\mathcal{D}|} \int_{\mathcal{D}} C(s_\alpha,\ s) ds$$

$\mathbf{1}$ is a vector of ones of length n and v is the Lagrange parameter associated with the unbiasedness condition. To evaluate the last integral we define a grid G on $\mathcal{D}$ and use the following approximation

$$C(s_\alpha, \mathcal{D}) = \frac{\mathbf{1}}{|\mathcal{D}|} \int_{\mathcal{D}} C(s_\alpha,\ s) ds \ \cong\ \frac{\mathbf{1}}{n_G} \sum_{s_\beta\ \in G} C(s_\alpha,\ s_\beta\ ).$$

This method consists in applying the kriging paradigm to the regional average of $Y(\cdot)$ and use the resulting kriging weights in the CART algorithm. The solution of the adpated kriging system is

$$\min_W var\ (W^T Y - Y_{\mathcal{D}}) \quad \text{with } \mathbf{1}^T\ W = 1 \text{ and } w_\alpha\ \geq 0$$

where $Y = (Y_1,\ \ldots,\ Y_n)^T$.
Finally, this approach allows to reduce the bias of the regression tree by taking into account the spatial redundancy of the data.
Instead of simply introducing weights, a second approach consists in deriving spatial estimates for all quantities involved the algorithm: proportions in leaves, Gini index and empirical risk.
In the case of two classes, only one parameter describers the proportion of each class and the variance of the indicator of, say class 1, and hence

$$D = 2p\,(1-p) = 2\sigma^2$$

Consider a leaf t of the tree T. The theoretical proportion $p\,(j \mid t)$ of class j in t is the conditional probability $P\ (Y = j |\, X\ \in\ B_t) = \mathbb{E}\{\mathbb{I}\ (Y = j |\, X\ \in\ B_t)\}$, where B_t is thee subdomain of $\mathbb{R}^p$ corresponding to the leaf t. It can thus be estimates by kriging the spatial average of the variable $\mathbb{I}\ \{(Y\,() = j |\, X\ (\cdot) \in\ B_t\}$ over the domain D_t defined as the set of location $s \in\ D$ such that $(X^1\ (s)\,,\ \ldots,\ X^p\ (s)) \in B_t$.
Applying the kriging approach on the estimation of $p\,(j \mid t)$ leads to

$$\widehat{p}\ (j \mid t) = \sum_{\alpha : X(s_\alpha\ ) \in\ B_t}^{n} \lambda_\alpha \mathbb{I}\{\ Y\,(s_\alpha\ ) = i\}$$

where λ_α is the solution of the system of n_t equations with $\alpha : X(s_\alpha) \in B_t$:

$$\sum_{\beta : X(s_\beta) \in B_t} \lambda_\beta C_j(s_\alpha, s_\beta) = \frac{1}{|D_t|} \int_{D_t} C_j(s_\alpha, s) ds \tag{4.3}$$

under the constraint $\sum_\alpha \lambda_\alpha = 1$. In the above equations, $C_j(s, s')$ is the covariance function of $\mathbb{I}\{(Y(s) = j | X(s) \in B_t\}$.
The Gini index is then computed from the estimated proportions:

$$\widehat{D}_t = 1 - \sum_i \widehat{p}\ (i \mid t)^2 \tag{4.4}$$

In this setting, the empirical risk is also estimated by kriging the spatial average of the variable $\mathbb{I}\{T\ (X() \neq Y()\}$ on D. Notice that the domain D_t is not known, it is approximated by the convex hull of the sample points lying in D_t and the integral is computed on the points of the grid G falling within that convex hull.

Proposition 3. *(L. Bel, D. Allard, J.M. Laurent, R. Cheddadi, A. Bar-Hen, 2009). For the model described above, let us denote D the population Gini index in $\mathcal{D}$, and $\hat{D}$ its estimate computed from (4.4). Let us further denote $n(\mathcal{D})$ the number of samples in $\mathcal{D}$. Assume that the density of samples tends to a strictly positive quantity as the domain increases: $\frac{n(\mathcal{D})}{|\mathcal{D}|} \to \lambda > 0$ as $\mathcal{D} \to \mathbb{R}^2$. Then, as $\mathcal{D} \to \mathbb{R}^2$ $E[\hat{D}] \to D$*

Proof. The estimated proportion $\hat{p}_j$ are obtained from the solution of the kriging equation (4.3)

$$\hat{p}_j = Y^T \Lambda_j, \qquad \Lambda_j = C_j^{-1} \left(C_{j,\mathcal{D}} + \frac{1 - \mathbf{1}\mathbf{C}_j^{-1} C_{j,\mathcal{D}}}{\mathbf{1^T} \mathbf{C}_j^{-1}\ \mathbf{1}} \mathbf{1} \right)$$

where $\mathbf{C}_j$ and $C_{j,\mathcal{D}}$ are the matrix that corresponding respectively to the left-hand side and right-hand side of (4.3). It is easy to show that $\mathbb{E}(\hat{p}_j) = p_j$ and $\left(\hat{p}_j^2\right) = {\Lambda_j}^T\ \mathbf{C}_j \Lambda_j$. Hence, we have:

$$[\widehat{\mathrm{D}}] = \mathrm{D} - 2 \sum_{\mathrm{j}} {\Lambda_j}^T\ \mathbf{C}_j \Lambda_j \tag{4.5}$$

After straightforward developments, each term of the sum in (4.5) is seen to be equal to twice ${C_{\mathcal{D},j}}^T \mathbf{C}_j^{-1} C_{\mathcal{D},j} + \{1 - (\mathbf{1}^T \mathbf{C}_j^{-1} C_{\mathcal{D},j})^2\} / \mathbf{1}^T \mathbf{C}_j^{-1} \mathbf{1}$. But, as $\mathcal{D} \in \mathbb{R}^2$, each element of the vector $C_{\mathcal{D},j} \to 0$ by the ergodic assumption; and $\mathbf{1}^T \mathbf{C}_j^{-1} \mathbf{1} \to \infty$ as $\mathcal{D} \in \mathbb{R}^2$ because the number of samples in $\mathcal{D}$ tends to infinity. Hence $\mathbb{E}\ [\widehat{\mathrm{D}}] \to D$ as $\mathcal{D} \in \mathbb{R}^2$.

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). In particular, they consider two alternative solutions for associative

classification: a propositional and a structural method. In the former, the classifier obtains a propositional representation of training data even in spatial domains which are inherently non-propositional, thus allowing the application of traditional data mining algorithms. In the latter, the Bayesian framework is extended following a multirelational data mining approach in order to cope with spatial classification tasks. Both methods were evaluated and compared on two real-world spatial datasets. The obtained results show that the use of different levels of granularity permitted to find the best tradeoff between bias and variance in spatial classification.

Spatial regression data mining. The problem of dealing with spatial autocorrelation in regression data mining tasks is faced by Malerba et al. (2005) that present a relation regression method to detect global and local effects over spatial data based on tight-integration between spatial regression method and spatial database systems. The stepwise mining faces the spatial need of distinguishing among explanatory attributes that have some global effect on the response attribute and others that have only local effect. Both splitting and regression nodes may involve several layers and spatial relationships among them.

Stojanova et al. (2011) propose a data mining method based on the concept of predictive clustering tree. In the predictive clustering trees (PCTs), a decision tree is viewed as a hierarchy of clusters. The principal difference from standard decision tree is that the PCTs select the best test by maximizing the (inter-cluster) variance reduction defined by:

$$\Delta_X(E,P) = Var(E) - \sum_{E_k \in P} \frac{|E_k|}{|E|} Var(E_k) \tag{4.6}$$

where E represent the examples in t and P defines the partition $\{E_1, E_2\}$ of E. By approriately defining the variance and predictive function PCTs can be applied to different domains and data mining tasks. In this paper, the authors describe the top-down induction algorithm for building *Spatial PCTs*. In particular, the best test is based on linear combination of the variance reduction and the measure of spatial autocorrelation, as follows:

$$h = \frac{\alpha}{|T|} \sum_{T \in Y} \Delta_T(E,P) + \frac{1-\alpha}{|T|} \sum_{T \in Y} S_T(E,P) \tag{4.7}$$

where $S_X(P,E)$ can be defined in terms of both Moran's I and Geary's C. In the case of Moran's I:

$$S_X(P,E) = \frac{1}{|E|} \sum_{E_k \in P} |E_k| \cdot \widehat{I_X}(E_k) \tag{4.8}$$

where $\widehat{I_X}(E_k)$ is the scaled Moran's I computed on E_k. In implementation of this algorithm they consider different sizes of neighbourhoods and different weighting schemes to obtain the optimal combination of them.

We can find further devolmpents and extensions in the following approach in the paper of Stajonova et al. (2013). The system that they propose is called **SCLUS**, for Spatial

Predictive Clustering System. It explicitly considers spatial autocorrelation when learning predictive clustering models. Their main contributions are:

- the Spatial PCTs is applied to classification and regression tasks;
- 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;
- an extensive evaluation of the effectiveness of the proposed approach on classification and regression problems in real-life spatial data;
- comparison of SCLUS with respect to predictive models.

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.

Finally, by considering the framework of spatial econometrics, Postiglione et al. (2010), propose an interesting application of regression tree, in the regional economics for both the classical and the spatial β-convergence model, in order to identify convergence clubs in European regions.
The proposed algorithm selects the splits, which maximize the difference between the parameters of the model in different sub-samples of regions. Starting from the statistical model suggested to measure beta-convergence in a cross of economies and by considering the SAR (spatial autoregressive) and SEM (spatial error) models, the identification of convergence clubs is based on the modification of criterion split. It is no longer the sum of squared residuals but the difference among the parameters of the model considered. This choice is coherent with the definition of convergence clubs, as expressed in terms of stationarity of model parameters: they can identify two different groups of geographical units if they show statistical significant difference in the parameters estimates of the model. The objective function is the following:

$$S = (\theta_B - \theta_{\bar{B}})^T (\sum_{B} + \sum_{\bar{B}})^{-1} (\theta_B - \theta_{\bar{B}}) \tag{4.9}$$

This statistic follows a Chi Squared distribution with d degress of freedom, d being the size of θ.
The stopping criteria are:

- the last optimal probability value exceeds $\tilde{p}$;
- further disaggregations of any current club generate sub-clubs whose cardinalities are less than a certain minimum club size;

- the constraint about the maximum number of clubs is active.

Following this approach, the proposed algorithm represent a statistical tool to segment regions in clubs. In particular, they test the convergence rate in different versions: regression tree based on classical beta-convergence model, regression tree using the filtered SAR and SEM specifications, in order to compare the geographical configuration.
In **our approach**, we extend the methodology of CART in the framework of spatial econometric models in large datasets. The contribution of this work is to evaluate the effect of including spatially lagged variables, geographical coordinates or a combination of them in the set of predictors of regression tree, in terms of spatial autocorrelation among pseudo-residuals.
To this end, we test several versions of CART and we compare the accuracy and performance of non-spatial and spatial regression tree to predict the response variable in the context of spatial database. In particular, we assess the sensibility of various predictive models with different spatial weights matrix specifications such that to remove the spatial autocorrelation on pseudo-residuals. Furthermore, in this work we show the significant reduction of computational complexity with respect to the traditional spatial econometric approach without losing the accuracy prediction.
The implementation is based on the package "*rpart*" (Therneau et al., 2012) in R version 3.0.1 [2], to build a decision tree on data with minimum prediction error. Pruning for the over-fit regression tree used the highest cross-validation error less than one standar error above the minium cross-validation error. The complexity table shows the rpart regression tree formula that constructed the tree, the variables actually used in the tree, the root node error, sample size and summary statistics for different sized regression trees. The minimum "xerror" or cross validation error was added to the "xstd" (standard devation) creating the one standard error (1-SE) bar. The resulting value was then to determine the proper number of splits of optimal tree. In addition to this value was also determined by plotting the cross-validation relative error against the cost-complexity parameter (cp-value).
To evaluate the accuracy of the fit it was determinated the apparent and X-relative R^2, where the first is derived by subtracting the relative error by one and the second is determined by subtracting one from the cross-validation error.
Finally, we calculate for different versions of CART the pseudo-residuals by function "*residuals.rpart*" (residuals from a fitted Rpart object).

4.3 Empirical Analysis

In this section we present several versions of non-spatial and spatial regression trees based on geographical coordinates and spatially lagged variables.
Our approach to spatial prediction is based on both non-spatial propertiers of CART and on attributes and function describing spatial relations and spatial proximity beetween the

[2]The associated source code is available from the author upon request.

objects.
We compare the performances of different versions of CART taking into account the effect of spatial dependence.
In this empirical part our contribution is *analyzing the pseudo-residuals of regression tree looking at their spatial features* (like, e.g. spatial autocorrelation) to see whether they contain some addition hidden information.

4.3.1 Datasets

In order to deal the spatial feature of pseudo-residuals and to test the computational properties of the methodology, we employ simulated regular grid lattice of various size and real datasets.

- **Real datasets**:

  1. **US Southern county homicides**: the dataset, used by Anselin (2007) is composed by 1,412 Souther US counties (Washington D.C., Texas, Oklahoma, Arkansas, Louisiana, Mississippi, Alabama, Tennessee, Kentucky, Georgia, South Carolina, North Carolina, Florida, Virginia, West Virginia, Maryland and Delaware) and 7 variables (pertaining to 1960) as follows:

Name	Description
FIPSNO	Code
HR60	Homicide Rate per 100,000
RD60	Resource Deprivation/Affluence Component (principal component: percent black, log of median family income, gini index of family income inequality, percent of families female headed (percent of families single parent for 1960) and percent of families below poverty (percent of families below 3,000 dollars for 1960)
PS60	Population Structure Component (principal component: log of population and the log of population density)
UE60	Percent of civilian labor force that is unemployed
DV60	Percent of males 14 and over who are divorced
MA60	Median age

Source: https://geodacenter.asu.edu/sdata

  2. **California Census Block Groups Housing**: it consists from 20,640 observations using all the block groups in California from 1990 Census. The response variable is the logarithm of median house value, measured in each neighborhood. The predictor variables are economics and demographics, such as median income, house densiting, average occupacy. Also included as predictors the location (latitude and longitude) of

each neighborhood and several covariates refleting the properties of houses in neighborhood as follows:

Name	**Description**
LAT	Latitude
LONG	Longitude
ln(PRICE)	Log median house value (response variable)
MEDIAN INCOME	Median income
MEDIAN INCOME2	Square of median income
MEDIAN INCOME3	Cube of median income
Ln(median age)	Log median age
ln(TOTAL ROOMS/ POPULATION)	Log per capita rooms
ln(BEDROOMS/ POPULATION)	Log per capita bedrooms
ln(POPULATION/ HOUSEHOLDS)	Log population per household
ln(HOUSEHOLDS)	Log households

Source: Pace and Barry (1997), "Sparse Spatial Autoregressions", Statistics and Probability Letters.

- **Regular grid simulated datasets**

We conduct a series of experiments by Monte Carlo simulations for various regular lattice structures. The regular lattices range from 15×15 ($n = 225$) to 1024×1024 ($n = 1048576$). We proceed as described in previous chapter. In first step we generate the explanatory variables x_{ik} from the Uniform distribution U (0,10) for $i = 1, ..n, k = 1, .., 20$. The vector of parameters $\beta_k = 1$ for $k = 0, .., 20$. Further, we generate the error terms ϵ which are assumed to be normally i.i.d. distributed, with mean 0 and variance 1.
In second step, we compute the dependent variable $\mathbf{y}$ for the spatial lag model
$\mathbf{y} = \rho\mathbf{W} + X\beta + \epsilon$, using $\mathbf{W}$, X and the errors obtained in the previous step. In particular we consider a single value of spatial parameter $\rho = 0.5$.

4.3.2 Results

1. **US Southern county homicides**

The spatial distribution of the homicide rate is shown in the following map.

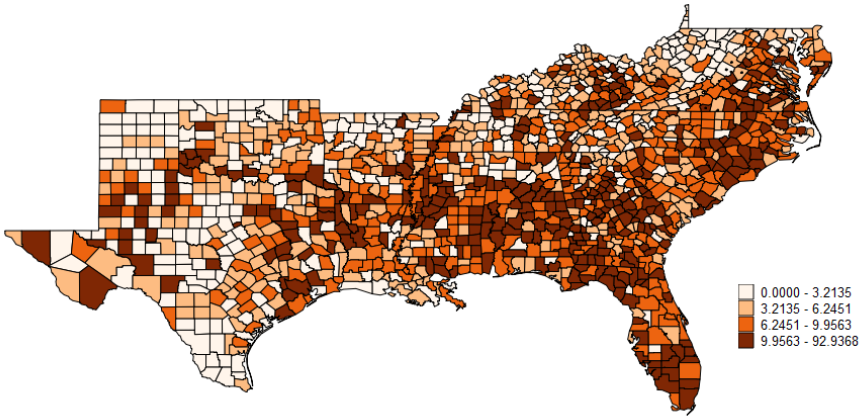


Figure 4.1: Map of homicide rate (HR60)

In particular we test four different versions of regression tree (RT) to predict the response variable: Homicide Rate (HR60).

Model	Set of predictors
Non-Spatial	resource deprivation,population structure, labour force unemployed, divorced rate, median age
Geographical Coordinates-based Spatial RT	resource deprivation, population structure, labour force unemployed, divorced rate, median age, coordx, coordy
W-based Spatial RT	resource deprivation,population structure, labour force unemployed, divorced rate, median age , spatially-lagged-homicide rate
Geographical coordinates + W-based Spatial RT	resource deprivation, population structure, labour force unemployed, divorced rate, median age, spatially-lagged-homicide rate, coordx, coordy

In the W-based spatial regression tree to construct spatially lagged response variable, we consider different spatial weights matrices in order to check the "robustness" of pseudo-

residuals spatial autocorrelation for each model. In particular we compute the following spatial weights matrices (row-standardization):

1. first-order contiguity (*rook*): the elements of which are $w_{ij} = 1$ when i and j share common border;

2. first and second order contiguity ($rook1-2$): it is a cumulative matrix that includes first and second order contiguity;

3. queen contiguity (*queen*): the elements of which are $w_{ij} = 1$ when i and j share common borders and common corners;

4. distance based contiguity: $dk1$, $dk2$, $dk3$, $dk4$, $dk5$ based on the minimum distance needed to make sure that all the areas are linked to at least k neighbours $\{k = 1, 2, 3, 4, 5\}$.

In order to check the influence of these matrices, in the Table 1 we present the summary measures for spatial weigths matrices: *number of regions, total number of links and average number of links*:

Table 4.1: Summary measures for different spatial weights matrices

Weigths matrix	n	total links	average number of links
rook	1412	7700	5.45
rook1-2	1412	23768	16.83
queen	1412	8096	5.73
dk1	1412	27648	19.58
dk2	1412	78432	55.55
dk3	1412	142394	100.85
dk4	1412	159558	113.00
dk5	1412	165048	116.89

The first version of regression tree is the "**non-spatial regression tree**" (Figure 4.2):

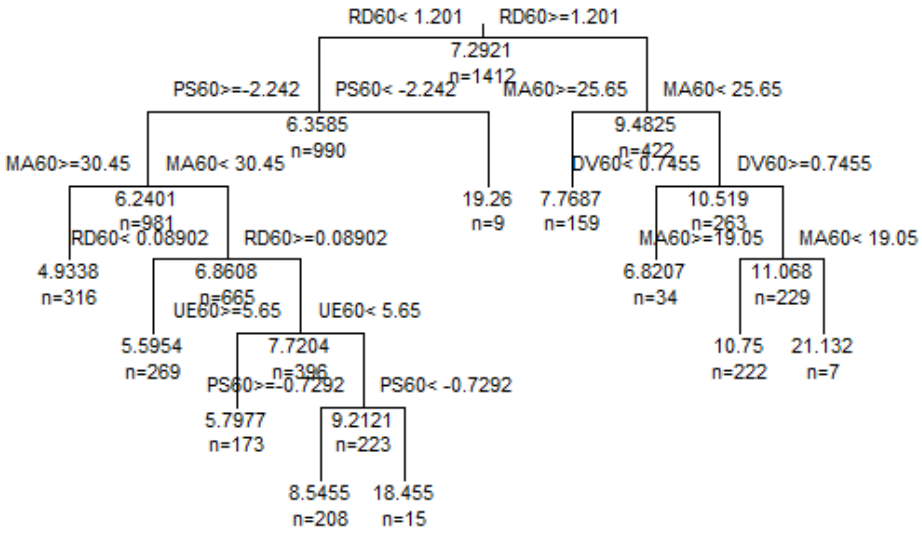
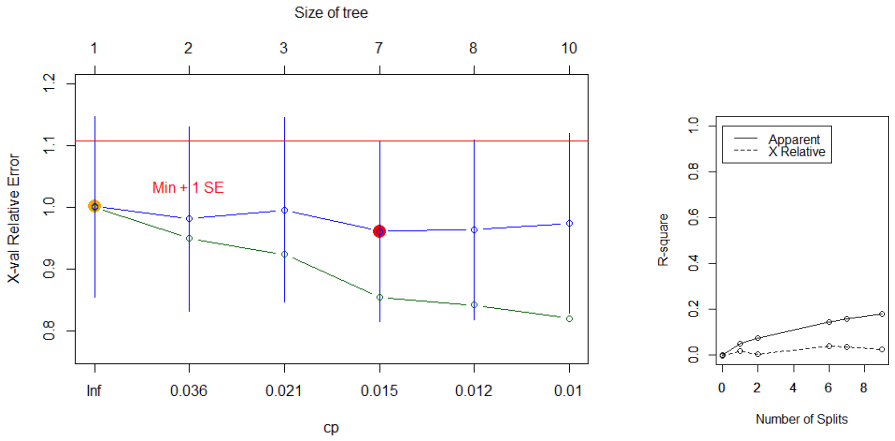


Figure 4.2: The non-spatial regression tree

The following plots show respectively the cross validation results and the "pseudo R-square" for different splits (Figure 4.3):



(a) (b)

Figure 4.3: (a) Cross validation results of Non-spatial Regression Tree (blu line: trend of xerror, green line: trend of relative error, red line: 1-SE bar) ; (b) Apparent, X-Relative R-Square and Cross Validation Relative Error graphs of Non-spatial Regression Tree (Apparent R^2= 1-relative error; X relative R^2=1- xerror)

The quantile map of pseudo-residuals suggests the possible presence of spatial clusters.

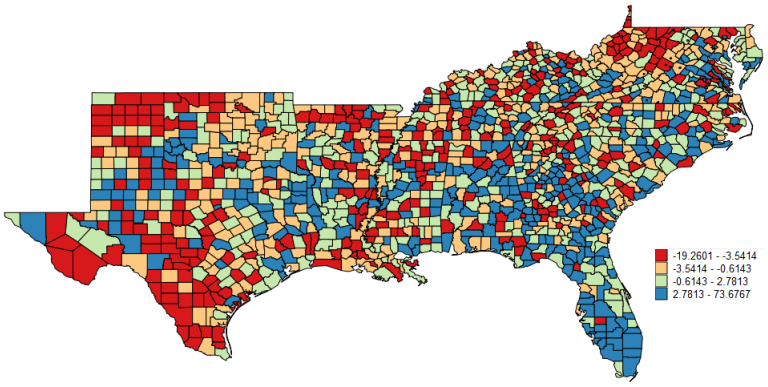


Figure 4.4: Quantile map of pseudo-residuals of Non-spatial Regression Tree

We also, note that in geographical coordinates-based spatial regression tree, the quantile map of pseudo-residuals shows still a spatial structure of pseudo-residuals.

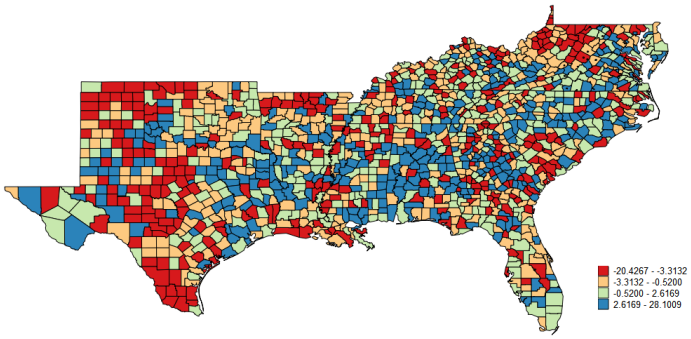


Figure 4.5: Quantile map of pseudo-residuals of geographical coordinates-based Spatial Regression Tree

We summarize the performance of different versions and the presence of pseudo-residuals spatial autocorrelation of regression tree. The Table 4.2 compares the values of permutational Moran's I on pseudo-residuals of non spatial regression tree (without geocoords) and regression tree based on geographical coordinates (with geocoords) using different spatial weights.

Table 4.2: Permutational Moran's I on pseudo-residuals of non-spatial regression tree and spatial regression tree based on geographical coordinates

Permutational Moran's I		
Weights matrix	Without geocoords	With geocoords
rook	0.1452	0.0989
	(0.001*)	(0.001*)
rook1-2	0.1340	0.0998
	(0.001*)	(0.001*)
queen	0.1357	0.0971
	(0.001*)	(0.001*)
dk1	0.1273	0.0897
	(0.001*)	(0.001*)
dk2	0.1022	0.0579
	(0.001*)	(0.001*)
dk3	0.0838	0.0406
	(0.001*)	(0.001*)
dk4	0.0776	0.0366
	(0.001*)	(0.001*)
dk5	0.0761	0.0356
	(0.001*)	(0.001*)

Notes: number of simulations=999, pseudo-pvalue in brackets, "*" statistically significant at 0.05 level.

Now, we check and compare the critical threshold distance that allows to remove the spatial autocorrelation on pseudo-residuals of non-spatial regression tree for different distance that includes at least k neighbours ($k = 1, 2, 3, 4, 5$) and we show the trend of pseudo-pvalue with respect to critical distance.

Table 4.3: Critical threshold distance such that to remove the spatial autocorrelation on pseudo-residuals of non-spatial regression tree

k	Threshold distance	Average number of links	Permutational Morans' I
≥ 1	2147.491	1383.572	-0.000471
			(0.052)
≥ 2	2167.518	1386.317	-0.000505
			(0.057)
≥ 3	2231.966	1394.048	-0.000538
			(0.062)
≥ 4	2247.105	1395.623	-0.000536
			(0.056)
≥ 5	2151.765	1384.183	-0.000513
			(0.084)

Notes: number of simulations=999, pseudo-pvalue in brackets, "*" statistically significant at 0.05 level

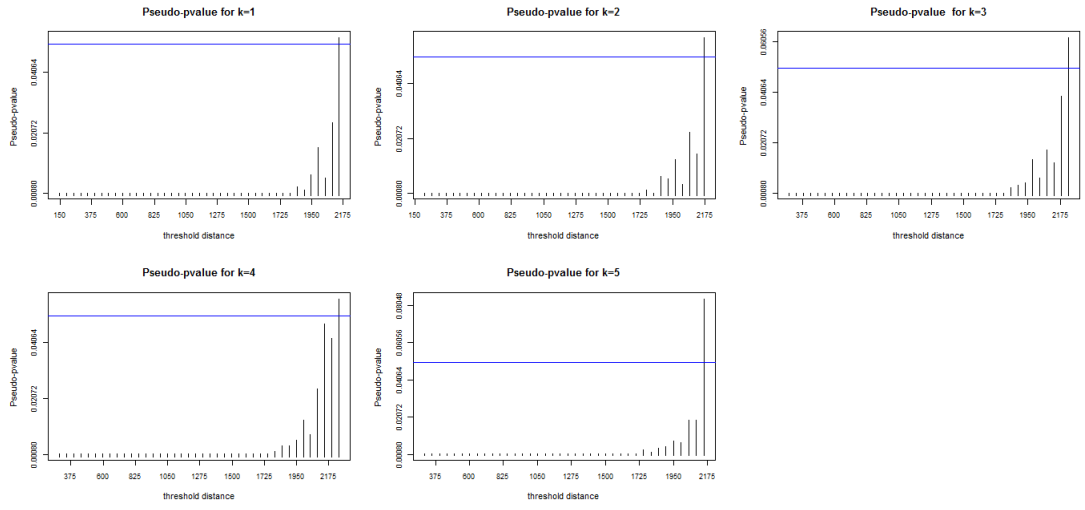


Figure 4.6: The trend of pseudo-pvalue on threshold distance of non-spatial tree (the blue bar indicates the significance level at 0.05)

We also, note that the inclusion of geographical coordinates in non-spatial version of

regression tree leads to a decrement of Permutational Moran's I for any spatial matrix and an improvement of accuracy, in particular the Apparent Rsquare increases from 0.180 to 0.385. The table 4.4 evaluates the permutational Moran's I on the pseudo-residuals of non-spatial regression tree and spatial regression tree based on the geographical coordinates, when we include in the set of predictors a specific lag, using different spatial weights.

Table 4.4: Comparison of Permutational Moran's I on pseudo-residuals of spatial lag combined with geographical coordinates regression tree

Spatial lag	Permutational Moran's I		Apparent Rsquare	
	without geocoords	with geocoords	without geocoords	with geocoords
rook	-0.0849	-0.0849	0.275	0.275
	(0.001*)	(0.001*)		
rook1-2	-0.0361	-0.0361	0.447	0.452
	(0.001*)	(0.001*)		
queen	-0.0891	-0.0871	0.267	0.287
	(0.001*)	(0.001*)		
dk1	-0.0301	-0.0424	0.391	0.421
	(0.001*)	(0.001*)		
dk2	-0.01	-0.0086	0.426	0.469
	(0.032*)	(0.066)		
dk3	0.0041	0.0054	0.388	0.464
	(0.885)	(0.922)		
dk4	0.0082	0.0076	0.459	0.461
	(0.983)	(0.964)		
dk5	0.0024	0.0021	0.350	0.433
	(0.802)	(0.788)		

Notes: number of simulations=999, pseudo-pvalue in brackets, "*" statistically significant at 0.05 level

As can be seen in Table 4.4, in geographical coordinates-based spatial regression tree, the inclusion of spatially lagged response variable by using spatial weights matrix that includes at least two neighbours (dk2), allows to remove the presence of pseudo-residuals spatial autocorrelation. We can also, note that the critical threshold distance such that to remove the spatial autocorrelation on pseudo-residuals is 167.518 and average numbers of links is 55.546, much lower than the threshold distance in the case of non-spatial regression tree (Table 4.3).

The predictive spatial regression tree selected is the following:

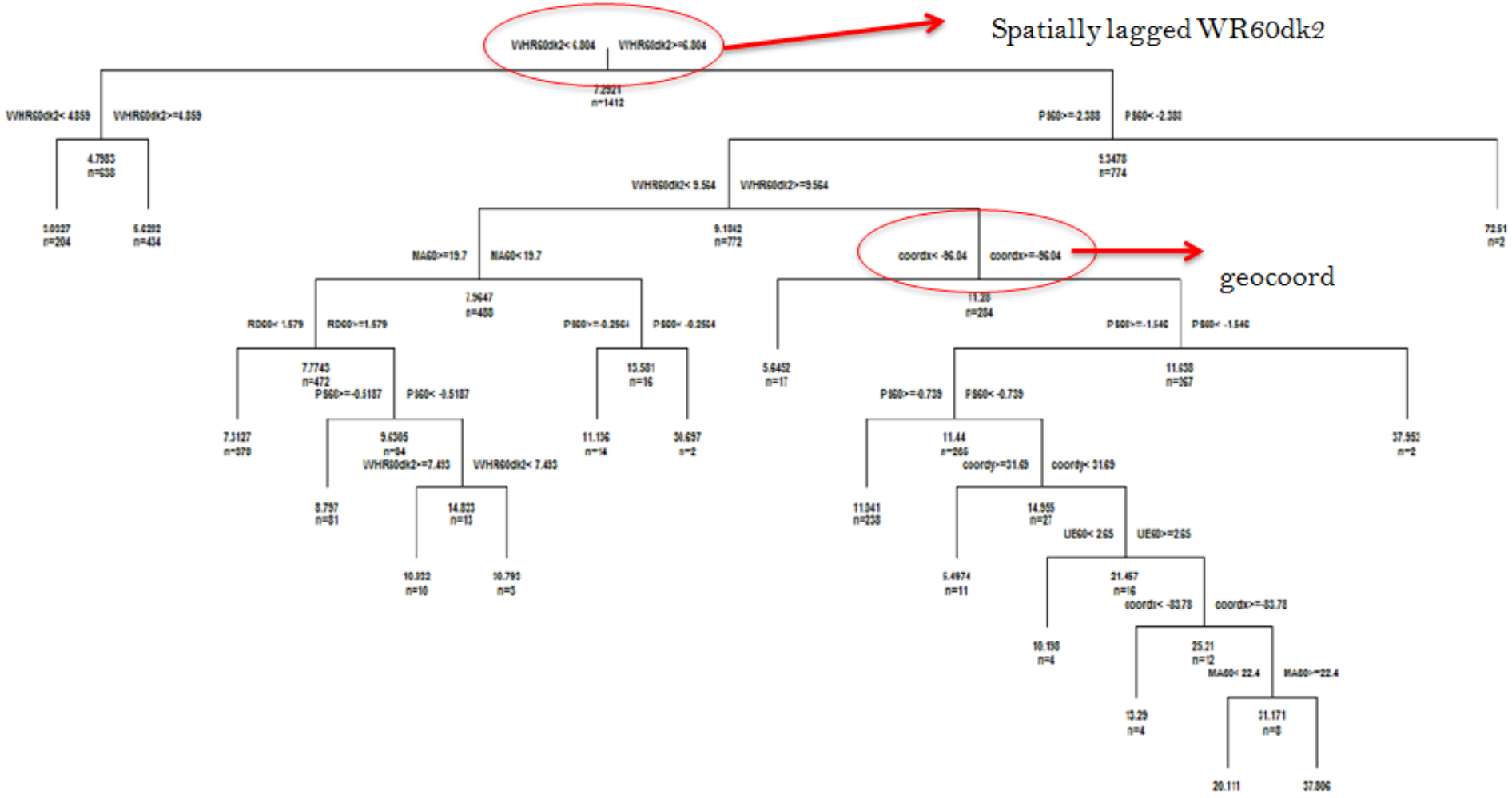


Figure 4.7: The minimun distance-based spatial regression tree with geographical coordinates allows to remove the presence of pseudo-residuals spatial autocorrelation (minimun distance that all regions are linked at least two neighbours)

Table 4.5: Comparison timing and RMSE of spatial lag model by traditional approach

	Spatial weights	Timing		RMSE	
		without geocoords	with geocoords	without geocoords	with geocoords
Ord	rook	22.47	24.36	5.887	5.844
	rook1-2	22.43	24.04	5.749	5.743
	queen	22.4	23.05	5.897	5.855
	dk1	22.41	23.14	5.805	5.793
	dk2	22.58	23.31	5.792	5.793
	dk3	22.57	23.24	5.819	5.820
	dk4	22.71	23.28	5.839	5.839
	dk5	23.09	24.97	5.843	5.843
Matrix	rook	22.93	22.46	5.887	5.844
	rook1-2	22.87	22.63	5.749	5.743
	queen	22.97	22.48	5.897	5.855
	dk1	23.06	22.52	5.805	5.793
	dk2	23.28	22.73	5.792	5.793
	dk3	23.41	22.93	5.819	5.819
	dk4	23.69	23.16	5.839	5.839
	dk5	23.65	23.35	5.483	5.843
LU	rook	25.4	23.09	5.887	5.844
	rook1-2	26.38	23.42	5.749	5.743
	queen	26.27	23.16	5.897	5.855
	dk1	23.94	24.04	5.805	5.793
	dk2	23.28	26.47	5.792	5.796
	dk3	32.54	29.2	5.819	5.820
	dk4	35.35	25.41	5.839	5.839
	dk5	31.25	24.77	5.483	5.843
Chebyshev	rook	22.52	23.12	5.887	5.844
	rook1-2	22.56	23.26	5.749	5.743
	queen	22.37	23.10	5.897	5.855
	dk1	22.59	23.34	5.806	5.793
	dk2	23.25	26.47	5.796	5.795
	dk3	24.15	29.20	5.826	5.825
	dk4	24.43	25.41	5.845	5.843
	dk5	24.46	24.77	5.850	5.847
MC	rook	22.25	22.33	5.887	5.844
	rook1-2	22.27	22.45	5.748	5.742
	queen	22.26	22.45	5.897	5.855
	dk1	22.30	22.43	5.806	5.793
	dk2	22.44	22.65	5.792	5.793
	dk3	22.58	22.78	5.878	5.820
	dk4	22.69	23.37	5.838	5.838
	dk5	22.84	25.13	5.845	5.842
2SLS	rook	0.02	0.11	5.888	5.859
	rook1-2	0.01	0.03	5.717	5.715
	queen	0.01	0.01	5.918	5.873
	dk1	0.01	0.02	5.782	5.783
	dk2	0.03	0.04	5.780	5.779
	dk3	0.05	0.07	5.812	5.809
	dk4	0.06	0.06	5.832	5.832
	dk5	0.04	0.07	5.836	5.837

Notes: Timing is expressed in seconds (elapsed time)

Table 4.6: Comparison timing and RMSE of spatial lag combined with geographical coordinates regression tree

Spatial lag	Timing		RMSE	
	without geocoords	with geocoords	without geocoords	with geocoords
rook	0.05	0.05	5.466	5.034
rook1-2	0.03	0.06	4.772	4.752
queen	0.05	0.06	5.495	5.418
dk1	0.04	0.05	5.009	4.883
dk2	0.05	0.04	5.009	4.679
dk3	0.03	0.05	5.020	4.698
dk4	0.05	0.05	4.721	4.712
dk5	0.04	0.06	5.175	4.834

Notes: Timing is expressed in seconds (elapsed time).

As can we underlined in the previous tables the timing of regression tree based on spatial lag combined with geographical coordinates is approximately egual to timing the spatial lag model based on 2SLS for different specifications of spatial matrix. Also, the RMSE in this version of regression tree is less than in spatial regression obtained by 2SLS procedure.

2. **California Census Block Groups Housing**

Table 4.7: Permutational Moran's I on pseudo-residuals of non-spatial regression tree and spatial regression tree based on geographical coordinates (California Census Block Groups Housing)

Permutational Moran's I		
Weights matrix	Without geocoords	With geocoords
dk1	0.3399	0.1606
	(0.001*)	(0.001*)
dk2	0.3107	0.1409
	(0.001*)	(0.001*)
dk3	0.2876	0.1241
	(0.001*)	(0.001*)
dk4	0.2876	0.1241
	(0.001*)	(0.001*)
dk5	0.2520	0.1100
	(0.001*)	(0.001*)

Notes: number of simulations=999, pseudo-pvalue in brackets, "*" statistically significant at 0.05 level.

As argued in the analysis on the first geodataset, the inclusion of geographical coordinates in non-spatial regression tree leads to a decrement on Permutational Moran's I. Now, we show the significant reduction of the complexity computational and accuracy of spatial lag combined with or without geographical coordinates regression tree with respect to spatial regression based on spatial 2SLS estimation for various specifications of spatial weights matrix.

Table 4.8: Comparison timing and RMSE of spatial lag combined with geographical coordinates regression tree (California Census Block Groups Housing)

Spatial lag	Timing		RMSE	
	without geocoords	with geocoords	without geocoords	with geocoords
dk1	1.456	1.706	0.3363	0.3363
dk2	1.497	1.723	0.3378	0.3378
dk3	1.430	1.674	0.3498	0.3498
dk4	1.430	1.663	0.3498	0.3498
dk5	1.541	1.749	0.3456	0.3456

Notes: Timing is expressed in seconds (elapsed time).

Table 4.9: Comparison timing and RMSE of spatial 2SLS (California Census Block Groups Housing)

Spatial lag	Timing		RMSE	
	without geocoords	with geocoords	without geocoords	with geocoords
dk1	44.888	58.095	0.2999	0.2974
dk2	51.806	67.526	0.3060	0.3022
dk3	58.353	76.253	0.3105	0.3050
dk4	58.347	76.255	0.3105	0.3050
dk5	62.677	81.906	0.3167	0.3089

Notes: Timing is expressed in seconds (elapsed time).

3. **Regular grids**

Table 4.10: Comparison of Permutational Moran's I on pseudo-residuals of Spatial Regression Tree with different orders of spatial weigths matrix

Regular grid	Permutational Moran's I						
	W1	W2	W3	W4	W5	W6	W7
320X320	0.2738	0.1887	0.095	0.0856	0.057	0.0537	0.0368
	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)
400X500	0.2754	0.1840	0.0960	0.0843	0.0576	0.0522	0.0381
	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)
500X1000	0.2727	0.1833	0.0937	0.0831	0.0559	0.0516	0.0393
	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)
800X1000	0.2758	0.1839	0.0940	0.0821	0.0555	0.0513	0.0392
	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)
1024X1024	0.2744	0.1833	0.0936	0.0826	0.0564	0.0521	0.0392
	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)	(0.001*)

Notes: number of simulations=999, pseudo-pvalue in brackets, "*" statistically significant at 0.05 level, number of explanatory variables=20

Table 4.11: Comparison of Percentage nonzero weights (measure of sparsity) of spatial lag with different orders of spatial weigths matrix

Regular grid	Percentage nonzero weights						
	W1	W2	W3	W4	W5	W6	W7
320X320	0.004	0.008	0.012	0.016	0.020	0.023	0.027
400X500	0.002	0.004	0.006	0.008	0.010	0.012	0.014
500X1000	0.001	0.001	0.002	0.003	0.004	0.005	0.006
800X1000	0.001	0.001	0.002	0.002	0.003	0.003	0.004
1024X1024	0.0003	0.0007	0.0011	0.0015	0.0019	0.0023	0.0027

Table 4.12: Comparison of RMSE and timing on regular grid of Spatial Regression Tree for different orders of spatial weigths matrix (number of explanatory variables=20)

Regular grid	RMSE							Timing						
	W1	W2	W3	W4	W5	W6	W7	W1	W2	W3	W4	W5	W6	W7
320X320	0.942	0.959	0.586	0.997	0.105	0.913	0.988	40.820	46.126	45.749	41.509	40.647	41.330	41.049
400X500	0.956	0.955	0.999	0.106	0.132	0.921	0.102	101.619	110.012	98.848	100.446	102.925	102.468	109.305
500X1000	0.957	0.962	1.003	0.982	0.936	0.927	0.107	004.041	420.258	427.128	418.034	418.296	417.320	417.291
800X1000	0.958	0.957	0.999	0.977	0.933	0.984	1.006	941.929	933.940	973.443	971.884	971.024	956.411	997.780
1024X1024	0.964	0.962	0.604	0.981	0.9989	0.999	0.108	1470.528	1576.031	1558.403	1606.713	1570.930	1603.432	1571.290

Table 4.13: Comparison of RMSE and timing on regular grid of Spatial 2SLS estimation for different orders of spatial weigths matrix (number of explanatory variables=20)

Regular grid	RMSE							Timing						
	W1	W2	W3	W4	W5	W6	W7	W1	W2	W3	W4	W5	W6	W7
320X320	1.143	0.997	1.002	1.004	0.999	1.000	1.042	267.699	328.522	371.750	413.292	454.604	500.125	567.573
400X500	1.878	0.990	1.000	0.997	0.998	0.998	1.030	526.765	633.082	714.077	812.650	885.486	1005.836	1096.275
500X1000	0.981	1.005	0.999	0.998	1.000	1.001	1.031	1417.174	1652.443	1889.167	2121.269	2331.244	2578.533	2758.312
800X1000	1.268	0.995	1.000	0.999	1.001	0.999	1.031	2478.502	2982.401	3418.173	3559.660	3781.194	4239.146	4629.470
1024X1024	1.068	0.989	1.000	0.999	0.999	0.999	1.031	3512.772	3925.829	4333.758	4970.177	5444.703	5813.897	6371.675

4.4 Data Mining Conclusions and Final remarks

1. In all experiments, in the presence of pseudo-residuals spatial autocorrelation in a structured tree, we note that the introduction of spatially lagged response variable or geographical coordinates allows to reduce or remove this effect.

2. Spatial Regression Tree is less computationally expensive than spatial 2SLS estimation and other techniques based on ML-approach.

3. The accuracy of Spatial Regression Tree is approximately equivalent to spatial 2SLS estimation.

4.4.1 Data Mining Future works

Possible improvements of the our proposal and ideas of further development are outlined in this subsection. In order to stress their importance, they are briefly summarized:

- Test the procedure in different real datasets or in simulated data.
- Test the procedure by using different spatial weights matrix.
- Test the potential performance using parallel implementation or increasing number of processors.
- Apply the Spatial Tree algorithm on various spatial econometric models (cross-sectional and panel data).
- Extend the approach to different mining techniques: Boosting, Bagging and Random Forests, SVM (Support Vector Machine), DBSCAN (Density-Based Spatial Clustering of Applications with Noise).
- Introduction the spatial measure in the split criterion.

Chapter 5

General conclusions

In this work we provided a broad overview of computational difficulties for decomposition and approximation techniques to solve the problem of computing the Jacobian in spatial models. We focused on the specif spatial econometric model: spatial lag or spatial autoregressive model and by Monte Carlo simulations we compared the accuracy and computational complexity, for various regular lattice structure of several methodologies based on Maximum Likelihood and Spatial Two-stage least squares estimation. Furthermore, we investigated new evidences as the double effect due the sparsity of spatial weights matrix and the size of datasets, in terms of computational complexity. In this context, we proposed the possible extentions of some notions of spatial econometrics in the spatial data mining framework.

SDM provides techniques for discovering unexpected patterns from large geographical databases. Those techniques derive benefits from e.g. database management, spatial statistics and artificial intelligence. Although this discipline brings new possibilities, it also faces many challenging research problems especially related to spatial data characteristics. To obtain relevant results the spatial autocorrelation and spatial heterogeneity have to be taken into consideration. After surveying the best known SDM techniques, this thesis concentrates on CART (Classification and Regression Tree) algorithm in more detail. We observed how the "space" may add significant insights in a regression tree approach. We investigated the possible role of spatial arrangement on the variables in the set of predictors in data mining model. Also, in the context of very large geodatasets, the **integration** of some notions of spatial econometrics and spatial data mining allows to evaluate different aspects:

- the importance of considering *spatial autocorrelation in spatial predictive data mining models*;
- the *reduction of computational complexity* with respect to traditional spatial econometric models;
- the *reduction of spatial autocorrelation on pseudo-residuals.*

In particular, we compared the performance of various versions of Classification and Regression Trees (CART), in terms of pseudo-residuals spatial autocorrelation, accuracy

and computational complexity for real-world and simulated datasets. In the presence of pseudo residuals spatial autocorrelation in a structured tree, the introduction of spatial lag variables and geographical coordinates allows to remove or reduce this effect among pseudo residuals. We also shown in severals real and simulated datasets the significant saving in computational complexity with respect to traditional spatial econometric approach.
By considering, the wide range of data mining methods developed and problems addressed, many directions for further work have opened up during the research presented in this dissertation. In terms of extensions of the developed methods, we should consider also the types of spatial autocorrelation and different split criteria based on a sort of spatial measure. In terms of applications, we would test the procedure in different spatial datasets (real or simulated). By considering the generality of our approach, we would apply the spatial tree algorithm on various spatial econometric models. Finally, we would extend the approach to different mining techniques: Boosting, Bagging and Random Forests, SVM (Support Vector Machine), DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

Bibliography

[1] L. Anselin. *Estimation Methods for Spatial Autoregressive Structures*. Regional Science Dissertation and Monography Series, Cornell University, Ithaca, New York, 1980.

[2] L. Anselin. *Spatial Econometrics: Methods and Models*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1988a.

[3] L. Anselin. Model validation in spatial econometrics: A review and evaluation of alternative approaches. *International Regional Science Review*, 11(3), 1988b.

[4] L. Anselin. Testing for spatial dependence in linear regression models: A review. *Regional Research Institute Research Paper, West Virginia University*, 94-16, 1994.

[5] L. Anselin. Spatial externalities, spatial multipliers and spatial econometrics. *International Regional Science Review*, 26(2):153–166, 2003.

[6] L. Anselin. *Spatial Regression Analysis in R: A Workbook*. Spatial Analysis Laboratory Department of Geography University of Illinois, Urbana-Champaign, 2007.

[7] L. Anselin and A.K. Bera. *Spatial dependence in linear regression models with an introduction to spatial econometrics*. In Handbook of Applied Economic Statistics, ed. by A. Ullah and D. E. A. Giles. New York: Marcel Dekker, 1998.

[8] L. Anselin and R. Florax. Small sample properties fo tests for spatial dependence in regression models: Some further results. *West Virginia University, Regional Reserach Institute*, (9414), 1994.

[9] L. Anselin and R. Florax. *New Directions in Spatial Econometrics*. Springer-Verlag, New York, 1995.

[10] L. Anselin, R. Florax, and S.J. Rey. *Advances in Spatial Econometrics. Methodology, Tools and Applications*. Berlin: Springer-Verlag, 2004.

[11] L. Anselin and D.A. Griffith. Do spatial effects really matter in regression analysis? *Papers, Regional Science Association*, (65):11–34, 1988.

[12] G. Arbia. *Spatial Econometrics: Statistical Foundations and Applications to Regional Convergence*. Berlin: Springer-Verlag, 2006.

[13] B.H. Baltagi. *Econometrics,(third edition)*. Berlin: Springer-Verlag, 1999.

[14] B.H. Baltagi. *Econometric Analysis of Panel Data, (second edition)*. John Wiley and Sons, Chichester, England, 2001.

[15] S. Banerjee et al. *Hierarchical Modeling and Analysis for Spatial Data*. Boca Raton: Chapman and Hall/CRC, 2004.

[16] M.S. Barlett. The spectral analysis of two-dimensional point processes. *Biometrika*, 51:299–311, 1964.

[17] M.S. Barlett. *The Statistical Analysis of Spatial Pattern*. London: Chapman and Hall, 1975.

[18] R. Barry and R. K. Pace. Monte carlo estimates of the log determinant of large sparse matrices. *Linear Algebra Applications*, 289:41–45, 1999.

[19] M.S. Bartlett. The spectral analysis of point processes. *Journal of the Royal Statistical Society*, (25):264–296, 1963.

[20] J. Behnke and E. Dobinson. Nasa workshop on issues in the application of data mining to scientific data. *SIGKDD Explor. Newsl.*, 2(1):70–79, June 2000.

[21] L. Bel, D. Allard, J. M. Laurent, R. Cheddadi, and A. Bar-Hen. Cart algorithm for spatial data: Application to environmental and ecological data. *Computational Statistics and Data Analysis*, 53(8):3082–3093, June 2009.

[22] A.K. Bera and M.J. Yoon. *Simple diagnostic tests for spatial dependence*. Department of Economics, University of Illinois, Champaign, IL, 1992.

[23] E. R. Berndt. *The Practice of Econometrics, Classic and Contemporary*. Addison-Wesley Publishing Company, Reading, 1991.

[24] J. Besag. Spatial interaction and the statistical analysis of lattice system. *Journal of the Royal Statistical Society B*, (36):235–260, 1974.

[25] R. Bivand. Regression modeling with spatial dependence: An application of some class selection and estimation methods. *Geographical Analysis*, 16(1):25–37, 1984.

[26] R. Bivand et al. Computing the jacobian in gaussian spatial autoregressive models: an illustrated comparison of available methods. *Geographical Analysis*, 45(2):150–179, 2013.

[27] G.E.P. Box and G. M. Jenkins. *Time-series Analysis, Forecasting and Control*. Wiley, 1970.

[28] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[29] L. Breiman, G. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. New York: Chapman and Hall, 1984.

[30] P. Burridge. On the cliff-ord test for spatial autocorrelation. *Journal of Royal Statisitcal Society B*, 42:107–108, 1980.

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

[32] A. Cliff and J. K. Ord. *Spatial Autocorrelation*. London:Pion, 1973.

[33] A. Cliff and J. K. Ord. *Spatial Processes:Models*. London:Pion, 1981.

[34] A. Cliff and J.K. Ord. Testing for spatial autocorrelation among regression residuals. *Geographical Analysis*, (4):267–84, 1972.

[35] N. Cressie. *Statistics for Spatial Data*. New York: Wiley, 1993.

[36] Sanjoy Dasgupta and Yoav Freund. Random projection trees and low dimensional manifolds. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 537–546, New York, NY, USA, 2008. ACM.

[37] J. Davidson. *Econometric Theory*. Blackwell Publishers, 2000.

[38] R. Davidson and J.G. Mackinnon. *Estimation and Inference in Econometrics*. Oxford University Press, Oxford, 1993.

[39] Li Deren and Wang Shuliang. Concepts, principles and application of spatial data mining and knowledge discovery. In *ISSTM*, pages 27–29, August 2005 Beijing, China.

[40] M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad, editors, *KDD*, pages 226–231. AAAI Press, 1996.

[41] U. Fayyad. Editorial. *ACM SIGKDD Explorations*, 2(5):1–3, 2003.

[42] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The kdd process for extracting useful knowledge from volumes data. *Comunications of the ACM, Special Issue on Data Mining*, 39(11):27–34, 1996.

[43] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. MIT Press, 1996.

[44] B. Fingleton. *Generalized Method of Moments Estimator for a Spatial Model with Moving Average Errors, with Application to Real Estate Prices*. In Arbia. G., and B.H.Baltagi (eds.), Spatial Econometrics: Methods and Applications. Physica-Verlag/Springer, 2008.

[45] M.J. Fortin and M. Dale. *Spatial Analysis: A Guide for Ecologists.* Cambridge: Cambridge University Press, 2005.

[46] A.S. Goldberg. *Introductory Econometrics.* Harvard edition world, 1998.

[47] C. Gourieroux and A. Montfort. *Statistics and Econometric Models, Vols. 1 and 2.* Cambridge University Press, 1995.

[48] W. H. Greene. *Econometric Analysis (fifth edition).* New York, Macmillan, 2003.

[49] D. A. Griffith. Simplifying the normalizing factor in spatial autoregressions for irregular lattices. *Papers in Regional Science*, 71(1):71–86, 1992.

[50] D. A. Griffith. Eigenfunction properties and approximations of selected incidencce matrices employed in spatial analyses. *Linear Algebra and its implication*, 321:95–112, 2000.

[51] D. A. Griffith. Faster maximum likelihood estimation of very large spatial autoregressive models: An extension of the smirnov and anselin result. *Journal of Statistical Computation and Simulation*, 74:855–866, 2004.

[52] D. A. Griffith. Selected challenges from spatial statistics for spatial econometricians. *Comparative Economic Research*, 15(4):71–85, 2012.

[53] D. A. Griffith and A. Sone. Trade-offs associated with normalizing constant computational simplifications for estimating spatial statistical models. *Journal of Statistical Computation and Simulation*, 51(2-4):165–183, 1995.

[54] D.A. Griffith. *Advanced Spatial Statistics.* Dordrecht: Kluwer Academic Publishers, 1988.

[55] D. Gujarati. *Basic Econometrics, (fourth edition).* McGraw-Hill, 2003.

[56] R. Haining. *Spatial Data Analysis in the Social and Environmental Sciences.* Cambridge: Cambridge University Press, 1990.

[57] R. Haining. *Spatial Data Analysis: Theory and Practice.* Cambridge: Cambridge University Press, 2003.

[58] Jiawei Han J., Micheline Kamber, and Anthony K. H. Tung. Spatial clustering methods in data mining: A survey. In Harvey J. Miller and Jiawei Han, editors, *Geographic Data Mining and Knowledge Discovery, Research Monographs in GIS.* Taylor and Francis, 2001.

[59] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition.* Springer Series in Statistics. Springer, 2009.

[60] N.J. Higham. *Accuracy and stability of numerical algorithms*. 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2002.

[61] L. Hordijk. Problems in estimating econometric relations in space. *Papers of the Regional Science Association*, 42(1):99–115, 1979.

[62] J.S. Huang. The autoregressive moving average model for spatial analysis. *Australian Journal of Statistics*, 26(2):169–78, 1984.

[63] Yan Huang, Shashi Shekhar, and Hui Xiong. Discovering colocation patterns from spatial data sets: A general approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(12):1472–1485, 2004.

[64] J. Johnston. *Econometric Methods*. McGraw Hill, New York, 1991.

[65] Baris M. Kazar, Shashi Shekhar, David J. Lilja, Ranga R. Vatsavai, and R. Kelley Pace. Comparing exact and approximate spatial auto-regression model solutions for spatial data analysis. In MaxJ. Egenhofer, Christian Freksa, and HarveyJ. Miller, editors, *Geographic Information Science*, volume 3234 of *Lecture Notes in Computer Science*, pages 140–161. Springer Berlin Heidelberg, 2004.

[66] H. Kelejian and I. Prucha. A generalized moments estimator for the autoregressive parameter in a spatial model. *International Economic Review*, 40(2):509–533, 1999.

[67] H.H. Kelejian and P.D.Robinson. Spatial autocorrelation: A new computationally simple test with an application to per capita county policy expenditures. *Regional Science and Urban Economics*, 22:317–31, 1992.

[68] P. Kennedy. *A Guide to Econometrics, (fifth edition)*. Blackwell Publishers, 2003.

[69] J. Kmenta. *Elements of Econometrics, (second edition)*. Mcmillan, New York, 1997.

[70] Krzysztof Koperski and Jiawei Han. Discovery of spatial association rules in geographic information databases. In MaxJ. Egenhofer and JohnR. Herring, editors, *Advances in Spatial Databases*, volume 951 of *Lecture Notes in Computer Science*, pages 47–66. Springer Berlin Heidelberg, 1995.

[71] J.P. LeSage and R. K. Pace. *Introduction to Spatial Econometrics*. Boca Raton:CRC Press /Taylor Francis Group, 2009.

[72] Y. Leung. *Knowledge Discovery in Spatial Data*. Advances in Spatial Science, The Regional Science Series. Springer-Verlag Berlin Heidelberg, 2010.

[73] Xiang Li and Christophe Claramunt. A spatial entropy-based decision tree for classification of geographical information. *Transactions in GIS*, 10(3):451–467, 2006.

[74] T. Liu et al. An investigation of practical approximate nearest neighbor algorithms. pages 825–832. MIT Press, 2004.

[75] G. Maddala. *Econometrics*. McGraw-Hill, New York, 2001.

[76] D. Malerba. Mining spatial data: Opportunities and challenges of a relational approach. In *IASC 07*, August 30th- September 1st, 2007, Aveiro, Portugal.

[77] D. Malerba and M. Ceci. Mining model trees from spatial data. in. In *Gama (Eds.), European Conference on Principles and Practice of Knowledge Discovery in Databases, LNAI 3721*. Springer, 2005.

[78] S. Mitra. Kddclus, a simple method for multi-density clustering. In *Proceedings of the International Workshop on Soft Computing Applications and Knowledge Discovery (SCAKD 2011)*, volume 758, 2011.

[79] S. Mitra, S. K. Pal, and P. Mitra. Data mining in soft computing framework: A survey. *IEEE Transactions on Neural Networks*, 1(13), January, 2002.

[80] Yasuhiko Morimoto. Mining frequent neighboring class sets in spatial databases. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 353–358. ACM Press, 2001.

[81] J. Ord. Estimation methods for models of spatial interaction. *Journal of the American Statistical Association*, 70(349):120–126, 1975.

[82] R. Pace and J. LeSage. Chebyshev approximation of log-determinants of spatial weight matrices. *Computational Statistics and Data Analysis*, 45:179–196, 2004.

[83] R.K. Pace and R. Barry. Performing large spatial regressions and autoregressions. *Economics Letters*, 54(3):283–291, 1997.

[84] R.K. Pace and R. Barry. Sparse spatial autoregressions. *Statistics and Probability Letters*, 33(3):291–297, 1997a.

[85] R.K. Pace and R. Barry. Quick computation of spatial autoregressive estimators. *Geographical Analysis*, 29(3):232–246, 1997b.

[86] R.K. Pace et al. Spatial statistics and real estate. *Journal of Real Estate Finance and Economics*, 17:5–13, 1998.

[87] J. Paelinck and L. Klassen. *Spatial Econometrics*. Farnbourough: Saxon House, 1979.

[88] P. Postiglione, R. Benedetti, and G. Lafratta. A regression tree algorithm for the identification of convergence clubs. *Computational Statistics and Data Analysis*, 54(11):2776–2785, November 2010.

[89] S. Rinzivillo and F. Turini. Knowledge discovery from spatial transactions. *J. Intell. Inf. Syst.*, 28(1):1–22, February 2007.

[90] B.D. Ripley. *Spatial Statistics*. New York: Wiley, 1981.

[91] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Manchester University Prees, Manchester, 1992.

[92] O. Schabengerger and C.A. Gotway. *Statistical Methods for Spatial Data Analysis*. Boca Raton: Chapman and Hall/CRC, 2005.

[93] Shashi Shekhar and Yan Huang. Discovering spatial co-location patterns: A summary of results. In *Lecture Notes in Computer Science*, pages 236–256, 2001.

[94] E. Simoudis. Reality check for data mining. *IEEE Expert*, 5(11):26–33, October, 1996.

[95] O. Smirnov. Computation of the information matrix for models with spatial interaction on a lattice. *Journal of Computational and Graphical Statistics*, 14(4):910–927, 2005.

[96] O. Smirnov and L. Anselin. Fast maximum likelihhod estimation of very large spatial autoregresive models: a charactheristic polynomial approach. *Computational Statistics and Data Analysis*, 35(3):301–319, 2001.

[97] O. A. Smirnov and L. Anselin. An o(n) parallel method of computing the log-jacobian of the variable transformation for models with spatial interaction on a lattice. *Computational Statistics and Data Analysis*, 53(8):2980–2988, June 2009.

[98] P. Smyth. Data mining: data analysis on a grand scale? *Statistical Methods in Medical Research*, 4(9):309–327, August, 2000.

[99] J. H. Stock and M. W. Watson. *Introduction to Econometrics*. Addison Wesley, 2003.

[100] D. Stojanova, M. Ceci, A. Appice, D. Malerba, and S. Džeroski. Global and local spatial autocorrelation in predictive clustering trees. In *Proceedings of the 14th International Conference on Discovery Science*, DS'11, pages 307–322, Berlin, Heidelberg, 2011. Springer-Verlag.

[101] D. Stojanova, M. Ceci, A. Appice, D. Malerba, and S. Džeroski. Dealing with spatial autocorrelation when learning predictive clustering trees. *Ecological Informatics*, 13:22–39, 2013.

[102] B. Therneau, T. M. Atkinson and B. Ripley. Rpart: Recursive partitioning, 2012. R package version 4.01-1.

[103] L. Thomas. *Modern Econometrics: An Introduction*. Pearson Education, 1997.

[104] W. Tobler. A computer movie simulating urban growth in the detroit region. *Economic Geography Supplement*, (46):234–40, 1970.

[105] G.J. Upton and B. Fingleton. *Spatial Data Analysis by Example. Volume 1: point pattern and Quantitative Data*. New York: Wiley, 1985.

[106] M. Verbeek. *A Guide to Modern Econometrics*. John Wiley and Sons, New York, 2000.

[107] Nakul Verma, Samory Kpotufe, and Sanjoy Dasgupta. Which spatial partition trees are adaptive to intrinsic dimension? In Jeff Bilmes and Andrew Y. Ng, editors, *UAI*, pages 565–574. AUAI Press, 2009.

[108] J. Walde et al. Performance contest between mle and gmm for huge spatial autogressive models. *Journal of Statistical Computation and Simulation*, 78:151–166, 2008.

[109] L.A. Waller and C.A. Gotway. *Applied Spatial Statistics for Public Health Data*. Hoboken, NJ: John Wiley, 2004.

[110] G.M. Weiss and B.D. Davison. *Data Mining. Handbook of Technology Management*. H. Bidgoli (Ed.), John Wiley and Sons, 2010.

[111] P. Whittle. On stationarity processes in the plane. *Biometrika*, (41):434–449, 1954.

[112] J.M. Woolridge. *Econometrics: A Modern Approach (second edition)*. The MIT Press, Cambridge, Massachusetts, 2002a.

[113] J.M. Woolridge. *Econometric Analysis of Cross-section and Panel Data*. The MIT Press, Cambridge, Massachusetts, 2002b.

[114] Xiangye Xiao, Xing Xie, Qiong Luo, and Wei ying Ma. Density based co-location pattern discovery, 2008.

[115] Donghui Yan, Ling Huang, and Michael I. Jordan. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 907–916, New York, NY, USA, 2009. ACM.

[116] K. Zeitouni. A survey on spatial data mining methods databases and statistics. In *Point of Views, Information Resources Management Association International Conference (IRMA, 2000), Data Warehousing and Mining Track*, 2000.

[117] Xin Zhang, Nikos Mamoulis, David W. Cheung, and Yutao Shou. Fast mining of spatial collocations. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 384–393, New York, NY, USA, 2004. ACM.

[118] Minyue Zhao and Xiang Li. An application of spatial decision tree for classification of air pollution index. In *Geoinformatics, 2011 19th International Conference on*, pages 1–6, June 2011.