

12-15-2018

Estimation of the Parameters in a Spatial Regressive-Autoregressive Model Using Ord's Eigenvalue Method

Sajib Mahmud Mahmud Tonmoy
tonmoysajib1989@gmail.com

Follow this and additional works at: <https://digitalscholarship.unlv.edu/thesesdissertations>



Part of the [Applied Mathematics Commons](#), and the [Statistics and Probability Commons](#)

Repository Citation

Tonmoy, Sajib Mahmud Mahmud, "Estimation of the Parameters in a Spatial Regressive-Autoregressive Model Using Ord's Eigenvalue Method" (2018). *UNLV Theses, Dissertations, Professional Papers, and Capstones*. 3459.

<https://digitalscholarship.unlv.edu/thesesdissertations/3459>

This Thesis is protected by copyright and/or related rights. It has been brought to you by Digital Scholarship@UNLV with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Thesis has been accepted for inclusion in UNLV Theses, Dissertations, Professional Papers, and Capstones by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.

ESTIMATION OF THE PARAMETERS IN A SPATIAL
REGRESSIVE-AUTOREGRESSIVE MODEL
USING ORD'S EIGENVALUE METHOD

By

Sajib Mahmud Tonmoy

Bachelor of Science – Statistics, Biostatistics & Informatics
University of Dhaka, Bangladesh
2014

Master of Science – Statistics, Biostatistics & Informatics
University of Dhaka, Bangladesh
2016

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science - Mathematical Sciences

Department of Mathematical Sciences
College of Sciences
The Graduate College

University of Nevada, Las Vegas
December 2018

©Copyright by Sajib Mahmud Tonmoy 2018

All Rights Reserved



Thesis Approval

The Graduate College
The University of Nevada, Las Vegas

November 15, 2018

This thesis prepared by

Sajib Mahmud Tonmoy

entitled

Estimation of the Parameters in a Spatial Regressive-Autoregressive Model Using Ord's Eigenvalue Method

is approved in partial fulfillment of the requirements for the degree of

Master of Science - Mathematical Sciences
Department of Mathematical Sciences

Petros Hadjicostas, Ph.D.
Examination Committee Chair

Kathryn Hausbeck Korgan, Ph.D.
Graduate College Interim Dean

Hokwon A. Cho, Ph.D.
Examination Committee Member

Kaushik Ghosh, Ph.D.
Examination Committee Member

Ashok K. Singh, Ph.D.
Graduate College Faculty Representative

ABSTRACT

Estimation of the Parameters in a Spatial Regressive-Autoregressive Model

Using Ord's Eigenvalue Method

by

Sajib Mahmud Tonmoy

Examination Committee Chair: Dr. Petros Hadjicostas

In this thesis, we study one of Ord's (1975) global spatial regression models. Ord considered spatial regressive-autoregressive models to describe the interaction between location and a response variable in the presence of several covariates. He also developed a practical estimation method for the parameters of this regression model using the eigenvalues of a weight matrix that captures the contiguity of locations. We review the theoretical aspects of his estimation method and implement it in the statistical package **R**.

We also implement Ord's methods on the Columbus, Ohio, crime data set from the year 1980, which involves the crime rate of each neighborhood of the city as a response variable and the average income and average house value of each neighborhood as covariates. We use different weight matrices that capture different "nearest neighbor" notions and compare the results.

ACKNOWLEDGEMENTS

I consider myself fortunate to have had the opportunity to pursue a thesis work toward my M.S. degree in Mathematical Sciences with concentration in Applied Statistics. This thesis work becomes a reality with the unwavering support of many individuals. I would like to acknowledge and express my deepest appreciation to all of them.

Firstly, I would like to express my special gratitude to my supervisor, Dr. Petros Hadjicostas. I have learned a great deal from him. I am particularly grateful for his kind support and guidance throughout my thesis work.

Also, I would like to extend my sincere thanks to my committee members, Dr. Hokwon Cho, Dr. Kaushik Ghosh, and Dr. Ashok Singh for their comments, suggestions, and continued assistance.

Finally, I would like to thank my family (especially, my wife Rubaiya Murshed) for their love and encouragement throughout my graduate studies.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
1 INTRODUCTION	1
2 ORD'S MODELS OF SPATIAL INTERACTION	7
2.1 Joint probability and conditional autoregressive models	7
2.2 Matrix formulation of autoregressive models	10
2.2.1 Matrix formulation of the joint probability model	11
2.2.2 Properties of the conditional model	12
2.3 The choice of weights in the weight matrix	15
2.3.1 Equally spaced locations	15
2.3.2 Weights for an AR(1) time series model	18
2.3.3 Weights for an irregular lattice	18

2.3.4	Standardization of the weight matrix W	21
2.4	Estimation using maximum likelihood	24
2.4.1	General estimation procedure for the parameters	25
2.4.2	Estimation of the parameters in the conditional model	29
2.4.3	Ord's procedure for solving the MLE equations	31
2.4.4	An implementation of Ord's method in R	33
2.4.5	Use of the R package spdep	34
2.4.6	Some hypothetical examples	35
2.5	A mixed regressive-autoregressive spatial model	41
2.6	Appendix to Chapter 2	43
3	IMPLEMENTATION OF ORD'S METHOD TO THE COLUM-	
	BUS, OHIO, DATA SET	47
3.1	The Columbus, Ohio, spatial data set	47
3.2	An R program for estimating Ord's model	49
3.3	Spatial regression using natural neighborhood contiguity	51
3.4	Spatial regression using nearest neighbours	55
3.5	Results and Discussion	57
3.6	Appendix to Chapter 3	59
4	CONCLUSION AND FUTURE RESEARCH	63
	BIBLIOGRAPHY	65
	CURRICULUM VITAE	69

LIST OF TABLES

3.2	“Nearby” neighborhoods of the first five neighborhoods for different $\widetilde{\mathbf{W}}$'s.	57
3.4	Estimates of the parameters of model (3.1) for different $\widetilde{\mathbf{W}}$'s.	57
3.6	Standard errors of the coefficients of model (3.1) for different $\widetilde{\mathbf{W}}$'s. . .	58

LIST OF FIGURES

2.1	Different connections on a rectangular grid: connected by rook's moves (A), queen's move (B), and bishop's moves (C) (see Sokal and Oden [14, pp. 201 and 203])	16
2.2	An example of an irregular lattice with five locations	19
2.3	Plot of eq. (2.31) from Theorem 2.2, as a function of ρ , for Example 1.	38
2.4	Plot of eq. (2.31) from Theorem 2.2, as a function of ρ , in Example 2.	40

CHAPTER 1

INTRODUCTION

Most regression models used in practice are typically “global”. That is, a response variable is regressed against a set of explanatory variables and the parameters of the fitted model are estimated using *all* the data simultaneously.

In a local regression, a regression model is fitted for each point separately and the parameters of the model are estimated using the “nearby” points. It can be thought of as a generalization of the moving average technique involving both a response and a set of explanatory variables. Thus, in local regression methodology, the number of regressions run is about equal to the number of data points. A *geographically weighted regression* (see O’Sullivan and Unwin [10, Section 8.5]) is the most common example of a local regression.

In this thesis, we study global regression models involving *spatial data*. The phrase “spatial data” (or “geospatial data”) may refer to data that describe the absolute or relative locations (e.g., longitude and latitude) of a geographical phenomenon, or the qualitative and quantitative characteristics (attributes and variables) of spatial features such as points, lines, polygons, polypolygons, surfaces, grids, and so on. For

example, a house can be considered as a point, a road can be considered as a line, a lake can be considered as a polygon, an archipelago of islands can be considered as a polypolygon, the topography of a mountain can be considered as a surface, a photograph can be considered as a grid, and so on.

Spatially referenced data can be represented as vector data or as raster data (see Hijmans [6, Tutorial, Section 2] and O’Sullivan and Unwin [10, section 7.2]).

Vector data refer to spatial objects (countries, provinces, rivers, roads, houses etc.) and they include information about their boundaries, information about their sizes, and information about some of their “attributes”. Vector data can contain points, lines, polygons, etc. (see the discussion above).

Raster data are commonly used to represent continuous phenomena. Continuous spatial data are sometimes also known as *fields* (see Hijmans [6, Tutorial, Section 2]). In a raster data set, we divide an area into a matrix of rectangular shaped cells, where each cell may include aerial and satellite imagery. Unlike vector data, in raster data we use the spatial extents (“bounding box”) of an area and the numbers of rows and columns in which an area is subdivided to determine the size of the raster cells (see again Hijmans [6, Tutorial, Section 2]). An extension of raster data is used for continuous surfaces, that can be represented by *triangulated irregular networks*, which consist of triangular facets (rather than rectangular cells). For example, data sets that can be represented by raster data are data on temperature and elevation.

One very important aspect of spatial data is the *coordinate reference system*. For geographical data, the natural coordinate reference system is an angular system which is called the *longitude and latitude* system (which is a coordinate pair in degrees).

Map projections are never absolutely accurate representations of the spherical Earth. Thus, according to Hijmans [6, Tutorial, Section 2], a “basic way to record a location is to provide a coordinate pair in degrees” and a (so-called) reference datum. Since we cannot measure these angles accurately, we use a “datum” (a model of the shape of the Earth) to estimate those angles. The most commonly used “datum” is *WGS84*, which stands for “World Geodesic System, 1984”. Projections are used to transform this three dimensional angular system to a two dimensional planar system.

In this thesis, we study one of Ord’s [9] global regression models for spatial data, the *spatial regressive-autoregressive linear model*. This model can be used to describe the interaction between location and a response variable in the presence of several covariates (predictors). Ord [9] also developed a practical estimation method of the parameters of this regression model. In this thesis, we review his estimation method and implement it using the statistical package **R**.

The two general global spatial regression models Ord [9] considered are the following:

$$(a) \quad \mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{Y} + \boldsymbol{\varepsilon}, \quad \text{where } \boldsymbol{\varepsilon} \sim \text{MVN}(\mathbf{0}, \sigma^2\mathbf{I}_n);$$

$$(b) \quad \mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{U} \quad \text{and} \quad \mathbf{U} = \rho\mathbf{W}\mathbf{U} + \boldsymbol{\varepsilon}, \quad \text{where } \boldsymbol{\varepsilon} \sim \text{MVN}(\mathbf{0}, \sigma^2\mathbf{I}_n).$$

Here, \mathbf{X} is an $n \times p$ design matrix, $\boldsymbol{\beta}$ is a $p \times 1$ vector of (unknown) coefficients, \mathbf{W} is an $n \times n$ weight matrix, $\mathbf{Y} = (y_1, \dots, y_n)^\top$ is an $n \times 1$ vector of responses, $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)^\top$ is an $n \times 1$ vector of iid random disturbance terms with unknown variance σ^2 , $\mathbf{U} = (u_1, \dots, u_n)^\top$ is a $n \times 1$ vector of disturbance terms, and ρ is an unknown parameter in the interval $(-1, 1)$. In this thesis, we only study model (a).

Model (a) is called a *regressive-autoregressive spatial model*. It is called *regressive* because of the term $\mathbf{X}\boldsymbol{\beta}$ and *autoregressive* because the vector of responses \mathbf{Y} is regressed against itself via the term $\mathbf{W}\mathbf{Y}$, which is a spatial lag term. Model (b) is called a *Simultaneous Autoregressive (SAR) error model*. Here, the error vector \mathbf{U} satisfies a first order autoregressive equation ($\mathbf{U} = \rho\mathbf{W}\mathbf{U} + \boldsymbol{\varepsilon}$). This model can be re-expressed as follows:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{Y} - \rho\mathbf{W}\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

In some way, model (b) seems to be a variation of model (a).

In Chapter 2 of this thesis, following Ord [9], we first discuss the regressive-autoregressive model (a) above for the special case $\mathbf{X} = \mathbf{1}_n = (1, 1, \dots, 1)^\top$ and $\boldsymbol{\beta} = \alpha\mathbf{1}_n^\top$; i.e., when there is *only* an intercept $\beta_0 = \alpha$ (we use Ord's notation). For this case, we first discuss the properties of the joint probability and conditional models. The conditional model is a variation of the joint probability model, which was developed in earlier decades by Bartlett [2] and Besag [3, 4].

For the special case $\boldsymbol{\beta} = \alpha\mathbf{1}_n^\top$, we then develop Ord's eigenvalue method to estimate the parameters of model (a) above. We use Jacobi's formula for the derivative of the determinant of a square matrix (see equation (2.23) later in the thesis) to find an equation that is satisfied by the MLE of the parameter ρ (something that Ord [9] did not do in his paper). We also develop an **R** code that implements Ord's eigenvalue method and use some hypothetical examples to validate our work.

In Chapter 2, we also explain how the weight matrix \mathbf{W} can be chosen in practice (depending on what kind of spatial data we have) and why it has to be row-

standardized in order for Ord’s eigenvalue method to provide reliable numerical estimates of the MLE of the parameter ρ .

The general results for the estimation of the parameters in the general model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{Y} + \boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon} \sim \text{MVN}(\mathbf{0}, \sigma^2\mathbf{I}_n)$, are given in Section 2.5 of the thesis. They involve the well-known hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ of classical linear regression. Some of the equations of that section appear in Ord [9], but the equation satisfied by the MLE of the parameter ρ (see equation (2.35) in this thesis) does not seem to be in that paper. It is clear, however, that this equation is known to later authors; see Anselin [1, Chapter 12].

In Chapter 3 of the thesis, we implement Ord’s eigenvalue method (using our own **R** code) to a real data set, the Columbus, Ohio, crime data, from 1980. This data set originally appeared in Anselin [1, Chapter 12]. We use different weight matrices \mathbf{W} (based on natural contiguity and the m^{th} nearest neighbour of each location) to estimate the parameters of the model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{Y} + \boldsymbol{\varepsilon}$. (Each such weight matrix \mathbf{W} has to be row-standardized first, otherwise, the numerical results of the estimation of ρ are not reliable.)

We use the built-in **R** function `lagsarlm` from the **R** package `spdep` to check our output from the **R** code we developed. The results (estimates of the parameters obtained using the MLE method) agree to 4 decimal digits.

Due to time constraints, we did not develop the theory for the calculation of the asymptotic standard errors of the MLE’s of the parameters. Thus, for the Columbus crime data, we use the asymptotic standard errors obtained using the **R** function `lagsarlm` when we compare different models with different weight matrices \mathbf{W} ; see

Table 3.6 of this thesis. It seems that Anselin [1, Chapters 10 and 12] actually lists the equations satisfied by the standard errors of the estimates of the parameters, but we did not examine them in detail.

In Chapter 4, we conclude the thesis by discussing some possible future research topics, such as the “Simultaneous Autoregressive (SAR)” error model, which is a generalization of the model with equations $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{U}$ and $\mathbf{U} = \rho\mathbf{W}\mathbf{U} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim \text{MVN}(\mathbf{0}, \sigma^2\mathbf{I}_n)$. As mentioned before, we do not examine this model in detail in this thesis (due to time constraints).

CHAPTER 2

ORD'S MODELS OF SPATIAL INTERACTION

Ord [9] considered autoregressive models to describe the interaction between location and a response variable. He also developed practical estimation methods of the parameters. In this chapter, we examine his regressive-autoregressive models for spatial regression.

2.1 Joint probability and conditional autoregressive models

Suppose Y is a random (response) variable at location i . To describe the spatial interaction between locations for the response variable Y , Ord [9] first considered the following autoregressive model:

$$Y_i = \alpha + \rho \sum_{j \in J(i)} w_{ij} Y_j + \varepsilon_i \quad (i = 1, \dots, n). \quad (2.1)$$

Here, ρ and α are parameters, and the ε_i 's ($i = 1, \dots, n$) are random disturbance terms that are uncorrelated, have equal variances, and zero means. Also, $\{w_{ij}\}$ is a

set of non-negative weights (by convention, we use $w_{ii} = 0$). Here, w_{ij} represents the “degree of interaction” of location i with the set of “neighboring” locations

$$J(i) = \{i_1, i_2, \dots, i_{m_i}\}.$$

We implicitly assume that $w_{ij} = 0$ when $j \notin J(i)$. The set $J(i)$ may include all locations except location i . Note that

$$\bigcup_{i=1}^n J(i) \subseteq \{1, 2, \dots, n\}.$$

The usual assumption about $\varepsilon_1, \dots, \varepsilon_n$ is that they have a $\text{MVN}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$ distribution, i.e., they are independently and identically distributed with a $N(0, \sigma^2)$ distribution.

Ord [9] mentions that the joint probability model (2.1) follows the general formulation of models developed by Whittle [16, 17]. He also considered the following conditional model, developed by Bartlett [2] and Besag [3, 4]:

$$E[Y_i | Y_j = y_j, j \in J(i)] = \alpha + \rho \sum_{k \in J(i)} w_{ik} y_k \quad (i = 1, \dots, n). \quad (2.2)$$

The following lemma shows the relationship between models (2.1) and (2.2). It essentially clarifies a statement by Ord [9, p. 120] about the relation between models (2.1) and (2.2).

Lemma 2.1. *1) If equations (2.1) and (2.2) hold, then*

$$E[\varepsilon_i | Y_j = y_j, j \in J(i)] = 0 \quad (i = 1, \dots, n). \quad (2.3)$$

2) If equations (2.1) and (2.3) hold, then equation (2.2) holds as well.

Proof. 1) Assume equations (2.1) and (2.2) hold. Then, taking conditional expectation on both sides of equation (2.1), given $\{Y_j = y_j, j \in J(i)\}$, we get:

$$\begin{aligned} E[Y_i | Y_j = y_j, j \in J(i)] &= E[\alpha + \rho \sum_{k \in J(i)} w_{ik} Y_k + \epsilon_i | Y_j = y_j, j \in J(i)] \\ &= \alpha + \rho \sum_{k \in J(i)} w_{ik} E[Y_k | Y_j = y_j, j \in J(i)] \\ &\quad + E[\epsilon_i | Y_j = y_j, j \in J(i)]. \end{aligned}$$

Since $E[Y_k | Y_k = y_k] = y_k$ for $k \in J(i)$, we can write the above equation as:

$$E[Y_i | Y_j = y_j, j \in J(i)] = \alpha + \rho \sum_{k \in J(i)} w_{ik} y_k + E[\epsilon_i | Y_j = y_j, j \in J(i)] \quad (i = 1, \dots, n). \quad (2.4)$$

Equating equations (2.2) and (2.4), we get equation (2.3), which proves the first part of the lemma.

2) Assume equations (2.1) and (2.3) hold. We have seen above that (2.1) implies (2.4). Substituting equation (2.3) into equation (2.4), we get:

$$E[Y_i | Y_j = y_j, j \in J(i)] = \alpha + \rho \sum_{k \in J(i)} w_{ik} y_k \quad (i = 1, \dots, n),$$

which is equation (2.2), and this proves the second part of the lemma. \square

Equation (2.3) shows the relationship between ϵ_i and $Y_j, j \in J(i)$, when both models (2.1) and (2.2) hold. It can be proved that, if equation (2.3) holds, then ϵ_i and Y_j , where $j \in J(i)$, are uncorrelated. We can explain this restriction using a time series model example (see Ord [9, p. 120]).

Example (AR(1) time series model): Suppose the set of non-negative weights

$\{w_{ij}\}$ is as follows:

$$w_{ij} = \begin{cases} 1, & \text{if } j = i - 1 \ (i \geq 2), \\ 0, & \text{elsewhere.} \end{cases} \quad (2.5)$$

In this case, in equation (2.1), if $i \geq 2$, we have $J(i) = \{i - 1\}$ (only one neighbour to location i), while $J(1) = \emptyset$. In addition, for $i \geq 2$,

$$\sum_{j \in J(i)} w_{ij} Y_j = \sum_{j \in \{i-1\}} w_{ij} Y_j = w_{i,i-1} Y_{i-1} = Y_{i-1}.$$

Also, for $i \geq 2$, ϵ_i is uncorrelated with $Y_{i-1}, Y_{i-2}, \dots, Y_1$. Thus, equation (2.1) reduces to:

$$Y_i = \alpha + \rho Y_{i-1} + \epsilon_i \quad (i = 2, \dots, n). \quad (2.6)$$

Equation (2.3) becomes a natural restriction because of the one-sided and time-dependent nature of equation (2.6). Thus, for this time series model, equation (2.3) becomes: $E[\epsilon_i | Y_{i-1} = y_{i-1}] = 0 \quad (i = 2, \dots, n)$. \square

Ord [9] says that equation (2.3) does not always hold, and as a result the Ordinary Least Squares (OLS) estimators of the parameters α , ρ , and σ^2 may not be consistent. That is, when equation (2.3) does not hold and n is large, $\hat{\alpha}_{OLS}$, $\hat{\rho}_{OLS}$, and $\hat{\sigma}_{OLS}^2$ may not necessarily converge in probability to the true values of the parameters α , ρ , and σ^2 , respectively. This the reason why he suggests to use the ML method to estimate the unknown parameters.

2.2 Matrix formulation of autoregressive models

In Section 2.2.1, we give Ord's matrix formulation of the joint probability model (2.1), while in Section 2.2.2, we give some matrix properties of the conditional model (2.2).

2.2.1 Matrix formulation of the joint probability model

Following Ord [9], model (2.1) can be written in matrix form as follows:

$$\mathbf{Y} = \boldsymbol{\alpha} + \rho \mathbf{W} \mathbf{Y} + \boldsymbol{\varepsilon}. \quad (2.7)$$

Here, $\boldsymbol{\alpha} = (\alpha, \alpha, \dots, \alpha)^\top$ is an $n \times 1$ vector of intercepts, \mathbf{W} is an $n \times n$ weight matrix whose (i, j) th element is w_{ij} , and $\mathbf{Y} = (y_1, y_2, \dots, y_n)^\top$ and $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)^\top$ are $n \times 1$ vectors.

From equation (2.7), if $E[\mathbf{Y}] = \boldsymbol{\mu}$, then taking expectation on both sides of (2.7), we get:

$$\begin{aligned} E[\mathbf{Y}] &= E[\boldsymbol{\alpha} + \rho \mathbf{W} \mathbf{Y} + \boldsymbol{\varepsilon}] \implies E[\mathbf{Y}] = \boldsymbol{\alpha} + \rho \mathbf{W} E[\mathbf{Y}] + \mathbf{0} \\ &\implies \boldsymbol{\mu} = \boldsymbol{\alpha} + \rho \mathbf{W} \boldsymbol{\mu} \\ &\implies \boldsymbol{\alpha} = (\mathbf{I} - \rho \mathbf{W}) \boldsymbol{\mu}. \end{aligned}$$

If $\mathbf{A} = \mathbf{I} - \rho \mathbf{W}$, then we can write $\boldsymbol{\alpha} = \mathbf{A} \boldsymbol{\mu}$. Since $\boldsymbol{\alpha} = (\alpha, \alpha, \dots, \alpha)^\top$, i.e., all components of $\boldsymbol{\alpha}$ are the same, the equation $\boldsymbol{\alpha} = \mathbf{A} \boldsymbol{\mu}$ puts a restriction on the mean vector $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_n)^\top$.

Again, from equation (2.7), after plugging in the value of $\boldsymbol{\alpha}$, we get the following expression for $\boldsymbol{\varepsilon}$:

$$\begin{aligned} \mathbf{Y} &= (\mathbf{I} - \rho \mathbf{W}) \boldsymbol{\mu} + \rho \mathbf{W} \mathbf{Y} + \boldsymbol{\varepsilon} \implies \boldsymbol{\varepsilon} = \mathbf{Y} - (\mathbf{I} - \rho \mathbf{W}) \boldsymbol{\mu} - \rho \mathbf{W} \mathbf{Y} \\ &\implies \boldsymbol{\varepsilon} = (\mathbf{I} - \rho \mathbf{W})(\mathbf{Y} - \boldsymbol{\mu}). \end{aligned}$$

Since $\mathbf{A} = \mathbf{I} - \rho \mathbf{W}$, we can write $\boldsymbol{\varepsilon} = \mathbf{A}(\mathbf{Y} - \boldsymbol{\mu})$.

2.2.2 Properties of the conditional model

When the conditional autoregressive model holds (see equations (2.2) and (2.3)), we can prove some properties, which allow us later to find the MLE's of the parameters (see Lemma 2.5 in Section 2.4). The following result clarifies some statements made by Ord [9, pp. 121–122] about the conditional autoregressive model.

Lemma 2.2. *If equations (2.1) and (2.3) hold, then:*

(a) $E(\boldsymbol{\varepsilon}^\top \mathbf{W}\mathbf{Y}) = 0;$

(b) $\text{Tr}[\mathbf{W}(\mathbf{I} - \rho\mathbf{W})^{-1}] = 0$ for all ρ which are not reciprocals of eigenvalues of \mathbf{W} .

Proof. (a) Let $\mathbf{U} = (U_1, \dots, U_n)^\top = \mathbf{W}\mathbf{Y}$. Using the law of iterated expectation, we get

$$\begin{aligned} E(\boldsymbol{\varepsilon}^\top \mathbf{W}\mathbf{Y}) &= E\left[(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n) \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{pmatrix}\right] \\ &= \sum_{i=1}^n E(\varepsilon_i U_i) = \sum_{i=1}^n E[E(\varepsilon_i U_i | Y_j; j \in J(i))]. \end{aligned} \quad (2.8)$$

Now, $U_i = (\mathbf{W}\mathbf{Y})_i = (w_{i1}, \dots, w_{in})(Y_1, \dots, Y_n)^\top = \sum_{\ell=1}^n w_{i\ell} Y_\ell$. We know that $w_{i\ell} = 0$ when $\ell \notin J(i)$. Plugging the value of U_i in equation (2.8), we get:

$$\begin{aligned} E(\boldsymbol{\varepsilon}^\top \mathbf{W}\mathbf{Y}) &= \sum_{i=1}^n E\left[E(\varepsilon_i \sum_{\ell \in J(i)} w_{i\ell} Y_\ell | Y_j; j \in J(i))\right] \\ &= \sum_{i=1}^n E\left[\sum_{\ell \in J(i)} w_{i\ell} Y_\ell E(\varepsilon_i | Y_j; j \in J(i))\right]. \end{aligned}$$

By equation (2.3),

$$E(\boldsymbol{\varepsilon}^\top \mathbf{W} \mathbf{Y}) = \sum_{i=1}^n E\left(\sum_{\ell \in J(i)} w_{i\ell} Y_\ell \cdot 0\right) = 0.$$

(b) Assume ρ is not a reciprocal of an eigenvalue of \mathbf{W} . Then

$$\det(\mathbf{I} - \rho \mathbf{W}) = (-\rho)^n \det\left(\mathbf{W} - \frac{1}{\rho} \mathbf{I}\right) \neq 0.$$

As a result, the matrix $\mathbf{I} - \rho \mathbf{W}$ is invertible. Then

$$\boldsymbol{\varepsilon} = (\mathbf{I} - \rho \mathbf{W})(\mathbf{Y} - \boldsymbol{\mu}) \implies \mathbf{Y} = \boldsymbol{\mu} + (\mathbf{I} - \rho \mathbf{W})^{-1} \boldsymbol{\varepsilon}.$$

Using part (a) we get:

$$\begin{aligned} 0 &= E(\boldsymbol{\varepsilon}^\top \mathbf{W} \mathbf{Y}) = E[\boldsymbol{\varepsilon}^\top \mathbf{W}(\boldsymbol{\mu} + (\mathbf{I} - \rho \mathbf{W})^{-1} \boldsymbol{\varepsilon})] \\ &= E[\boldsymbol{\varepsilon}^\top \mathbf{W} \boldsymbol{\mu} + \boldsymbol{\varepsilon}^\top \mathbf{W}(\mathbf{I} - \rho \mathbf{W})^{-1} \boldsymbol{\varepsilon}] \\ &= E[\boldsymbol{\varepsilon}^\top \mathbf{W} \boldsymbol{\mu}] + E[\boldsymbol{\varepsilon}^\top \mathbf{W}(\mathbf{I} - \rho \mathbf{W})^{-1} \boldsymbol{\varepsilon}] \\ &= E[\boldsymbol{\varepsilon}^\top] \mathbf{W} \boldsymbol{\mu} + \text{Tr}\{E[\boldsymbol{\varepsilon}^\top \mathbf{W}(\mathbf{I} - \rho \mathbf{W})^{-1} \boldsymbol{\varepsilon}]\} \\ &= \mathbf{0}^\top \mathbf{W} \boldsymbol{\mu} + E\{\text{Tr}[\boldsymbol{\varepsilon}^\top \mathbf{W}(\mathbf{I} - \rho \mathbf{W})^{-1} \boldsymbol{\varepsilon}]\} \\ &= 0 + E\{\text{Tr}[\mathbf{W}(\mathbf{I} - \rho \mathbf{W})^{-1} \boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^\top]\} \\ &= \text{Tr}[\mathbf{W}(\mathbf{I} - \rho \mathbf{W})^{-1} E(\boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^\top)] \\ &= \text{Tr}[\mathbf{W}(\mathbf{I} - \rho \mathbf{W})^{-1} \sigma^2 \mathbf{I}] \\ &= \sigma^2 \text{Tr}[\mathbf{W}(\mathbf{I} - \rho \mathbf{W})^{-1}] \\ &\implies \text{Tr}[\mathbf{W}(\mathbf{I} - \rho \mathbf{W})^{-1}] = 0. \end{aligned}$$

This completes the proof of part (b) of the lemma. □

Lemma 2.3. *If \mathbf{W} is either an upper or a lower triangular matrix, then equations (2.2) and (2.3) hold, and thus (from Lemma 2.2),*

$$E(\boldsymbol{\varepsilon}^\top \mathbf{W} \mathbf{Y}) = 0 \text{ and } \text{Tr}[\mathbf{W}(\mathbf{I} - \rho \mathbf{W})^{-1}] = 0.$$

Proof. Let the weight matrix \mathbf{W} be a lower triangular matrix. Then:

$$w_{ij} = 0 \text{ for } 1 \leq i < j \leq n \quad \text{and} \quad J(i) \subseteq \{1, 2, \dots, i-1\} \text{ for } 2 \leq i \leq n.$$

Let us consider the autoregressive model from equation (2.1):

$$Y_i = \alpha + \rho \sum_{j \in J(i)} w_{ij} Y_j + \varepsilon_i \quad (i = 1, \dots, n).$$

In the conditional expectation $E(\varepsilon_i | Y_j, j \in J(i))$ we condition on the values of some of the random variables Y_1, Y_2, \dots, Y_{i-1} . But the random variables Y_1, Y_2, \dots, Y_{i-1} depend on $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{i-1}$. Since the ε_k 's are independent with equal variances and zero means, we get

$$E(\varepsilon_i | Y_j, j \in J(i)) = E(\varepsilon_i) = 0.$$

Thus, equation (2.3) holds.

In addition,

$$\begin{aligned} E(Y_i | Y_j, j \in J(i)) &= E(\alpha + \rho \sum_{j \in J(i)} w_{ij} Y_j + \varepsilon_i | Y_\ell, \ell \in J(i)) \\ &= \alpha + \rho \sum_{j \in J(i)} w_{ij} Y_j + E(\varepsilon_i | Y_\ell, \ell \in J(i)) \\ &= \alpha + \rho \sum_{j \in J(i)} w_{ij} Y_j. \end{aligned}$$

Hence, equation (2.2) holds as well. □

2.3 The choice of weights in the weight matrix

An important and difficult step in spatial data analysis is to choose the weights w_{ij} in the spatial weight matrix \mathbf{W} based on the spatial influence among the different geographical areas. The criteria for choosing the elements of the weight matrix \mathbf{W} depend on the problem under study and the investigator. We can measure this spatial influence based on the length of shared border between geographical objects or based on the “crow fly” distance (i.e., the distance from the most direct path between two geographical centers). But the most commonly used approach is to choose weights w_{ij} that are based on “adjacency” (nearest neighbors).

In Section 2.3.1, we explain how to choose the weights in a regular grid, while in Section 2.3.2, we discuss how to choose the weights for an AR(1) time series model. Also, in Section 2.3.3, we give some procedures for choosing weights for an irregular lattice and in Section 2.3.4 we explain how to standardize a weight matrix.

2.3.1 Equally spaced locations

For equally spaced locations, we can set $w_{ij} > 0$ if and only if location j is a “neighbor” of location i . In a regular grid, generally, we consider the following forms of connections to find the weight matrix: connections by rook’s moves, by bishop’s moves and by queen’s moves (where a rook, a bishop, and a queen are common pieces in chess).

A location i in the center of a square of a regular grid is connected by a rook’s move to the locations in the centers of all the squares that are directly above, below,

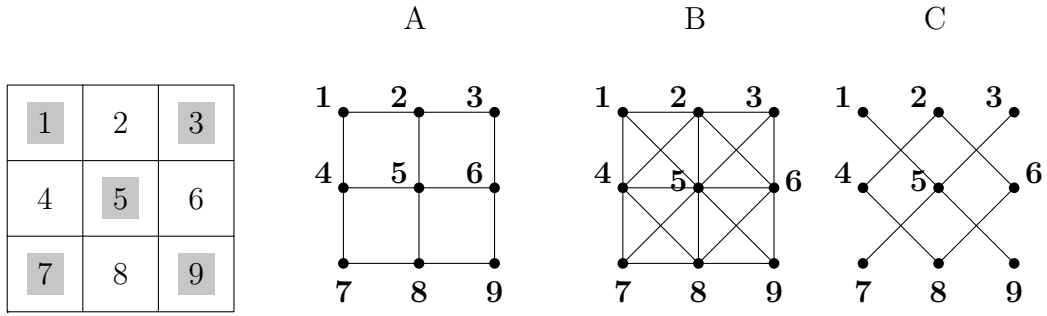


Figure 2.1: Different connections on a rectangular grid: connected by rook's moves (A), queen's move (B), and bishop's moves (C) (see Sokal and Oden [14, pp. 201 and 203])

to the left, or to the right of i in the grid. A queen's move connects a square to all the squares that are orthogonally or diagonally adjacent. A location i in a square of a grid is connected by a bishop's move to all the squares that are diagonally adjacent to the square.

Figure 2.1 shows the connections for these three moves. In the figure, the numbers $1, \dots, 9$ are the centers of the squares of a 3×3 lattice. Graphs A , B , and C in the figure show the different possible connections among the nine locations. Matrices \mathbf{W}_A , \mathbf{W}_B and \mathbf{W}_C , in equations (2.9), (2.10), and (2.11) below, show the weight matrices based on the rook's, queen's, and bishop's moves, respectively. In these three cases, location i is adjacent to location j if $w_{ij} = 1$, and not adjacent if $w_{ij} = 0$.

$$\mathbf{W}_A = \mathbf{W}_{\text{rook}} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{matrix} \quad (2.9)$$

$$\mathbf{W}_B = \mathbf{W}_{\text{queen}} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix} \end{matrix} \quad (2.10)$$

$$\mathbf{W}_C = \mathbf{W}_{\text{bishop}} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix} \quad (2.11)$$

In order to illustrate the notation from Section 2.1, we show the elements of each $J(i)$ for case C above (connection by bishop's moves). Recall that $J(i)$ is set of all neighbours of location i (and $J(i)$ does not include location i). We have:

$$\begin{aligned} J(1) &= \{5\}, & J(2) &= \{4, 6\}, & J(3) &= \{5\}, & J(4) &= \{2, 8\}, & J(5) &= \{1, 3, 7, 9\}, \\ J(6) &= \{2, 8\}, & J(7) &= \{5\}, & J(8) &= \{4, 6\}, & J(9) &= \{5\}. \end{aligned}$$

2.3.2 Weights for an AR(1) time series model

Recall that equation (2.1) refers to an autoregressive model that shows the interaction of location i with its neighbors for variable Y :

$$Y_i = \alpha + \rho \sum_{j \in J(i)} w_{ij} Y_j + \epsilon_i \quad (i = 1, \dots, n).$$

If the set of weights w_{ij} is defined by equation (2.5), then we get an AR(1) time series model:

$$Y_i = \alpha + \rho Y_{i-1} + \epsilon_i \quad (i = 2, \dots, n).$$

For example, when $n = 6$, the weight matrix becomes

$$\mathbf{W} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

2.3.3 Weights for an irregular lattice

In real world data, most of the time we have to deal with irregular lattice structures of many locations. In the case of an irregular lattice, instead of using the connectivity structures of a regular lattice (rook's, bishop's, or queen's moves), we use the inverse of

the distances among locations. As an example, see Figure 2.2 and the corresponding weight matrix \mathbf{W} in equation (2.12).

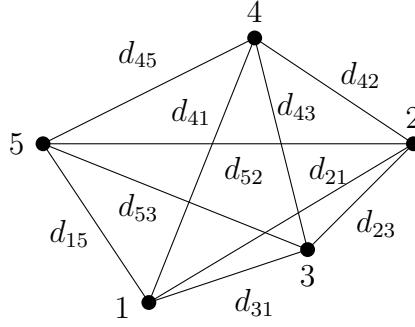


Figure 2.2: An example of an irregular lattice with five locations

$$\mathbf{W} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 0 & 1/d_{12} & 1/d_{13} & 1/d_{14} & 1/d_{15} \\ 1/d_{21} & 0 & 1/d_{23} & 1/d_{24} & 1/d_{25} \\ 1/d_{31} & 1/d_{32} & 0 & 1/d_{34} & 1/d_{35} \\ 1/d_{41} & 1/d_{42} & 1/d_{43} & 0 & 1/d_{45} \\ 1/d_{51} & 1/d_{52} & 1/d_{53} & 1/d_{54} & 0 \end{pmatrix} \end{matrix} \quad (2.12)$$

Here, for the distances between locations, we have $d_{ij} = d_{ji}$ for $1 \leq i \neq j \leq 5$.

The relative values of the Euclidean distances d_{ij} among the five points in Figure 2.2 (with respect to some unit of measurement) are as follows:

$$\begin{aligned} d_{21} &= 5.83, & d_{23} &= 2.83, & d_{42} &= 3.61, & d_{52} &= 7.00, & d_{31} &= 3.16, \\ d_{43} &= 4.12, & d_{53} &= 5.39, & d_{41} &= 5.39, & d_{15} &= 3.61, & d_{45} &= 4.47. \end{aligned}$$

Following Pace and Barry [11], we can create a weight matrix, \mathbf{W}_m , using the m^{th} nearest neighbour to each location i . For fixed $m \in \{1, \dots, n-1\}$, and for each

$i \in \{1, \dots, n\}$, we let

$$d_{(max,i)}^{(m)} := m^{th} \text{ order statistic of the distances } d_{i,1}, d_{i,2}, \dots, d_{i,i-1}, d_{i,i+1}, \dots, d_{i,n}.$$

(We use a modification of the notation in Pace and Barry [11] for the m^{th} order statistic. Thus, $d_{(max,i)}^{(1)}$ is the smallest observation among the elements of the i^{th} row (excluding d_{ii}), $d_{(max,i)}^{(2)}$ is the second smallest observation of those numbers, and so on.) For $1 \leq i, j \leq n$, define the (i, j) -th element of matrix \mathbf{W}_m by

$$w_{ij}^{(m)} = \begin{cases} 1, & \text{if } d_{ij} \leq d_{(max,i)}^{(m)} \quad \text{and} \quad i \neq j, \\ 0, & \text{otherwise.} \end{cases} \quad (2.13)$$

For $m = 1$, the distances from each location i to the first nearest neighbour are as follows:

$$d_{(max,1)}^{(1)} = 3.16, \quad d_{(max,2)}^{(1)} = 2.83, \quad d_{(max,3)}^{(1)} = 2.83, \quad d_{(max,4)}^{(1)} = 3.61, \quad d_{(max,5)}^{(1)} = 3.61.$$

Then the corresponding weight matrix is as follows:

$$\mathbf{W}_1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix} \quad (2.14)$$

Also, for $m = 2$, the distances from each location i to the second nearest neighbour are as follows:

$$d_{(max,1)}^{(2)} = 3.61, \quad d_{(max,2)}^{(2)} = 3.61, \quad d_{(max,3)}^{(2)} = 3.16, \quad d_{(max,4)}^{(2)} = 4.12, \quad d_{(max,5)}^{(2)} = 4.47.$$

Then the corresponding weight matrix is as follows:

$$\mathbf{W}_2 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix} \quad (2.15)$$

In a similar way, we can create the weight matrices \mathbf{W}_3 , \mathbf{W}_4 , and \mathbf{W}_5 for the cases $m = 3$, $m = 4$, and $m = 5$, respectively.

2.3.4 Standardization of the weight matrix \mathbf{W}

In all previous examples, the weight matrix \mathbf{W} depends on the proximity of each location to the other locations. Occasionally, we have to standardize the weight matrix \mathbf{W} , and create a new weight matrix \mathbf{S} , using the following formula:

$$\mathbf{S}_{ij} = \frac{w_{ij}}{\sum_{\ell=1}^n w_{i\ell}} \quad \text{for } 1 \leq i, j \leq n.$$

The above equation implies $\mathbf{S}\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is an $n \times 1$ vector of 1's.

Example (Standardization of matrices $\mathbf{W}_{\text{queen}}$ and \mathbf{W}_2): For a regular lattice, for the queen’s moves in Figure 2.1, the corresponding weight matrix $\mathbf{W}_{\text{queen}}$, in equation (2.10), can be standardized as follows:

$$\mathbf{S}_{\text{queen}} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \left(\begin{array}{ccccccccc} 0 & 0.333 & 0 & 0.333 & 0.333 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 \\ 0 & 0.333 & 0 & 0 & 0.333 & 0.333 & 0 & 0 & 0 \\ 0.2 & 0.2 & 0 & 0 & 0.2 & 0 & 0.2 & 0.2 & 0 \\ 0.125 & 0.125 & 0.125 & 0.125 & 0 & 0.125 & 0.125 & 0.125 & 0.125 \\ 0 & 0.2 & 0.2 & 0 & 0.2 & 0 & 0 & 0.2 & 0.2 \\ 0 & 0 & 0 & 0.333 & 0.333 & 0 & 0 & 0.333 & 0 \\ 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0.2 \\ 0 & 0 & 0 & 0 & 0.333 & 0.333 & 0 & 0.333 & 0 \end{array} \right) \end{matrix}$$

For an irregular lattice, like in Section 2.3.3, the weight matrix \mathbf{W}_2 from equation (2.15), can be standardized as follows:

$$\mathbf{S}_2 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \left(\begin{array}{ccccc} 0 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 & 0 \end{array} \right) \end{matrix}$$

Root [13] gave the following reasons for using standardized weight matrices:

1. “Row standardization is used to create proportional weights in cases where features have an unequal number of neighbors.”
2. “Use [if] you want comparable spatial parameters across different data sets with different connectivity structures.”

3. “Row-standardized weights increase the influence of links from observations with few neighbours.”

The following lemma states that the eigenvalues of a row standardized matrix are less than or equal to 1 in absolute value. The result is related to the Perron-Frobenius theorem about square matrices (see Marshall and Olkin [8, Chapter 9, Section I]), but it is not exactly that theorem. The proof below is based on a similar result in Matthews [7]. The result becomes important in Section 2.4.4, where we explain how Ord’s eigenvalue method can be used to estimate the parameter ρ . (Here, for a complex number $\lambda = \Re(\lambda) + \Im(\lambda)\mathbf{i}$, with $\mathbf{i} = \sqrt{-1}$, the *modulus* is $\|\lambda\| = \sqrt{\Re(\lambda)^2 + \Im(\lambda)^2}$.)

Lemma 2.4. *Let \mathbf{W} be a row-standardized matrix with non-negative elements. Then \mathbf{W} has (possibly complex) eigenvalues $\lambda_1, \dots, \lambda_n$ such that $\|\lambda_i\| \leq 1$ for $i = 1, \dots, n$, and one of the eigenvalues is equal to 1.*

Proof. Let \mathbf{W} be a row-standardized matrix, that is, each row of \mathbf{W} sums up to 1. Then $\mathbf{W}\mathbf{1} = \mathbf{1}$, which shows that 1 is an eigenvalue of \mathbf{W} with $\mathbf{1}$ as an eigenvector. Now, let λ be an eigenvalue of \mathbf{W} and \mathbf{x} be a corresponding non-zero (possibly complex) eigenvector. Then $\mathbf{W}\mathbf{x} = \lambda\mathbf{x}$. Equating the i^{th} components of each side, we get:

$$\sum_{j=1}^n w_{ij}x_j = \lambda x_i \quad \text{for } i = 1, 2, \dots, n. \quad (2.16)$$

Let x_k be an entry of \mathbf{x} such that

$$\|x_k\| = \max\{\|x_1\|, \|x_2\|, \dots, \|x_n\|\}.$$

Note that $\|x_k\| > 0$ because \mathbf{x} is a non-zero eigenvector.

Now, for $i = k$, taking modulus on both sides of equation (2.16), we get:

$$\|\lambda x_k\| = \|\lambda\| \|x_k\| = \left\| \sum_{j=1}^n w_{kj} x_j \right\| \leq \sum_{j=1}^n w_{kj} \|x_j\| \leq \sum_{j=1}^n w_{kj} \|x_k\| = \|x_k\|$$

because $w_{kj} \geq 0$ and $\sum_{j=1}^n w_{kj} = 1$. Thus, $\|\lambda\| \|x_k\| \leq \|x_k\|$. Since $\|x_k\| > 0$, we get $\|\lambda\| \leq 1$. □

2.4 Estimation using maximum likelihood

As Ord [9] states, if the conditional assumption (2.3) does not hold, then the OLS estimators for the joint probability model $\mathbf{Y} = \boldsymbol{\alpha} + \rho \mathbf{W}\mathbf{Y} + \boldsymbol{\varepsilon}$ are inconsistent.

This is the reason we need an alternative procedure to estimate the parameters. In this section, we explain how to estimate the parameters α , σ^2 , and ρ in the model $\mathbf{Y} = \boldsymbol{\alpha} + \rho \mathbf{W}\mathbf{Y} + \boldsymbol{\varepsilon}$ using the maximum likelihood method.

In Section 2.4.1, we explain a general estimation procedure for the parameters using the MLE method, while in Section 2.4.2, we explain the estimation procedure of ρ in the conditional probability model (2.2). Also, in Section 2.4.3, we discuss Ord's eigenvalue method for solving the MLE equations that we got in Section 2.4.1. In particular, we use Lemma 2.4 to avoid singularities in MLE equations while using Ord's eigenvalue method for row-standardized matrices \mathbf{W} . Finally, in Section 2.4.4, we estimate the parameters α , σ^2 , and ρ by implementing Ord's method using the **R** programming language for the case of row-standardized weight matrices.

2.4.1 General estimation procedure for the parameters

In the following theorem, we only require the weight matrix \mathbf{W} to have non-negative elements. In general, it does not have to be symmetric or row-standardized. Equations (2.17) and (2.18) appear in Ord [9], while equation (2.19) does not.

Theorem 2.1. *Assume $\boldsymbol{\varepsilon} \sim \text{MVN}(\mathbf{0}, \sigma^2 \mathbf{I})$ and $\mathbf{Y} = \boldsymbol{\alpha} + \rho \mathbf{W} \mathbf{Y} + \boldsymbol{\varepsilon}$. If the MLEs of the parameters α , σ^2 , and ρ exist, then they satisfy the following equations:*

$$\hat{\alpha} = \frac{\mathbf{1}^\top (\mathbf{I} - \hat{\rho} \mathbf{W}) \mathbf{y}}{n}, \quad (2.17)$$

$$\hat{\sigma}^2 = \frac{1}{n} \mathbf{y}^\top (\mathbf{I} - \hat{\rho} \mathbf{W})^\top \left[\mathbf{I} - \frac{\mathbf{1} \mathbf{1}^\top}{n} \right] (\mathbf{I} - \hat{\rho} \mathbf{W}) \mathbf{y}, \quad (2.18)$$

$$\text{and } -\hat{\sigma}^2 \text{Tr}[(\mathbf{I} - \hat{\rho} \mathbf{W})^{-1} \mathbf{W}] + \mathbf{y}^\top \mathbf{W} \mathbf{y} - \hat{\rho} \mathbf{y}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y} - \hat{\boldsymbol{\alpha}}^\top \mathbf{W} \mathbf{y} = 0. \quad (2.19)$$

Proof. The proof of the theorem is divided into four parts: (a) description of the likelihood equation, (b) estimation of α , (c) estimation of σ^2 , and (d) estimation of ρ .

(a) Description of the likelihood equation. We first reformulate equation (2.7). Since $\boldsymbol{\varepsilon} = (\mathbf{I} - \rho \mathbf{W})(\mathbf{Y} - \boldsymbol{\mu})$, we can solve for \mathbf{Y} :

$$\mathbf{Y} = \boldsymbol{\mu} + (\mathbf{I} - \rho \mathbf{W})^{-1} \boldsymbol{\varepsilon} \implies \mathbf{Y} = \boldsymbol{\mu} + \mathbf{A}^{-1} \boldsymbol{\varepsilon}.$$

Here, $\mathbf{A} = \mathbf{I} - \rho \mathbf{W}$. We also know that $E[\mathbf{Y}] = \boldsymbol{\mu}$. In addition,

$$\text{Cov}(\mathbf{Y}) = \text{Cov}(\boldsymbol{\mu} + \mathbf{A}^{-1} \boldsymbol{\varepsilon}) = \mathbf{A}^{-1} \text{Cov}(\boldsymbol{\varepsilon}) (\mathbf{A}^{-1})^\top \implies \text{Cov}(\mathbf{Y}) = \sigma^2 (\mathbf{A}^\top \mathbf{A})^{-1} = \boldsymbol{\Sigma}.$$

Since $\boldsymbol{\varepsilon} \sim \text{MVN}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$, we have $\mathbf{Y} \sim \text{MVN}(\boldsymbol{\mu}, \sigma^2 (\mathbf{A}^\top \mathbf{A})^{-1})$. It follows that the joint p.d.f of \mathbf{Y} given α , σ^2 , and ρ is:

$$f(\mathbf{y}; \alpha, \sigma^2, \rho) = \frac{1}{(2\pi)^{n/2} (\det \boldsymbol{\Sigma})^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{y} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu}) \right]$$

$$\begin{aligned}
&= \frac{1}{(2\pi)^{n/2}[\det(\sigma^2(\mathbf{A}^\top \mathbf{A})^{-1})]^{1/2}} \exp\left[-\frac{1}{2\sigma^2}(\mathbf{y} - \boldsymbol{\mu})^\top (\mathbf{A}^\top \mathbf{A})(\mathbf{y} - \boldsymbol{\mu})\right] \\
&= \frac{1}{(2\pi)^{n/2}\sigma^{2n/2}(\det \mathbf{A})^{-2/2}} \exp\left[-\frac{1}{2\sigma^2}(\mathbf{y}^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} - \mathbf{y}^\top \mathbf{A}^\top \mathbf{A} \boldsymbol{\mu} - \boldsymbol{\mu}^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} + \boldsymbol{\mu}^\top \mathbf{A}^\top \mathbf{A} \boldsymbol{\mu})\right] \\
&= \frac{\det(\mathbf{A})}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{1}{2\sigma^2}(\mathbf{y}^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} - 2\boldsymbol{\mu}^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} + \boldsymbol{\mu}^\top \mathbf{A}^\top \mathbf{A} \boldsymbol{\mu})\right].
\end{aligned}$$

Now, the log-likelihood from the above joint pdf of \mathbf{Y} , given α , σ^2 , and ρ , is:

$$\ell(\alpha, \sigma^2, \rho | \mathbf{y}) = \ln(\det \mathbf{A}) - \frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}[\mathbf{y}^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} - 2\boldsymbol{\mu}^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} + \boldsymbol{\mu}^\top \mathbf{A}^\top \mathbf{A} \boldsymbol{\mu}].$$

Since $\boldsymbol{\alpha} = \mathbf{A}\boldsymbol{\mu}$, we have $\boldsymbol{\mu} = \mathbf{A}^{-1}\boldsymbol{\alpha}$. Plugging the value of $\boldsymbol{\mu}$ in the above equation, we get:

$$\begin{aligned}
\ell(\alpha, \sigma^2, \rho | \mathbf{y}) &= \ln(\det \mathbf{A}) - \frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}[\mathbf{y}^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} - 2(\mathbf{A}^{-1}\boldsymbol{\alpha})^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} \\
&\quad + (\mathbf{A}^{-1}\boldsymbol{\alpha})^\top \mathbf{A}^\top \mathbf{A} (\mathbf{A}^{-1}\boldsymbol{\alpha})] \\
&= \ln(\det \mathbf{A}) - \frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}[\mathbf{y}^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} - 2\boldsymbol{\alpha}^\top (\mathbf{A}^\top)^{-1} \mathbf{A}^\top \mathbf{A} \mathbf{y} \\
&\quad + \boldsymbol{\alpha}^\top (\mathbf{A}^\top)^{-1} \mathbf{A}^\top \mathbf{A} \mathbf{A}^{-1} \boldsymbol{\alpha}] \\
&= \ln(\det \mathbf{A}) - \frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{y}^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} - 2\boldsymbol{\alpha}^\top \mathbf{A} \mathbf{y} + \boldsymbol{\alpha}^\top \boldsymbol{\alpha}). \quad (2.20)
\end{aligned}$$

Let $\boldsymbol{\gamma} = \mathbf{A}\mathbf{y}$. Then we may re-write the log-likelihood as follows:

$$\begin{aligned}
\ell(\alpha, \sigma^2, \rho | \mathbf{y}) &= \ln(\det \mathbf{A}) - \frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \left[\boldsymbol{\gamma}^\top \boldsymbol{\gamma} - 2 \begin{pmatrix} \alpha & \alpha & \cdots & \alpha \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_n \end{pmatrix} \right. \\
&\quad \left. + \begin{pmatrix} \alpha & \alpha & \cdots & \alpha \end{pmatrix} \begin{pmatrix} \alpha \\ \alpha \\ \vdots \\ \alpha \end{pmatrix} \right]
\end{aligned}$$

$$\begin{aligned}
&= \ln(\det \mathbf{A}) - \frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} [\boldsymbol{\gamma}^\top \boldsymbol{\gamma} - 2(\alpha\gamma_1 + \alpha\gamma_2 + \dots + \alpha\gamma_n) + n\alpha^2] \\
&= \ln(\det \mathbf{A}) - \frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} [\boldsymbol{\gamma}^\top \boldsymbol{\gamma} - 2\alpha \mathbf{1}^\top \boldsymbol{\gamma} + n\alpha^2],
\end{aligned}$$

where $\mathbf{1} = (1, 1, \dots, 1)^\top$ is $n \times 1$.

(b) Estimation of α . Taking the derivative of the log-likelihood function with respect to α and setting that derivative equal to zero, we get the MLE of α :

$$\frac{\partial \ell}{\partial \alpha} = -\frac{1}{2\sigma^2} (-2\mathbf{1}^\top \boldsymbol{\gamma} + 2n\alpha) \stackrel{\text{set}}{=} 0 \implies \hat{\alpha} = \frac{\mathbf{1}^\top (\mathbf{I} - \hat{\rho}\mathbf{W})\mathbf{y}}{n}.$$

(c) Estimation of σ^2 . Similarly, we can find the MLE of σ^2 by taking derivative of the log-likelihood function with respect to σ^2 and setting that derivative equal to zero:

$$\begin{aligned}
\frac{\partial \ell}{\partial \sigma^2} &= -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} (\mathbf{y}^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} - 2\boldsymbol{\mu}^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} + \boldsymbol{\mu}^\top \mathbf{A}^\top \mathbf{A} \boldsymbol{\mu}) \stackrel{\text{set}}{=} 0 \\
&\implies \frac{1}{2\sigma^4} (\mathbf{y} - \boldsymbol{\mu})^\top \mathbf{A}^\top \mathbf{A} (\mathbf{y} - \boldsymbol{\mu}) = \frac{n}{2\sigma^2} \\
&\implies \hat{\sigma}^2 = \frac{1}{n} (\mathbf{y} - \hat{\boldsymbol{\mu}})^\top \hat{\mathbf{A}}^\top \hat{\mathbf{A}} (\mathbf{y} - \hat{\boldsymbol{\mu}}),
\end{aligned}$$

where $\hat{\mathbf{A}} = (\mathbf{I} - \hat{\rho}\mathbf{W})$ and $\hat{\boldsymbol{\mu}} = \hat{\mathbf{A}}^{-1} \hat{\boldsymbol{\alpha}}$. The MLE of σ^2 can then be re-written as:

$$\begin{aligned}
\hat{\sigma}^2 &= \frac{1}{n} (\hat{\mathbf{A}}\mathbf{y} - \hat{\mathbf{A}}\hat{\boldsymbol{\mu}})^\top (\hat{\mathbf{A}}\mathbf{y} - \hat{\mathbf{A}}\hat{\boldsymbol{\mu}}) \\
&= \frac{1}{n} (\hat{\mathbf{A}}\mathbf{y} - \hat{\boldsymbol{\alpha}})^\top (\hat{\mathbf{A}}\mathbf{y} - \hat{\boldsymbol{\alpha}}) \\
&= \frac{1}{n} (\hat{\mathbf{A}}\mathbf{y} - \frac{\mathbf{1}^\top \hat{\mathbf{A}}\mathbf{y}}{n} \mathbf{1})^\top (\hat{\mathbf{A}}\mathbf{y} - \frac{\mathbf{1}^\top \hat{\mathbf{A}}\mathbf{y}}{n} \mathbf{1}) \\
&= \frac{1}{n} (\hat{\mathbf{A}}\mathbf{y} - \frac{\mathbf{1}\mathbf{1}^\top}{n} \hat{\mathbf{A}}\mathbf{y})^\top (\hat{\mathbf{A}}\mathbf{y} - \frac{\mathbf{1}\mathbf{1}^\top}{n} \hat{\mathbf{A}}\mathbf{y}) \\
&= \frac{1}{n} (\hat{\mathbf{A}}\mathbf{y})^\top [\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n}]^\top [\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n}] (\hat{\mathbf{A}}\mathbf{y})
\end{aligned}$$

Since $\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n}$ is idempotent and symmetric, we have

$$\hat{\sigma}^2 = \frac{1}{n} \mathbf{y}^\top (\mathbf{I} - \hat{\rho}\mathbf{W})^\top [\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n}] (\mathbf{I} - \hat{\rho}\mathbf{W}) \mathbf{y}. \quad (2.21)$$

(d) **Estimation of ρ .** To find the MLE of ρ , we need to differentiate the log-likelihood function with respect to ρ and set the derivative equal to zero:

$$\frac{\partial \ell}{\partial \rho} = \frac{\partial \ln(\det \mathbf{A})}{\partial \rho} - 0 - \frac{1}{2\sigma^2} [\mathbf{y}^\top \frac{\partial(\mathbf{A}^\top \mathbf{A})}{\partial \rho} \mathbf{y} - 2\boldsymbol{\alpha}^\top \frac{\partial \mathbf{A}}{\partial \rho} \mathbf{y} + 0]. \quad (2.22)$$

From Jacobi's formula, we know that

$$\frac{\partial(\det \mathbf{B}(t))}{\partial t} = \text{Tr}[\text{adj}(\mathbf{B}(t)) \frac{\partial \mathbf{B}(t)}{\partial t}], \quad (2.23)$$

where $\text{adj}(\mathbf{B}(t))$ is the *adjugate* of the matrix $\mathbf{B}(t)$, that is, the transpose of its cofactor matrix. Therefore,

$$\frac{\partial \ln(\det \mathbf{A})}{\partial \rho} = \frac{1}{\det \mathbf{A}} \times \frac{\partial}{\partial \rho} \det \mathbf{A} = \frac{1}{\det \mathbf{A}} \text{Tr}[\text{adj}(\mathbf{A}) \frac{\partial \mathbf{A}}{\partial \rho}],$$

Since $\frac{1}{\det \mathbf{A}}$ is a scalar, we can write the above equation as follows:

$$\frac{\partial \ln(\det \mathbf{A})}{\partial \rho} = \text{Tr}[\frac{1}{\det \mathbf{A}} \text{adj}(\mathbf{A}) \frac{\partial \mathbf{A}}{\partial \rho}].$$

Again, we know $\mathbf{B}(t)^{-1} = \frac{1}{\det \mathbf{B}(t)} \text{adj}(\mathbf{B}(t))$. So, the above equation can be written as:

$$\frac{\partial \ln(\det \mathbf{A})}{\partial \rho} = \text{Tr}[\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \rho}] = \text{Tr}[(\mathbf{I} - \rho \mathbf{W})^{-1} \frac{\partial(\mathbf{I} - \rho \mathbf{W})}{\partial \rho}] = -\text{Tr}[(\mathbf{I} - \rho \mathbf{W})^{-1} \mathbf{W}]. \quad (2.24)$$

Also,

$$\begin{aligned} \frac{\partial(\mathbf{A}^\top \mathbf{A})}{\partial \rho} &= \frac{\partial[(\mathbf{I} - \rho \mathbf{W})^\top (\mathbf{I} - \rho \mathbf{W})]}{\partial \rho} \\ &= -\mathbf{W}^\top (\mathbf{I} - \rho \mathbf{W}) - (\mathbf{I} - \rho \mathbf{W})^\top \mathbf{W} \\ &= -\mathbf{W}^\top + \rho \mathbf{W}^\top \mathbf{W} - \mathbf{W} + \rho \mathbf{W}^\top \mathbf{W} \\ &= -(\mathbf{W}^\top + \mathbf{W}) + 2\rho \mathbf{W}^\top \mathbf{W}. \end{aligned}$$

Also, $\frac{\partial \mathbf{A}}{\partial \rho} = \frac{\partial(\mathbf{I} - \rho \mathbf{W})}{\partial \rho} = -\mathbf{W}$.

Plugging the above expressions for $\frac{\partial(\det \mathbf{A})}{\partial \rho}$, $\frac{\partial(\mathbf{A}^\top \mathbf{A})}{\partial \rho}$, and $\frac{\partial \mathbf{A}}{\partial \rho}$ into equation (2.22) and setting the derivative equal to zero, we get an equation for the MLE of ρ :

$$\begin{aligned} \frac{\partial \ell}{\partial \rho} &= -\text{Tr}[(\mathbf{I} - \hat{\rho} \mathbf{W})^{-1} \mathbf{W}] + \frac{1}{2\hat{\sigma}^2} [\mathbf{y}^\top (\mathbf{W}^\top + \mathbf{W}) \mathbf{y}] - \frac{\hat{\rho}}{\hat{\sigma}^2} \mathbf{y}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y} - \frac{1}{\hat{\sigma}^2} \hat{\boldsymbol{\alpha}}^\top \mathbf{W} \mathbf{y} \stackrel{\text{set}}{=} 0 \\ \implies -\hat{\sigma}^2 \text{Tr}[(\mathbf{I} - \hat{\rho} \mathbf{W})^{-1} \mathbf{W}] + \frac{1}{2} [\mathbf{y}^\top \mathbf{W}^\top \mathbf{y} + \mathbf{y}^\top \mathbf{W} \mathbf{y}] - \hat{\rho} \mathbf{y}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y} - \hat{\boldsymbol{\alpha}}^\top \mathbf{W} \mathbf{y} &= 0 \\ \implies -\hat{\sigma}^2 \text{Tr}[(\mathbf{I} - \hat{\rho} \mathbf{W})^{-1} \mathbf{W}] + \mathbf{y}^\top \mathbf{W} \mathbf{y} - \hat{\rho} \mathbf{y}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y} - \hat{\boldsymbol{\alpha}}^\top \mathbf{W} \mathbf{y} &= 0. \end{aligned}$$

This completes the proof of the theorem. \square

Remark 2.1. In general, the system of equations (2.17)–(2.19) can have more than one solution $(\hat{\boldsymbol{\alpha}}, \hat{\sigma}^2, \hat{\rho})$. In addition, in Theorem 2.1, we need to check that the 3×3 matrix of second partial derivatives of the log-likelihood (with respect to the parameters) is negative definite at the MLE point. Due to time constraints, we did not investigate that.

2.4.2 Estimation of the parameters in the conditional model

In general, it is very difficult to solve the equation (2.19) for $\hat{\rho}$ explicitly. But if the conditional assumption (2.3) holds, then we can get a closed form solution for $\hat{\rho}$.

Lemma 2.5. *If equation (2.3) holds, then the MLE of ρ becomes:*

$$\begin{aligned} \hat{\rho} &= \left(\mathbf{y}^\top \mathbf{W} \mathbf{y} - \frac{1}{n} \mathbf{y}^\top \mathbf{1} \mathbf{1}^\top \mathbf{W} \mathbf{y} \right) \left(\mathbf{y}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y} - \frac{1}{n} \mathbf{y}^\top \mathbf{W}^\top \mathbf{1} \mathbf{1}^\top \mathbf{W} \mathbf{y} \right)^{-1} \\ &= \frac{\mathbf{y}^\top [\mathbf{I} - \frac{1}{n} \mathbf{1} \mathbf{1}^\top] \mathbf{W} \mathbf{y}}{\mathbf{y}^\top \mathbf{W}^\top [\mathbf{I} - \frac{1}{n} \mathbf{1} \mathbf{1}^\top] \mathbf{W} \mathbf{y}}. \end{aligned} \tag{2.25}$$

Proof. Assume that equation (2.3) holds. From Lemma 2.2, part (b), in Section 2.2, we have ¹ $\text{Tr}[\mathbf{W}(\mathbf{I} - \hat{\rho}\mathbf{W})^{-1}] = \text{Tr}[(\mathbf{I} - \hat{\rho}\mathbf{W})^{-1}\mathbf{W}] = 0$. Plugging this value and the value of $\hat{\alpha}$ from (2.17) into equation (2.19), we get:

$$\begin{aligned}
& \mathbf{y}^\top \mathbf{W} \mathbf{y} - \hat{\rho} \mathbf{y}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y} - \left[\frac{\mathbf{1}^\top (\mathbf{I} - \hat{\rho} \mathbf{W}) \mathbf{y}}{n} \right]^\top \mathbf{1}^\top \mathbf{W} \mathbf{y} = 0 \\
\implies & \mathbf{y}^\top \mathbf{W} \mathbf{y} - \hat{\rho} \mathbf{y}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y} - \frac{1}{n} \mathbf{y}^\top (\mathbf{I} - \hat{\rho} \mathbf{W})^\top \mathbf{1} \mathbf{1}^\top \mathbf{W} \mathbf{y} = 0 \\
\implies & \mathbf{y}^\top \mathbf{W} \mathbf{y} - \hat{\rho} \mathbf{y}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y} - \frac{1}{n} \mathbf{y}^\top \mathbf{1} \mathbf{1}^\top \mathbf{W} \mathbf{y} + \frac{1}{n} \hat{\rho} \mathbf{y}^\top \mathbf{W}^\top \mathbf{1} \mathbf{1}^\top \mathbf{W} \mathbf{y} = 0 \\
\implies & \mathbf{y}^\top \mathbf{W} \mathbf{y} - \frac{1}{n} \mathbf{y}^\top \mathbf{1} \mathbf{1}^\top \mathbf{W} \mathbf{y} = \hat{\rho} \left[\mathbf{y}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y} - \frac{1}{n} \mathbf{y}^\top \mathbf{W}^\top \mathbf{1} \mathbf{1}^\top \mathbf{W} \mathbf{y} \right] \\
\implies & \hat{\rho} = \left[\mathbf{y}^\top \mathbf{W} \mathbf{y} - \frac{1}{n} \mathbf{y}^\top \mathbf{1} \mathbf{1}^\top \mathbf{W} \mathbf{y} \right] \left[\mathbf{y}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y} - \frac{1}{n} \mathbf{y}^\top \mathbf{W}^\top \mathbf{1} \mathbf{1}^\top \mathbf{W} \mathbf{y} \right]^{-1} \\
\implies & \hat{\rho} = \{ \mathbf{y}^\top [\mathbf{I} - \frac{1}{n} \mathbf{1} \mathbf{1}^\top] \mathbf{W} \mathbf{y} \} \{ \mathbf{y}^\top \mathbf{W}^\top [\mathbf{I} - \frac{1}{n} \mathbf{1} \mathbf{1}^\top] \mathbf{W} \mathbf{y} \}^{-1}.
\end{aligned}$$

This completes the proof of the lemma. □

Example (AR(1) time series model): Consider the AR(1) time series model $Y_i = \alpha + \rho Y_{i-1} + \epsilon_i$ for $i = 2, \dots, n$. From equation (2.5) in Section 2.1, the non-negative weights $\{w_{ij}\}$ are as follows:

$$w_{ij} = \begin{cases} 1, & \text{if } j = i - 1 \ (i \geq 2), \\ 0, & \text{elsewhere.} \end{cases}$$

Using the corresponding weight matrix \mathbf{W} , we can re-express equation (2.25) as:

$$\hat{\rho} = \frac{\sum_{i=2}^n y_i y_{i-1} - \frac{1}{n} \sum_{i=1}^n y_i \sum_{j=1}^{n-1} y_j}{\sum_{i=1}^{n-1} y_i^2 - \frac{1}{n} (\sum_{i=1}^{n-1} y_i)^2}. \tag{2.26}$$

¹Strictly speaking, according to Lemma 2.2, the value of $\hat{\rho}$ in equation (2.5) should *not* be the reciprocal of an eigenvalue of the weight matrix \mathbf{W} , but this seems unlikely in practice.

Now, if the sample size n is large, then

$$n \approx n - 1 \quad \text{and} \quad \frac{1}{n} \sum_{i=1}^n y_i \sum_{j=1}^{n-1} y_j \approx \frac{1}{n-1} \sum_{i=1}^{n-1} y_i \sum_{j=2}^n y_j.$$

Let

$$\bar{y}^{(1)} = \frac{1}{n-1} \sum_{i=1}^{n-1} y_i \quad \text{and} \quad \bar{y}^{(2)} = \frac{1}{n-1} \sum_{i=2}^n y_i.$$

After some algebra, equation (2.26) for $\hat{\rho}$ can be re-expressed approximately as:

$$\hat{\rho} \approx \frac{\sum_{i=2}^n y_i y_{i-1} - \frac{1}{n-1} \sum_{i=1}^{n-1} y_i \sum_{j=2}^n y_j}{\sum_{i=1}^{n-1} y_i^2 - \frac{1}{n-1} (\sum_{i=1}^{n-1} y_i)^2} = \frac{\sum_{i=2}^n (y_{i-1} - \bar{y}^{(1)})(y_i - \bar{y}^{(2)})}{\sum_{i=2}^n (y_{i-1} - \bar{y}^{(1)})^2}.$$

The last expression is an estimate of the (first order) autocorrelation of the time series (Y_1, Y_2, \dots, Y_n) . Thus, for large n , the estimate of ρ we get from equation (2.25) is approximately equal to an estimate of the (first order) autocorrelation.

2.4.3 Ord's procedure for solving the MLE equations

The ML equations (2.17), (2.18), and (2.19) cannot be solved explicitly. However, we can solve them numerically. But then the main difficulty arises when we try to determine $\hat{\rho}$, because equation (2.19) involves the inverse of the matrix $\mathbf{I} - \hat{\rho}\mathbf{W}$. As a result, when the sample size n is large, the computation of $\hat{\rho}$ becomes more time-consuming. To circumvent this difficulty, Ord [9] uses the eigenvalues of the weight matrix \mathbf{W} .

Lemma 2.6. *If \mathbf{W} has (possibly complex) eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$, then*

$$\det(\mathbf{I} - \rho\mathbf{W}) = \prod_{i=1}^n (1 - \rho\lambda_i), \quad (2.27)$$

and

$$\frac{\partial \ln \det(\mathbf{I} - \rho\mathbf{W})}{\partial \rho} = - \sum_{i=1}^n \frac{\lambda_i}{1 - \rho\lambda_i}. \quad (2.28)$$

Proof. Since $\lambda_1, \lambda_2, \dots, \lambda_n$, are the eigenvalues of \mathbf{W} , we can write

$$\det(h\mathbf{I} - \mathbf{W}) = \prod_{i=1}^n (h - \lambda_i).$$

Then, from the above equation, we get:

$$\begin{aligned} \det(h\mathbf{I} - \mathbf{W}) = \prod_{i=1}^n (h - \lambda_i) &\implies h^n \det(\mathbf{I} - \mathbf{W}h^{-1}) = h^n \prod_{i=1}^n (1 - \lambda_i h^{-1}) \\ &\implies \det(\mathbf{I} - \mathbf{W}h^{-1}) = \prod_{i=1}^n (1 - \lambda_i h^{-1}). \end{aligned}$$

Let $\rho = h^{-1}$, then we get:

$$\det(\mathbf{I} - \rho\mathbf{W}) = \prod_{i=1}^n (1 - \rho\lambda_i). \quad (2.29)$$

This completes the proof of first part of this lemma.

Now, taking the natural log of equation (2.29) and differentiating with respect to ρ , we get the following:

$$\begin{aligned} \ln \det(\mathbf{I} - \rho\mathbf{W}) &= \ln(1 - \rho\lambda_1) + \ln(1 - \rho\lambda_2) + \dots + \ln(1 - \rho\lambda_n) \\ \implies \frac{\partial \ln \det(\mathbf{I} - \rho\mathbf{W})}{\partial \rho} &= -\frac{\lambda_1}{1 - \rho\lambda_1} - \frac{\lambda_2}{1 - \rho\lambda_2} - \dots - \frac{\lambda_n}{1 - \rho\lambda_n} = -\sum_{i=1}^n \frac{\lambda_i}{1 - \rho\lambda_i}. \end{aligned}$$

This completes the proof of the second part of the lemma. \square

Remark 2.2. From equations (2.24) and (2.28) it follows that:

$$\text{Tr}[(\mathbf{I} - \rho\mathbf{W})^{-1}\mathbf{W}] = \sum_{i=1}^n \frac{\lambda_i}{1 - \rho\lambda_i}. \quad (2.30)$$

Remark 2.3. Equation (2.18) can be expressed as a polynomial in $\hat{\rho}$ as follows:

$$\hat{\sigma}^2 = \hat{\rho}^2 \mathbf{y}^\top [\mathbf{W}^\top \frac{1}{n} (\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n}) \mathbf{W}] \mathbf{y} - 2\hat{\rho} \mathbf{y}^\top [\mathbf{W}^\top \frac{1}{n} (\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n})] \mathbf{y} + \mathbf{y}^\top \frac{1}{n} (\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n}) \mathbf{y}.$$

Theorem 2.2. *The MLE of the parameter ρ satisfies the following equation:*

$$-\frac{\hat{\rho}^2}{n} \left[h_1 \sum_{i=1}^n \frac{\lambda_i}{1 - \hat{\rho}\lambda_i} \right] + \hat{\rho} \left[-h_1 + \frac{2h_2}{n} \sum_{i=1}^n \frac{\lambda_i}{1 - \hat{\rho}\lambda_i} \right] + \left[h_2 - \frac{h_3}{n} \sum_{i=1}^n \frac{\lambda_i}{1 - \hat{\rho}\lambda_i} \right] = 0, \quad (2.31)$$

where $\lambda_1, \dots, \lambda_n$ are the eigenvalues of \mathbf{W} and

$$h_1 = \mathbf{y}^\top \left[\mathbf{W}^\top \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n} \right) \mathbf{W} \right] \mathbf{y};$$

$$h_2 = \mathbf{y}^\top \left[\mathbf{W}^\top \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n} \right) \right] \mathbf{y};$$

$$h_3 = \mathbf{y}^\top \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n} \right) \mathbf{y}.$$

Proof. The proof of the theorem follows from equations (2.17), (2.18), (2.19), and (2.30). □

2.4.4 An implementation of Ord's method in R

When \mathbf{W} is a row-standardized matrix, Lemma 2.4 in Section 2.3.4 guarantees that \mathbf{W} has no eigenvalues with absolute value in the interval $(1, \infty)$. As a result, if $|\rho| < 1$, then $1 - \rho\lambda_i \neq 0$ for $i = 1, \dots, n$. Because, if $1 - \rho\lambda_i = 0$ for some i , then $|\rho| = |\frac{1}{\lambda_i}| \geq 1$, a contradiction.

This means, in equation (2.31), if we search for a solution for ρ such that $|\rho| < 1$, we would not have any problem with the n denominators $1 - \rho\lambda_1, 1 - \rho\lambda_2, \dots, 1 - \rho\lambda_n$ (since none of them would be equal to zero). In other words, when ρ is real, the function on the LHS of equation (2.31) has no vertical asymptotes for all values of ρ in the interval $(-1, 1)$.

We should also consider the case when the eigenvalues of weight matrix \mathbf{W} are complex. When an eigenvalue λ_i is complex, say $\lambda = a + b\mathbf{i}$, where $\mathbf{i} = \sqrt{-1}$, then the conjugate $\bar{\lambda}_i = a - b\mathbf{i}$ is also an eigenvalue. In such a case, for ρ real, the part of the LHS of equation (2.31) that involves λ_i and $\bar{\lambda}_i$ becomes:

$$\begin{aligned} \frac{\lambda_i}{1 - \rho\lambda_i} + \frac{\bar{\lambda}_i}{1 - \rho\bar{\lambda}_i} &= \frac{a + b\mathbf{i}}{1 - \rho(a + b\mathbf{i})} + \frac{a - b\mathbf{i}}{1 - \rho(a - b\mathbf{i})} \\ &= \frac{a + b\mathbf{i}}{(1 - \rho a) - \rho b\mathbf{i}} + \frac{a - b\mathbf{i}}{(1 - \rho a) + \rho b\mathbf{i}} = \frac{2a - 2\rho(a^2 + b^2)}{(1 - \rho a)^2 + \rho^2 b^2}. \end{aligned}$$

This means that the LHS of equation (2.31) is real even if \mathbf{W} has some complex eigenvalues.

In the **R** program that appears in the appendix of this chapter (Section 2.6), we implement Ord's eigenvalue method for the model (2.7) in Section 2.2.1 with independent normal errors. The program relies on Theorems 2.1 and 2.2. Also, the results from the program are valid even when \mathbf{W} has complex eigenvalues. The matrix \mathbf{W} , however, has to be row-standardized.

Note that the program does not apply to \mathbf{W} 's that can be found in either Section 2.3.2 or 2.4.2, which are related to the conditional model (2.2). In such a case, \mathbf{W} may have a row of zeros, or more generally, the term $\sum_{i=1}^n \frac{\lambda_i}{1 - \rho\lambda_i}$ is zero for all ρ with $|\rho| < 1$. Instead, in this case, we may use Theorem 2.1 and Lemma 2.5.

2.4.5 Use of the **R** package `spdep`

In this section, we explain how to use the function `lagsarlm` of the **R** package `spdep`. According to the online documentation of the function, it “provides maximum likelihood estimation of spatial simultaneous autoregressive lag and spatial Durbin (mixed)

models of the form”:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{Y} + \boldsymbol{\varepsilon}.$$

In our case $\mathbf{X} = (1, \dots, 1)^\top$ and $\boldsymbol{\beta} = \boldsymbol{\alpha} = (\alpha, \dots, \alpha)$. The **R** code below explains how one can use this function:

```
library(spdep)
lw=mat2listw(w, style="M")
data=data.frame(y)
autore=lagsarlm(y~1, data=data, lw, method="eigen", tol.solve=1.0e-30)
autore
autore$s2
```

In the next section, we use two hypothetical examples with row-standardized weight matrices: one with a \mathbf{W} that has real eigenvalues and another one with a \mathbf{W} that has complex eigenvalues. The purpose of these examples is to illustrate the program described in Section 2.4.4 (that appears in the appendix, i.e., Section 2.6) and to compare the results with those obtained by using the function **lagsarlm** described above.

2.4.6 Some hypothetical examples

In this section, using two examples, we test the **R** program that is discussed in Section 2.4.4 (and appears in the appendix of this chapter, i.e., Section 2.6).

Example 1. We first use a 9×9 row-standardized weight matrix \mathbf{W} from a rook connectivity structure. We thus row-standardize the weight matrix in equation (2.9):

$$\mathbf{W} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \begin{pmatrix} 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0.333 & 0 & 0.333 & 0 & 0.333 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0.333 & 0 & 0 & 0 & 0.333 & 0 & 0.333 & 0 & 0 \\ 0 & 0.25 & 0 & 0.25 & 0 & 0.25 & 0 & 0.25 & 0 \\ 0 & 0 & 0.333 & 0 & 0.333 & 0 & 0 & 0 & 0.333 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.333 & 0 & 0.333 & 0 & 0.333 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 \end{pmatrix} \end{matrix}$$

We also generate a random sample Y_1, Y_2, \dots, Y_9 of size $n = 9$ from the Uniform(10, 100) distribution. Under these assumptions about \mathbf{W} and Y_1, \dots, Y_n , we implement Ord's eigenvalue method to estimate the parameters α , σ^2 , and ρ . The **R** program for this hypothetical example appears in the appendix of this chapter (Section 2.6).

Partial R output for example 1

```
> y= matrix(runif(n,10,100))
>   y
      [,1]
[1,] 59.93222
[2,] 26.69796
[3,] 26.75263
[4,] 57.67429
[5,] 55.32099
[6,] 44.72922
[7,] 45.65528
[8,] 14.19529
[9,] 37.80421
>
> ordeigen(n, y, w)
$eigen
      [,1]
[1,] -1.000000e+00
[2,]  1.000000e+00
[3,]  5.773503e-01
[4,] -5.773503e-01
[5,] -5.773503e-01
[6,]  5.773503e-01
```

```

[7,] 4.338382e-18
[8,] -3.604462e-18
[9,] 0.000000e+00

$rho
[1] 0.01018761

> alpha(n, y, w, ordeigen(n,y,w)$rho)
      [,1]
[1,] 40.55892
> sigma2(n, y, w, ordeigen(n,y,w)$rho)
      [,1]
[1,] 223.6851
> loglik(n, y, w, ordeigen(n,y,w)$rho)
      [,1]
[1,] -37.1167
>
> library(spdep)
> lw=mat2listw(w, style="M")
> data=data.frame(y)
> autore=lagsarlm(y~1, data=data, lw, method="eigen", tol.solve=1.0e-30)
> autore

Call:
lagsarlm(formula = y ~ 1, data = data, listw = lw, method = "eigen",
          tol.solve = 1e-30)
Type: lag

Coefficients:
      rho (Intercept)
0.01016997 40.55963401

Log likelihood: -37.1167
>      autore$s2
[1] 223.6851

```

From the above output, we get the following results:

$$\hat{\rho} = 0.01018761, \hat{\alpha} = 40.55892, \hat{\sigma}^2 = 223.6851, \text{ and } \log\text{-likelihood} = \ell = -37.1167.$$

Up to 4 decimal points, the results of our **R** program agree with the results obtained using the built-in **R** function **lagsarlm** from the package **spdep**.

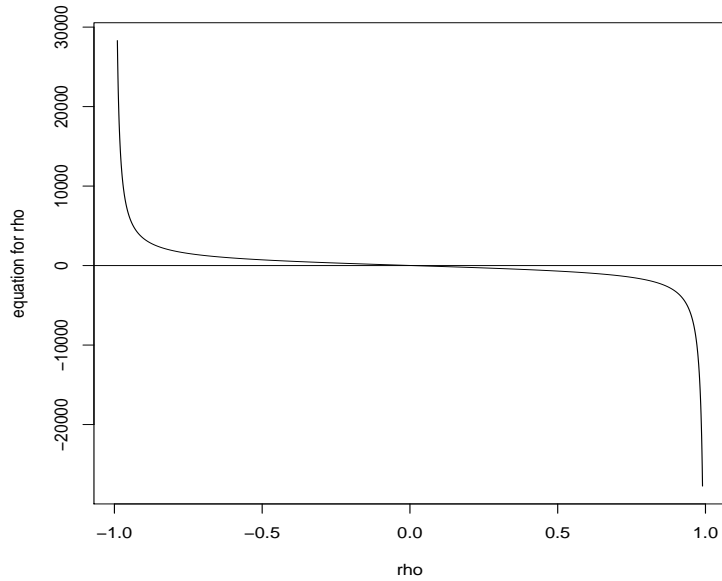


Figure 2.3: Plot of eq. (2.31) from Theorem 2.2, as a function of ρ , for Example 1.

In Figure 2.3, for this example, we graph the left-hand side of equation (2.31) as a function of ρ to see where the MLE of ρ , i.e., $\hat{\rho}$, is located in the interval $(-1, 1)$.

Example 2. We next use the following 9×9 row-standardized weight matrix \mathbf{W} that has complex eigenvalues:

$$\mathbf{w} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

Clearly, the above matrix is a permutation of the rows of the identity matrix \mathbf{I}_9 . As in the previous example, we generate a random sample Y_1, Y_2, \dots, Y_9 of size $n = 9$

from the Uniform(10, 100) distribution, and we implement Ord's eigenvalue method to estimate the parameters α , σ^2 , and ρ . The **R** program for this hypothetical example appears again in the appendix of this chapter (Section 2.6).

Partial R output for example 2

```
> y= matrix(runif(n,10,100))
>   y
      [,1]
[1,] 99.51419
[2,] 13.73728
[3,] 33.76472
[4,] 72.56716
[5,] 56.00784
[6,] 45.92709
[7,] 47.57752
[8,] 98.88120
[9,] 97.98247
>
> ordeigen(n, y, w)
$eigen
      [,1]
[1,] -0.5+0.8660254i
[2,] -0.5-0.8660254i
[3,]  1.0+0.0000000i
[4,] -1.0+0.0000000i
[5,]  1.0+0.0000000i
[6,] -1.0+0.0000000i
[7,]  1.0+0.0000000i
[8,] -1.0+0.0000000i
[9,]  1.0+0.0000000i

$rho
[1] 0.1120304
>
> alpha(n, y, w, ordeigen(n,y,w)$rho)
      [,1]
[1,] 55.83942
> sigma2(n, y, w, ordeigen(n,y,w)$rho)
      [,1]
[1,] 840.2854
> loglik(n, y, w, ordeigen(n,y,w)$rho)
```

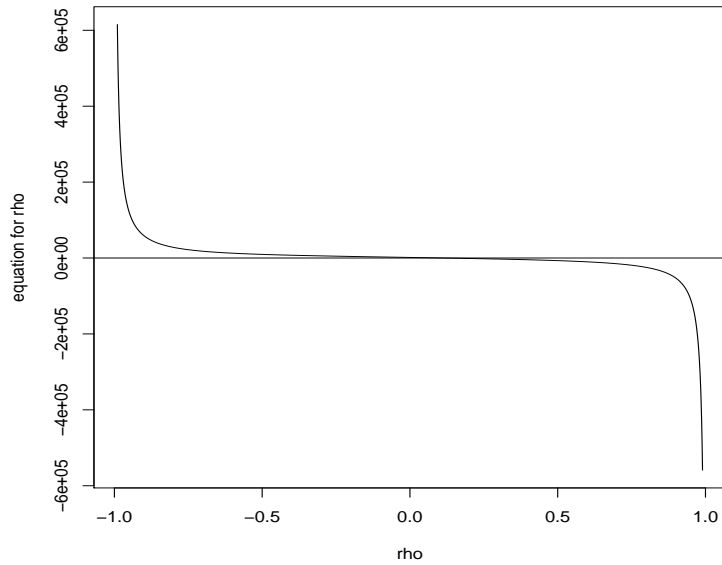


Figure 2.4: Plot of eq. (2.31) from Theorem 2.2, as a function of ρ , in Example 2.

```

      [,1]
[1,] -43.11158
>
> library(spdep)
> lw=mat2listw(w, style="M")
> data=data.frame(y)
> autore=lagsarlm(y~1, data=data, lw, method="eigen", tol.solve=1.0e-30)
> autore

```

```

Call:
lagsarlm(formula = y ~ 1, data = data, listw = lw, method = "eigen",
          tol.solve = 1e-30)
Type: lag

```

```

Coefficients:
      rho (Intercept)
 0.112031  55.839387

```

```

Log likelihood: -43.11158
> autore$s2
[1] 840.2853

```

From the above output, we get the following results about the second example:

$$\hat{\rho} = 0.1120304, \hat{\alpha} = 55.83942, \hat{\sigma}^2 = 840.2854, \text{ and } \log\text{-likelihood} = \ell = -43.11158.$$

Again, up to 4 decimal points, our results agree with those obtained using the built-in **R** function `lagsarlm` from the package `spdep`. Also, in Figure 2.4, we graph the left-hand side of equation (2.31) as a function of ρ to see where the MLE of ρ , i.e., $\hat{\rho}$, is located in the interval $(-1, 1)$.

2.5 A mixed regressive-autoregressive spatial model

If we consider variations in the mean level of \mathbf{Y} , then we get the following *mixed regressive-autoregressive model* that generalizes equation (2.7):

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{Y} + \boldsymbol{\varepsilon}. \quad (2.32)$$

Here, \mathbf{X} is an $n \times (p + 1)$ design matrix with all ones in its first column, $\boldsymbol{\beta}$ is the vector of coefficients with dimension $(p + 1) \times 1$, \mathbf{W} is an $n \times n$ weight matrix, and $\mathbf{Y} = (y_1, y_2, \dots, y_n)^\top$ and $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)^\top$ are $n \times 1$ matrices.

In general, the parameter ρ is unknown and we use the maximum likelihood procedure to estimate it (see also Section 2.4.5). Theorems 2.3 and 2.4 below give the ML equations for $\boldsymbol{\beta}$, σ^2 , and ρ . Equations (2.33) and (2.34) in Theorem 2.3 appear also in Section 4 of Ord [9]. The proofs of Theorem 2.3 and Theorem 2.4 are similar to the proofs of Theorem 2.1 and Theorem 2.2, respectively, and hence we omit them.

Theorem 2.3. *Assume $\boldsymbol{\varepsilon} \sim \text{MVN}(\mathbf{0}, \sigma^2\mathbf{I})$ and $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{Y} + \boldsymbol{\varepsilon}$. If the MLEs of*

the parameters $\boldsymbol{\beta}$, σ^2 , and ρ exist, then they satisfy the following equations:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top (\mathbf{I} - \hat{\rho} \mathbf{W}) \mathbf{y}, \quad (2.33)$$

$$\hat{\sigma}^2 = \frac{1}{n} \mathbf{y}^\top (\mathbf{I} - \hat{\rho} \mathbf{W})^\top (\mathbf{I} - \mathbf{H}) (\mathbf{I} - \hat{\rho} \mathbf{W}) \mathbf{y}, \quad (2.34)$$

$$\text{and } -\hat{\sigma}^2 \text{Tr}[(\mathbf{I} - \hat{\rho} \mathbf{W})^{-1} \mathbf{W}] + \mathbf{y}^\top \mathbf{W} \mathbf{y} - \hat{\rho} \mathbf{y}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y} - \hat{\boldsymbol{\beta}}^\top \mathbf{X}^\top \mathbf{W} \mathbf{y} = 0. \quad (2.35)$$

Here, $\mathbf{I} - \mathbf{H}$ is symmetric and idempotent, where $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$ is the usual hat matrix in linear regression.

Theorem 2.4. *The MLE of the parameter ρ satisfies the following equation:*

$$-\frac{\hat{\rho}^2}{n} \left[h_1 \sum_{i=1}^n \frac{\lambda_i}{1 - \hat{\rho} \lambda_i} \right] + \hat{\rho} \left[-h_1 + \frac{2h_2}{n} \sum_{i=1}^n \frac{\lambda_i}{1 - \hat{\rho} \lambda_i} \right] + \left[h_2 - \frac{h_3}{n} \sum_{i=1}^n \frac{\lambda_i}{1 - \hat{\rho} \lambda_i} \right] = 0, \quad (2.36)$$

where $\lambda_1, \dots, \lambda_n$ are the eigenvalues of \mathbf{W} and

$$h_1 = \mathbf{y}^\top [\mathbf{W}^\top (\mathbf{I} - \mathbf{H}) \mathbf{W}] \mathbf{y};$$

$$h_2 = \mathbf{y}^\top [\mathbf{W}^\top (\mathbf{I} - \mathbf{H})] \mathbf{y};$$

$$h_3 = \mathbf{y}^\top (\mathbf{I} - \mathbf{H}) \mathbf{y}.$$

Remark 2.4. For the mixed regressive-autoregressive model, the log-likelihood equation has the following form:

$$\begin{aligned} \ell(\boldsymbol{\beta}, \sigma^2, \rho; \mathbf{y}) &= \ln(\det(\mathbf{I} - \rho \mathbf{W})) - \frac{n}{2} \ln(2\pi\sigma^2) \\ &\quad - \frac{1}{2\sigma^2} [\mathbf{y}^\top (\mathbf{I} - \rho \mathbf{W})^\top (\mathbf{I} - \rho \mathbf{W}) \mathbf{y} - 2\boldsymbol{\beta}^\top \mathbf{X}^\top (\mathbf{I} - \rho \mathbf{W}) \mathbf{y} + \boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{X} \boldsymbol{\beta}]. \end{aligned}$$

2.6 Appendix to Chapter 2

In this section, we give an **R** program for implementing Ord's eigenvalue method for the model (2.7) in Section 2.2.1 with independent normal errors. We also give the **R** programs for running the two hypothetical examples in Section 2.4.6.

(a) R program for implementing Ord's eigenvalue method

```
## The function "ordeigen" prints the eigenvalues of W, finds an
## estimate for the MLE of rho, and graphs the score function for rho.

ordeigen=function(n,y,w) {
  uvec=matrix(rep(1,n))
  I=diag(n)
  h1=t(y)%*%t(w)%*(I-(uvec%*%t(uvec)/n))%*%w%*%y
  h2=t(y)%*%t(w)%*(I-(uvec%*%t(uvec)/n))%*%y
  h3=t(y)%*(I-(uvec%*%t(uvec)/n))%*%y

  ## The following command prints the eigenvalues of W.

  e=matrix(eigen(w)$values)

  ## The MLE of rho is a root of the function below. Notice that
  ## we use the R function Re to extract the real part of a
  ## complex number whose imaginary part is close to zero:

  mle=function(x){
    a1= -(x^2/n)*h1*sum(e/(1-x*e))
    a2= x*(-h1+(2*h2/n)*sum(e/(1-x*e)))
    a3= (h2-(h3/n)*sum(e/(1-x*e)))
    aa= Re(a1+a2+a3)
    return(aa)
  }

  ## These commands are needed for graphing the above function mle()
  ## as a function of rho:

  tt = seq(-0.99,0.99,.001)
  yy=tt
  for (i in 1:length(tt)){yy[i] = mle(tt[i])}
```

```

plot(tt, yy, type="l", xlab="rho", ylab="equation for rho")
abline(h=0)

## Prints the estimate of the MLE of rho and the eigenvalues of W.

root<- uniroot(mle, c(-0.9999,0.9999))
list(eigen=e, rho=root$root)
}
#####

## The function "alpha" prints the estimate of the MLE of the
## intercept term alpha.

alpha=function(n,y,w,v) {
  uvec=matrix(rep(1,n))
  I=diag(n)
  a = 1/n*t(uvec)%*(I-v*w)%*y
  return(a)
}
#####

## The function "sigma2" prints the estimate of the error variance,
## i.e., the estimate of the sigma-squared term.

sigma2= function(n,y,w,v){
  uvec=matrix(rep(1,n))
  I=diag(n)
  b=1/n*t(y)%*(I-v*w)%*(I-uvec)%*t(uvec)/n)%*(I-v*w)%*y
  return(b)
}
#####

## The function "loglik" calculates the value of the
## maximized log-likelihood.

loglik= function(n,y,w,v) {
  uvec=matrix(rep(1,n))
  I=diag(n)
  a1 = log(det(I-v*w))-n/2*log(2*pi*sigma2(n,y,w,v))
  a2 = -1/(2*sigma2(n,y,w,v))
  a3 = t(y)%*(I-v*w)%*(I-v*w)%*y
  a4 = - 2*alpha(n,y,w,v)*t(uvec)%*(I-v*w)%*y + n*(alpha(n,y,w,v))^2
  aa = a1+a2*(a3+a4)
}

```

```

    return(aa)
}

```

(b) R code for Example 1 in Section 2.4.6

```

## This is hypothetical example 1: W has real eigenvalues.
## Sample size
    n=9

## Data generated randomly from a uniform distribution.
    y= matrix(runif(n,10,100))
    y

## Row-standardized weight matrix W for a 3 x 3 grid with
## rook connectivity structure (that is, W is 9 x 9).
    w= matrix(c(
    0, 1/2, 0, 1/2, 0, 0, 0, 0, 0,
    1/3, 0, 1/3, 0, 1/3, 0, 0, 0, 0,
    0, 1/2, 0, 0, 0, 1/2, 0, 0, 0,
    1/3, 0, 0, 0, 1/3, 0, 1/3, 0, 0,
    0, 1/4, 0, 1/4, 0, 1/4, 0, 1/4, 0,
    0, 0, 1/3, 0, 1/3, 0, 0, 0, 1/3,
    0, 0, 0, 1/2, 0, 0, 0, 1/2, 0,
    0, 0, 0, 0, 1/3, 0, 1/3, 0, 1/3,
    0, 0, 0, 0, 0, 1/2, 0, 1/2, 0),
    nrow=9,ncol=9,byrow=TRUE)

## For the hypothetical example, we calculate the estimates
## of the MLEs of rho, alpha, and sigma-squared along with
## the value of the maximized log-likelihood. In addition,
## we plot the graph of the equation for rho and print
## the eigenvalues of W.

    ordeigen(n, y, w)
    alpha(n, y, w, ordeigen(n,y,w)$rho)
    sigma2(n, y, w, ordeigen(n,y,w)$rho)
    loglik(n, y, w, ordeigen(n,y,w)$rho)

```

(c) R code for Example 2 in Section 2.4.6

```

## Hypothetical example 2: matrix W has complex eigenvalues.

## Sample size

```

```

n=9

## Data generated randomly from a uniform distribution.
y= matrix(runif(n,10,100))
y
## Row-standardized weight matrix W of dimension 9 x 9
## with complex eigenvalues
w= matrix(c(
  0,1,0,0,0,0,0,0,0,
  0,0,1,0,0,0,0,0,0,
  1,0,0,0,0,0,0,0,0,
  0,0,0,0,1,0,0,0,0,
  0,0,0,1,0,0,0,0,0,
  0,0,0,0,0,0,1,0,0,
  0,0,0,0,0,1,0,0,0,
  0,0,0,0,0,0,0,0,1,
  0,0,0,0,0,0,0,0,1,0
),
nrow=9,ncol=9,byrow=TRUE)

## For the hypothetical example, we calculate the estimates
## of the MLEs of rho, alpha, and sigma-squared along with
## the value of the maximized log-likelihood. In addition,
## we plot the graph of the equation of rho and print
## the eigenvalues of W.

ordeigen(n, y, w)
alpha(n, y, w, ordeigen(n,y,w)$rho)
sigma2(n, y, w, ordeigen(n,y,w)$rho)
loglik(n, y, w, ordeigen(n,y,w)$rho)

```


CHAPTER 3

IMPLEMENTATION OF ORD'S METHOD TO THE COLUMBUS, OHIO, DATA SET

In Chapter 2 of this thesis, we developed Ord's [9] eigenvalue method for estimating the parameters of a mixed regressive-autoregressive model, but we only applied it to some simple hypothetical data. In this chapter, we implement Ord's method to real world data.

3.1 The Columbus, Ohio, spatial data set

In this section, we describe the Columbus, Ohio, crime data from 1980. This data set originally appeared in Table 12.1 of Anselin [1, p. 187–190]. The data set concerns 49 contiguous neighborhoods in Columbus, Ohio. The neighborhood list corresponds to a list of *census tracts* created by the US Census Bureau. Essentially, census tracts are small and stable subdivisions of a county or an equivalent entity of a USA state, and they usually have between 1200 and 8000 residents [15].

From the data set, we use three variables: CRIME, HOUSE, and INC. For each neighborhood, the CRIME variable represents the total number of burglaries and vehicle thefts per 1000 households, while the HOUSE and INC variables represent the housing value and household income in thousand dollars, respectively.

From the description of the data set, it is not clear whether the HOUSE and INC variables represent the total or the average housing value and household income, respectively, for each neighborhood. But it seems more logical to assume that these variables are considering the average housing value and the average household income for each neighborhood, respectively.

In the dataset, the X and Y variables represent the coordinates of the neighborhood. It is not stated in the description of the data what exactly (X, Y) are for each neighborhood, but most probably they are the relative coordinates (with respect to some origin) of the geographical center of each neighborhood. We will use these coordinates to construct different weight matrices for our spatial regression model as described in Section 2.3.3 of the thesis.

In the **R** package **spdep**, the old name of the data set is **oldcol** while the current name of the data frame is **COL.OLD** [12]. The data set has 49 rows and 22 columns that contain information about the 49 neighborhoods in Columbus, OH.

The data set also includes the **COL.nb** object, which includes the neighbourhood list. From this object, we may find out the bordering neighbourhoods of each of the 49 neighbourhood. This allows us to create a contiguity weight matrix for these 49 Columbus neighbourhoods.

Using different weight matrices, we implement Ord's eigenvalue method to estimate the parameters of a mixed regressive-autoregressive model between the response variable CRIME and the predictors HOVAL (housing value in \$1000) and INC (household income in \$1000).

3.2 An R program for estimating Ord's model

In the following R program, we implement Ord's eigenvalue method to the mixed regressive-autoregressive model

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{Y} + \boldsymbol{\varepsilon}, \quad \text{where } \boldsymbol{\varepsilon} \sim \text{MVN}(\mathbf{0}, \sigma^2\mathbf{I}). \quad (3.1)$$

The program relies on Theorems 2.3 and 2.4, and it is more general than the R program in Section 2.4.4 because it takes into account the values of the covariates. Note, also, that the program is valid even when \mathbf{W} has complex eigenvalues. The formula for the corresponding log-likelihood (that we implement in the program) appears in Remark 2.4.

```
## The function "ordeigen" prints the eigenvalues of W, finds an
## estimate for the MLE of rho, and graphs the equation for rho.
```

```
ordeigen=function(n,y,x,w) {
  I=diag(n)
  H= x%%solve(t(x)%%x)%%t(x)
  h1=(t(y)%%t(w)%%(I-H)%%w%%y)
  h2=(t(y)%%t(w)%%(I-H)%%y)
  h3=(t(y)%%(I-H)%%y)

  ## printing the eigenvalues of W:
  e=matrix(eigen(w)$values)

  mle=function(m){
    a1= -(m^2/n)*h1*sum(e/(1-m*e))
    a2= m*(-h1+(2*h2/n)*sum(e/(1-m*e)))
    a3= (h2-(h3/n)*sum(e/(1-m*e)))
    aa= Re(a1+a2+a3)
    return(aa)
  }

  ## graphing the function mle() as a function of rho:
  tt = seq(-0.99,0.99,.001)
  yy=tt
  for (i in 1:length(tt)){yy[i] = mle(tt[i])}
```

```

plot(tt, yy, type="l", xlab="rho", ylab="equation for rho")
abline(h=0)

## printing the estimate of the MLE of rho:
root<- uniroot(mle, c(-0.9999,0.9999))
list(eigen=e, rho=root$root)
}
#####

## The function "beta" prints the MLEs of the coefficients.

beta=function(n,y,x,w,v) {
  I=diag(n)
  a=solve(t(x)%*%x)%*%t(x)%*(I-v*w)%*%y
  return(a)
}
#####

## The function "sigma2" prints the estimate of the error variance,
## i.e., the estimate of the sigma-squared term.

sigma2= function(n,y,x,w,v){
  I=diag(n)
  H= x%*%solve(t(x)%*%x)%*%t(x)
  b=1/n*t(y)%*%t(I-v*w)%*(I-H)%*(I-v*w)%*%y
  return(b)
}
#####

## The function "loglik" calculates the value of the
## maximized log-likelihood.

loglik= function(n,y,x,w,v) {
  I=diag(n)
  a1 = log(det(I-v*w))-n/2*log(2*pi*sigma2(n,y,x,w,v))
  a2 = -1/(2*sigma2(n,y,x,w,v))
  a3 = t(y)%*%t(I-v*w)%*(I-v*w)%*%y
  a4 = - 2*t(beta(n,y,x,w,v))%*%t(x)%*(I-v*w)%*%y
  a5 = t(beta(n,y,x,w,v))%*%t(x)%*%x%*%beta(n,y,x,w,v)
  aa = a1+a2*(a3+a4+a5)
  return(aa)
}

```

3.3 Spatial regression using natural neighborhood contiguity

In this section, we use the modified **R** program from Section 3.2 of this thesis on the Columbus, OH, crime data from 1980 in order to estimate the parameters of the model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{Y} + \boldsymbol{\varepsilon}$ (where $\boldsymbol{\varepsilon} \sim \text{MVN}(\mathbf{0}, \sigma^2\mathbf{I})$). We use a weight matrix **W** that can be obtained using the natural contiguity of the neighborhoods. For example, neighborhood 1 of Columbus is contiguous to the neighborhoods 2, 5, and 6, while neighborhood 2 is contiguous to the neighborhoods 1, 3, 6, and 7, and so on. A complete list of neighborhood contiguity can be found in Table 12.2 of Anselin [1, p. 190].

For this data set, **X** is a 49×3 design matrix with all ones in its first column, $\boldsymbol{\beta}$ is a 3×1 vector of coefficients, **W** is a 49×49 weight matrix, and **Y** and $\boldsymbol{\varepsilon}$ are 49×1 matrices.

The **R** commands needed to read the Columbus data, create the **W** matrix, and apply the program from Section 3.2 for the estimation of the parameters, along with some necessary explanations, appear in the appendix of this chapter (Section 3.6). Below we give some of the output of these **R** commands. At the beginning of the output, we have the 49 eigenvalues λ_i of **W**.

Partial output for the **R** program in Section 3.6

```
> ordeigen(n, y, x, w)
$eigen
      [,1]
[1,] 1.000000e+00
[2,] 9.683166e-01
[3,] 9.392775e-01
```

[4,] 8.783334e-01
[5,] 8.400875e-01
[6,] 7.616861e-01
[7,] 6.930989e-01
[8,] -6.509666e-01
[9,] -6.017333e-01
[10,] 5.804820e-01
[11,] -5.637252e-01
[12,] 5.523933e-01
[13,] 5.138516e-01
[14,] -5.060598e-01
[15,] -5.000000e-01
[16,] -4.919854e-01
[17,] -4.864307e-01
[18,] -4.612340e-01
[19,] -4.468910e-01
[20,] 4.298413e-01
[21,] -4.241157e-01
[22,] -4.041236e-01
[23,] -3.944806e-01
[24,] -3.777849e-01
[25,] -3.454928e-01
[26,] -3.435605e-01
[27,] 3.416926e-01
[28,] -3.211139e-01
[29,] 3.040721e-01
[30,] -2.930730e-01
[31,] -2.896441e-01
[32,] 2.751662e-01
[33,] -2.622402e-01
[34,] -2.500000e-01
[35,] -2.283925e-01
[36,] -2.211569e-01
[37,] 1.974175e-01
[38,] -1.878548e-01
[39,] -1.675348e-01
[40,] 1.564255e-01
[41,] -1.444632e-01
[42,] 1.097787e-01
[43,] -9.236810e-02
[44,] -8.041567e-02
[45,] 6.003855e-02
[46,] -5.978906e-02
[47,] -3.330931e-02
[48,] 2.798021e-02

```

[49,] 3.722489e-18

$rho
[1] 0.4310229

> beta(n, y, x, w, ordeigen(n,y,x,w)$rho)
      [,1]
[1,] 45.0792665
[2,] -1.0316161
[3,] -0.2659263
> sigma2(n, y, x, w, ordeigen(n,y,x,w)$rho)
      [,1]
[1,] 95.4945
> loglik(n, y, x, w, ordeigen(n,y,x,w)$rho)
      [,1]
[1,] -182.3904
>

> library(spdep)
> lw=mat2listw(w, style="M")
> data(oldcol)
> data2=data.frame(COL.OLD)
> autore=lagsarlm(CRIME ~ INC + HOVAL, data=data2,lw, method="eigen",
  tol.solve=1.0e-30)
> autore

Call:
lagsarlm(formula = CRIME ~ INC + HOVAL, data = data2, listw = lw,
  method = "eigen", tol.solve = 1e-30)
Type: lag

Coefficients:
      rho (Intercept)          INC          HOVAL
  0.4310232  45.0792499 -1.0316157 -0.2659263

Log likelihood: -182.3904
> autore$s2
[1] 95.4945

> summary(autore)

Call:lagsarlm(formula = CRIME ~ INC + HOVAL, data = data2, listw = lw,
  method = "eigen", tol.solve = 1e-30)

Residuals:

```

	Min	1Q	Median	3Q	Max
	-37.68585	-5.35636	0.05421	6.02013	23.20555

Type: lag

Coefficients: (asymptotic standard errors)

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	45.079250	7.177347	6.2808	3.369e-10
INC	-1.031616	0.305143	-3.3808	0.0007229
HOVAL	-0.265926	0.088499	-3.0049	0.0026570

Rho: 0.43102, LR test value: 9.9736, p-value: 0.001588

Asymptotic standard error: 0.11768

z-value: 3.6626, p-value: 0.00024962

Wald statistic: 13.415, p-value: 0.00024962

Log likelihood: -182.3904 for lag model

ML residual variance (sigma squared): 95.494, (sigma: 9.7721)

Number of observations: 49

Number of parameters estimated: 5

AIC: 374.78, (AIC for lm: 382.75)

LM test for residual autocorrelation

test value: 0.31954, p-value: 0.57188

Using both our **R** program (that uses the theory of Chapter 2 of this thesis) and the built-in **R** function **lagsarlm**, we get the following results (rounded to 3 decimal digits):

$$\hat{\rho} = 0.431, \hat{\beta} = (45.079, -1.032, -0.266)^\top,$$

$$\hat{\sigma}^2 = 95.495, \text{ and } \log\text{-likelihood} = \ell = -182.390.$$

From the output of the built-in **R** function **lagsarlm**, we may also get some summary statistics for the fitted model. The above partial **R** output shows the following asymptotic standard errors of the estimates of the parameters (except for the asymptotic standard error of $\hat{\sigma}^2$):

$$\text{ASE}(\hat{\beta}_0) = 7.1773, \text{ ASE}(\hat{\beta}_1) = 0.3051, \text{ ASE}(\hat{\beta}_2) = 0.0885, \text{ and } \text{ASE}(\hat{\rho}) = 0.1177.$$

In this thesis, due to time constraints, we did not develop the theory of the calculation of the asymptotic standard errors of the estimates of the parameters in model (3.1).

3.4 Spatial regression using nearest neighbours

Following Pace and Barry [11], in this section, we use weight matrices \mathbf{W}_m that are based on the m^{th} nearest neighbour to each location i (as described in Section 2.3.3 of this thesis).

Since the census tracts are very close to each other and the data has been collected on a small geographical area, we can assume that the surface area in Columbus, Ohio, is approximately flat. Therefore, instead of using the spherical distance formula, we may use the Euclidean distance formula to find the distances between every two census tracts (neighborhoods).

Suppose (x_i, y_i) and (x_j, y_j) are the coordinates of (the geographical centers of) neighborhoods i and j , respectively. If we assume d_{ij} is the Euclidean distance between neighborhoods i and j , then $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

For each $m \geq 1$ and $1 \leq i, j \leq n$, we define the (i, j) -th element of matrix \mathbf{W}_m by

$$w_{ij}^{(m)} = \begin{cases} 1, & \text{if } d_{ij} \leq d_{(\max, i)}^{(m)} \quad \text{and} \quad i \neq j, \\ 0, & \text{otherwise.} \end{cases}$$

Here, $d_{(\max, i)}^{(m)} := m^{\text{th}}$ order statistic of the distances $d_{i,1}, d_{i,2}, \dots, d_{i,i-1}, d_{i,i+1}, \dots, d_{i,n}$.

(Recall that $d_{i,i} = 0$.)

The above condition on the (i, j) -th element of matrix \mathbf{W}_m yields a weight of 1 for the census tracts j (with $j \neq i$) that are within the m^{th} nearest neighbour of location

i , and 0 for census tracts j whose distance to location i is more than $d_{(max,i)}^m$. (The diagonal elements of \mathbf{W}_m are of course 0.)

Each of the matrices \mathbf{W}_m must be row-standardized (as described in Section 2.3.4):

$$\widetilde{\mathbf{W}}_m = \left(\widetilde{w}_{ij}^{(m)} \right)_{i,j=1}^n, \quad \text{where} \quad \widetilde{w}_{ij}^{(m)} = \frac{w_{ij}^{(m)}}{\sum_{\ell=1}^n w_{i\ell}^{(m)}} \quad \text{for} \quad 1 \leq i, j \leq n.$$

This is the weight matrix we need to use in the **R** program given in Section 3.2 of this thesis when we want to utilize m^{th} nearest neighbours to create a weight matrix.

For the Columbus, Ohio, crime data, we may use the coordinates of the neighborhoods (the X and Y variables in the data set) to calculate the distances from each location i to the neighborhood locations. For example, when $m = 3$, the distances from each location i to the third nearest neighbour are as follows:

$$d_{(max,1)}^{(3)} = 3.828, \quad d_{(max,2)}^{(3)} = 3.385, \quad d_{(max,3)}^{(3)} = 2.624, \dots, \quad d_{(max,49)}^{(3)} = 3.444.$$

An **R** program for creating the row-standardized weight matrix $\widetilde{\mathbf{W}}_m$, which is based on m^{th} nearest neighbor, appears in the appendix of this chapter (Section 3.6). Essentially, in the **R** program given in Section 3.2, instead of using the row-standardized weight matrix that is based on natural neighborhood contiguity (see Section 3.3), we may now use the matrices $\widetilde{\mathbf{W}}_m$.

Table 3.2 shows the “nearby” neighborhoods for each of the first five neighborhoods (out of the 49) in Columbus according to various criteria, that is, different weight matrices $\widetilde{\mathbf{W}}$. For each census tract $i \in \{1, 2, 3, 4, 5\}$, we list the “nearby” neighborhoods according to the $\widetilde{\mathbf{W}}$ for natural contiguity, $\widetilde{\mathbf{W}}_1$, $\widetilde{\mathbf{W}}_2$, and $\widetilde{\mathbf{W}}_3$.

Neighborhood	$\widetilde{\mathbf{W}}$ for contiguity	$\widetilde{\mathbf{W}}$ with $m = 1$	$\widetilde{\mathbf{W}}$ with $m = 2$	$\widetilde{\mathbf{W}}$ with $m = 3$
1	2, 5, 6	2	2, 5	2, 3, 5
2	1, 3, 6, 7	3	1, 3	1, 3, 6
3	2, 4, 7, 37, 38, 39	2	2, 39	2, 37, 39
4	3, 37, 39, 40	39	3, 39	3, 39, 40
5	1, 6	6	1, 6	1, 2, 6

Table 3.2: “Nearby” neighborhoods of the first five neighborhoods for different $\widetilde{\mathbf{W}}$ ’s.

3.5 Results and Discussion

Table 3.4 shows the estimates of the parameters $\hat{\boldsymbol{\beta}}$, $\hat{\rho}$, and $\hat{\sigma}^2$, and the value of the maximized log-likelihood ℓ , for different row-standardized weight matrices.

Estimates	$\widetilde{\mathbf{W}}$ for contiguity	$\widetilde{\mathbf{W}}$ with $m = 1$	$\widetilde{\mathbf{W}}$ with $m = 2$	$\widetilde{\mathbf{W}}$ with $m = 3$
$\hat{\beta}_0$ (intercept)	45.080	50.539	41.192	42.040
$\hat{\beta}_1$ (INC)	-1.032	-1.158	-0.924	-0.941
$\hat{\beta}_2$ (HOVAL)	-0.266	-0.282	-0.260	-0.266
$\hat{\rho}$	0.431	0.306	0.453	0.443
$\hat{\sigma}^2$	95.495	93.644	81.885	86.261
ℓ	-182.390	-182.124	-179.407	-180.065
AIC	374.780	374.248	368.814	370.130

Table 3.4: Estimates of the parameters of model (3.1) for different $\widetilde{\mathbf{W}}$ ’s.

From Table 3.4, we see that the estimate of ρ for $\widetilde{\mathbf{W}}$ when $m = 1$ is 0.306, which is smaller in absolute value than the estimates of ρ for other $\widetilde{\mathbf{W}}$ ’s. Also, the estimates of the coefficients β_0 , β_1 , and β_2 for $\widetilde{\mathbf{W}}$ when $m = 2$ are close to the corresponding estimates for $\widetilde{\mathbf{W}}$ when $m = 3$ (and not far away from the corresponding estimates for the $\widetilde{\mathbf{W}}$ for natural contiguity). The smallest estimate of σ^2 occurs for the $\widetilde{\mathbf{W}}$ when $m = 2$ while the largest estimate of σ^2 occurs for the $\widetilde{\mathbf{W}}$ for natural contiguity. The estimate of the maximized log-likelihood ℓ for all $\widetilde{\mathbf{W}}$ ’s are very close to each other.

In order to compare the fitted models for different $\widetilde{\mathbf{W}}$'s, we may use the *Akaike Information Criterion* (AIC). The AIC is defined by the formula

$$\text{AIC} = 2(\text{number of estimated parameters}) - 2(\text{maximized log-likelihood value}).$$

From Table 3.4, we see that the smallest AIC value is 368.814, which occurs for $\widetilde{\mathbf{W}}$ when $m = 2$. Thus, based on the AIC value, it seems that best model is the one where $\widetilde{\mathbf{W}}$ is constructed using 2nd nearest neighbors for each neighborhood.

Table 3.6 shows the asymptotic standard errors of $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\beta}_2$, and $\hat{\rho}$ (but not for $\hat{\sigma}^2$) for different row-standardized weight matrices $\widetilde{\mathbf{W}}$'s. (Here, $n = 49$, which cannot be considered “small” for a linear regression with 5 estimated parameters.)

Standard Error	$\widetilde{\mathbf{W}}$ for contiguity	$\widetilde{\mathbf{W}}$ with $m = 1$	$\widetilde{\mathbf{W}}$ with $m = 2$	$\widetilde{\mathbf{W}}$ with $m = 3$
ASE($\hat{\beta}_0$)	7.1773	5.6959	6.2953	6.6462
ASE($\hat{\beta}_1$)	0.3051	0.2885	0.2803	0.2884
ASE($\hat{\beta}_2$)	0.0885	0.0875	0.0820	0.0841
ASE($\hat{\rho}$)	0.1177	0.0833	0.0976	0.1058

Table 3.6: Standard errors of the coefficients of model (3.1) for different $\widetilde{\mathbf{W}}$'s.

As we can see, the asymptotic standard error for $\hat{\beta}_0$ is the smallest for $\widetilde{\mathbf{W}}$ with $m = 1$. But for $\widetilde{\mathbf{W}}$ with $m = 2$, the asymptotic standard errors for $\hat{\beta}_1$, and $\hat{\beta}_2$ are the smallest compared to the corresponding estimates for other $\widetilde{\mathbf{W}}$'s. Also, the smallest asymptotic standard error for $\hat{\rho}$ is 0.0833, which occurs for $\widetilde{\mathbf{W}}$ with $m = 2$.

Remark 3.5. If we let r be the Pearson correlation coefficient between the explanatory variables **INC** and **HOVAL**, the *VIF* (Variance Inflation Factor) for each of the explanatory variables is $\frac{1}{1-r^2} = 1.333$, which is very small. This indicates that there exists no multi-collinearity among the explanatory variables.

3.6 Appendix to Chapter 3

In this appendix, we give **R** commands and some explanations that are needed in Sections 3.3 and 3.4 for the following tasks: (a) read the Columbus data into **R**, (b) create the **W** matrix indicating natural neighborhood contiguity among the 49 neighborhoods in Columbus, (c) create a weight matrix based on m^{th} nearest neighbors, and (d) apply the program from Section 3.2 to estimate the unknown model parameters. For task (b), in the **R** data frame **COL.OLD**, we may use the built-in neighborhood object **COL.nb** to find the weight matrix **W** that indicates natural contiguity.

R program for performing tasks (a), (b), and (d)

```
## We get the Columbus crime data, from year 1980, from the package
## "spdep". The old name of the data set is "oldcol" and the name
## of the current data frame is "COL.OLD".

library(spdep)
data(oldcol)
data2=data.frame(COL.OLD)
attach(data2)

## In order to use the R code from Section 3.2, we define
## the dependent variable "CRIME" as a "y" variable

y=data2$CRIME

## Sample size of the data (here n = 49):

n=length(y)

## Here, we create a 49 by 3 design matrix using all ones
## in the first column and the explanatory variables INC (income)
## and HOVAL (house value) in the other two columns.

x=cbind(rep(1,n),data2$INC,data2$HOVAL)
```

```

## To get the weight matrix W, we first create
## a weight "list" object from the neighborhood object COL.nb
## (a built-in object in R for this data set).
## The nb2listw function creates a weight list object
## from a binary neighborhood list object. Here COL.nb is
## a binary list object, in which regions are either listed
## as neighbours (1) or are absent (0).

lst=nb2listw(COL.nb, style="W")

## The following command creates a weight matrix from the "list" object.

w=listw2mat(lst)

## The following R code generates the estimates
## of the MLEs of rho, beta, and sigma-squared along with
## the value of the maximized log-likelihood.

ordeigen(n, y, x, w)
beta(n, y, x, w, ordeigen(n,y,x,w)$rho)
sigma2(n, y, x, w, ordeigen(n,y,x,w)$rho)
loglik(n, y, x, w, ordeigen(n,y,x,w)$rho)

## To check the results we get above, we use the built-in
## R function "lagsarlm" to fit a mixed regressive-autoregressive
## model and find the MLEs of rho, beta, and sigma-squared along with
## the value of the maximized log-likelihood.

library(spdep)
lw=mat2listw(w, style="M")
data(oldcol)
data2=data.frame(COL.OLD)
autore=lagsarlm(CRIME ~ INC + HOVAL, data=data2,lw, method="eigen",
               tol.solve=1.0e-30)

autore
autore$s2

## The following R code produce the estimates of standard errors
## for the Columbus, Ohio, crime data.

summary(autore)

```

R code for creating the weight matrix W_m (task (c))

The following R code produces weight matrices based on m^{th} nearest neighbours for the Columbus crime data set. First, we use the coordinates values from the variables X and Y to find the distances for each location i to its neighbors. Then the R code produces different weight matrices for $m = 1$, $m = 2$, $m = 3$, and so on.

```
## The following R code yields an n by n row-standardized weight matrix.
## In the function "weight", the argument "n" represents the sample size,
## "m" represents the order of the nearest neighbor, and
## "D" represents the n by n distance matrix, which can be obtained
## by using the X and Y coordinates of the n locations.
```

```
weight = function(n,m,D){
  mat_dis=matrix(c(D),nrow=n,ncol=n,byrow=TRUE)
  x=t(apply(mat_dis,1,sort))
  w=matrix(data=0, nrow=n, ncol=n)
  for(j in 1:n){
    for(i in 1:n)
      for(k in 1:n)
        {if (mat_dis[k,j]<=x[k,m+1] & k!=j) {w[k,j]=1}}
  }
  r_total=rowSums(w)
  return(w/r_total)
}
```

```
#####
```

```
## In order to find the distances between locations in
## Columbus, Ohio, we need the following R code:
```

```
library(spdep)
data(oldcol)
data2=data.frame(COL.OLD)
attach(data2)
X=data2$X
Y=data2$Y
location=as.matrix(cbind(X,Y))
distance <- dist(location)
D=as.matrix(distance)
```

```
## Here we find the weight matrix using third nearest neighbours.  
## For different values of m we may get different weight matrices.
```

```
m=3  
n= length(Y)  
w=weight(n,m,D)  
head(w)
```


CHAPTER 4

CONCLUSION AND FUTURE RESEARCH

In this thesis, we used a mixed regressive-autoregressive global model to describe the effect of interaction among neighboring locations on the response variable in the presence of several covariates (predictors). We also estimated the parameters of the model using Ord's [9] eigenvalue method with the help of the programming language **R**.

In spatial data analysis, it is crucial to choose the weights that indicate the contiguity of neighboring locations in a proper way. One important aspect of this thesis is the fact that we used different kinds of weight matrices based on the natural contiguity of each location and various notions of "nearest neighbor". This allowed us to compare the estimates of the parameters of the mixed regressive-autoregressive model for various choices of the weight matrix (see Table 3.4 in the previous chapter).

In order to make Ord's numerical method more reliable, we row-standardized each of those weight matrices by dividing each element of the matrix by the sum of the elements of each row. Since a row-standardized matrix with non-negative elements (sometimes known as Markov matrix) has (possibly complex) eigenvalues with modulus less than or equal to 1 (see Lemma 2.4 in this thesis), the estimate of

the parameter ρ in the model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{Y} + \boldsymbol{\varepsilon}$ lies in the interval $[-1, 1]$.

Our results, obtained by the implementation of Ord’s method in the programming language **R**, agree (up to 4 decimal points) with the results obtained by using the built-in **R** function **lagsarlm** from the **R** package **spdep**.

One of the major disadvantages of this thesis is that we did not develop the theory of the calculation of the asymptotic standard errors of the parameters in the model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{Y} + \boldsymbol{\varepsilon}$ with $E(\boldsymbol{\varepsilon}) = \mathbf{0}$ and $\text{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}_n$. This could have been achieved by using the Hessian matrix of the second partial derivatives of the log-likelihood function. In principle, these calculations are possible, but they are very tedious. Due to time constraints, we did not attempt these calculations.

For the Columbus, Ohio, data set, however, we did calculate the asymptotic standard errors for all estimated parameters (except for $\hat{\sigma}^2$) using the built-in **R** function **lagsarlm** of the **R** package **spdep**.

Although we have only considered the situation where the response variable regresses with itself, in real life statistical modelling of spatial data, a very common situation is to assume that there is a spatial autocorrelation among the error terms. In that case, we need to consider an extended model, called “Simultaneous Autoregressive (SAR)” error model, which can be specified as follows:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{U} \quad \text{and} \quad \mathbf{U} = \rho\mathbf{W}\mathbf{U} + \boldsymbol{\varepsilon}.$$

Here, $\mathbf{U} = (u_1, \dots, u_n)^\top$ is an $n \times 1$ vector of (correlated) disturbance terms and $\boldsymbol{\varepsilon} \sim \text{MVN}(\mathbf{0}, \sigma^2\mathbf{I}_n)$. We may replace \mathbf{U} to get the following model:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{Y} - \rho\mathbf{W}\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

The above model is basically a generalization of the *Durbin* model in the theory of time series. Again, in this case (as was the case with the spatial models examined in Chapter 2 of this thesis), the OLS estimates of the parameters are inconsistent, and we may use the maximum likelihood approach to estimate the parameters β , ρ , and σ^2 .

Due to time constraints, we did not consider the estimation of the simultaneous autoregressive (SAR) error model in this thesis. The parameters of the model can be estimated using the **R** function `errorsarlm` of the **R** package `spdep`.

Bibliography

- [1] Anselin, L. (1988), *Spatial Econometrics: Methods and Models*. Boston: Kluwer Academic Publishers.
- [2] Bartlett, M. S. (1971), Physical nearest neighbour models and non-linear time series, *Journal of Applied Probability*, **8**(2), 222–232.
- [3] Besag, J. E. (1972), Nearest neighbour systems and the auto-logistic model for binary data, *Journal of the Royal Statistical Society, Ser. B*, **34**(1), 75–83.
- [4] Besag, J. E. (1974), Spatial interaction and the statistical analysis of lattice systems, *Journal of the Royal Statistical Society, Ser. B*, **36**(2), 192–236.
- [5] Fotheringham, A. S., Brunson, C., and Charlton, M. (2002), *Geographical Weighted Regression*. Hoboken, New Jersey: John Wiley & Sons.
- [6] Hijmans, R. (2016), *Spatial Data Analysis and Modeling with R*. Available at <http://rspatial.org/>.
- [7] Matthews, K. (2017), *Markov matrices*, online notes in the Number Theory Web, available at: <http://www.numbertheory.org/courses/MP274/markov.pdf>, accessed August 31, 2018.

- [8] Marshall, A. W., and Olkin, I., (1979), *Inequalities: Theory of Majorization and its Applications*. New York: Academic Press.
- [9] Ord, K. (1975), Estimation methods for models of spatial interaction, *Journal of the American Statistical Association*, **70**(349), 120–126.
- [10] O’Sullivan, D. and Unwin, D. J. (2010), *Geographic information analysis*. Hoboken, New Jersey: John Wiley & Sons.
- [11] Pace, R. K. and Barry, R. (1997), Sparse spatial autoregression, *Statistics & Probability Letters*, **33**(3), 291–297.
- [12] R Package Documentation. *COL.OLD: Columbus OH spatial analysis data set - old numbering*. Available at <https://rdr.io/rforge/spdep/man/COL.OLD.html>.
- [13] Root, E. (2012), *Spatial Autocorrelation and Spatial Regression*, notes for a workshop at the University of Colorado, available at https://www.colorado.edu/ibs/crs/workshops/R_1-5-2011/root/SpatialReg.pptx, accessed June 19, 2018.
- [14] Sokal, R. R. and Oden, N. L. (1978), Spatial autocorrelation in biology 1. Methodology, *Biological Journal of the Linnean Society*, **10**(2), 199–228.
- [15] United States Census Bureau. *Geographic Terms and Concepts - Census Tract*. Available at https://www.census.gov/geo/reference/gtc/gtc_ct.html, accessed October 7, 2018.

- [16] Whittle, P. (1954), On Stationary Processes in the Plane, *Biometrika*, **41**, 434–449.
- [17] Whittle, P. (1963), Stochastic Processes in Several Dimensions, *Bulletin of the International Statistical Institute*, **40**, 974–994.

CURRICULUM VITAE

Graduate College

University of Nevada, Las Vegas

Sajib Mahmud Tonmoy

E-mail: tonmoysajib1989@gmail.com

Degrees:

Bachelor of Science in Statistics and Biostatistics, 2013

Master of Science in Statistics and Biostatistics, 2015

University of Dhaka, Dhaka, Bagladesh

Thesis Title: Estimation of the parameters in a spatial regressive-autoregressive model
using Ord's eigenvalue method

Thesis Examination Committee:

Chairperson, Petros Hadjicostas, Ph.D.

Committee Member, Hokwon A. Cho, Ph.D.

Committee Member, Kaushik Ghosh, Ph.D.

Graduate Faculty Representative, Ashok K. Singh, Ph.D.