2. Codifying the neighbourhood structure

Marie-Pierre de Bellefon, Vincent Loonis, Ronan Le Gleut
INSEE

Abstract

Once the data aggregation scale has been selected and an initial descriptive analysis using mapping tools has been made, the second step of a spatial analysis consists in defining an object's neighbourhood. Defining the neighbourhood is an essential step toward measuring the strength of the spatial relationships between objects, in other words the way in which neighbours influence each other. This makes it possible to compute spatial autocorrelation indices, implement spatial econometrics techniques, study the spatial distribution of observations, as well as perform spatial sampling or graph partitioning.

The challenge in this chapter is to succeed in defining neighbourhood relationships consistent with the actual spatial interactions between objects. This chapter introduces several concepts of neighbourhood, based on contiguity or distances between observations. The issue of the weight assigned to each neighbour is also addressed. Practical implementation is based on R packages *spdep*, *tripack*, *spsurvey* and *tsp*.

Prior reading of Chapter 1: "Descriptive spatial analysis" is recommended.

2.1 Defining neighbours

2.1.1 Characteristics of the relationships between spatial objects

Consider a surface $\Re$. This surface may be divided into n mutually exclusive zones. Two adjacent zones are separated by a common boundary. Boundaries can arise from spatial discontinuities (administrative or environmental boundaries). They may also rely on Voronoï polygons calculated from points of interest (see Chapter 1: “Descriptive spatial analysis”).

Box 2.1.1 — Mathematical definition of spatial relationships . Spatial relationships $\mathscr{B}$ are a subset of the Cartesian product $\mathbb{R}^2 \times \mathbb{R}^2 = \left\{(i,j) : i \in \mathbb{R}^2, j \in \mathbb{R}^2\right\}$ of couples (i,j) of spatial objects, *i.e.* all couples (i,j) such that i and j are both spatial objects identified by their geographical coordinates, and such that (i,j) is different from (j,i).
A spatial object cannot be linked to itself: $(i,i) \nsubseteq \mathscr{B}$. Moreover, if $(i,j) \subseteq \mathscr{B}$ and $(j,i) \subseteq \mathscr{B}$ for all couples of spatial objects, the spatial relationships are said to be *symmetrical* (Tiefelsdorf 1998).

Spatial relationships are multidirectional and multilateral. They are distinct, in this sense, from temporal relationships, which allow only sequential relationships along the past-present-future axis.

Figure 2.1 illustrates the codifying process of spatial relationships. This approach makes it possible to systematically transcribe the complexity of geographic space into a final set of data analysable by a computer.

First, the study zone is divided into mutually exclusive areas. Each area contains a reference point (often its centroid). Then, the spatial relationships can be specified by a neighbourhood graph connecting the areas considered to be neighbouring, or by a matrix containing the geographical coordinates of the reference points. The third step consists in coding the graph in a neighbourhood matrix, or transforming the geographic coordinates into a distance matrix.

The neighbourhood matrix measures how similar observations are. A value strictly greater than zero indicates that the observations are considered to be neighbouring. For example, in the case of the binary matrix shown in Figure 2.1:

$$w_{ij} = \begin{cases} 1 & if \quad i \quad and \quad j \quad are \quad spatially \quad linked \quad to \quad each \quad other \\ 0 & otherwise \end{cases} \tag{2.1}$$

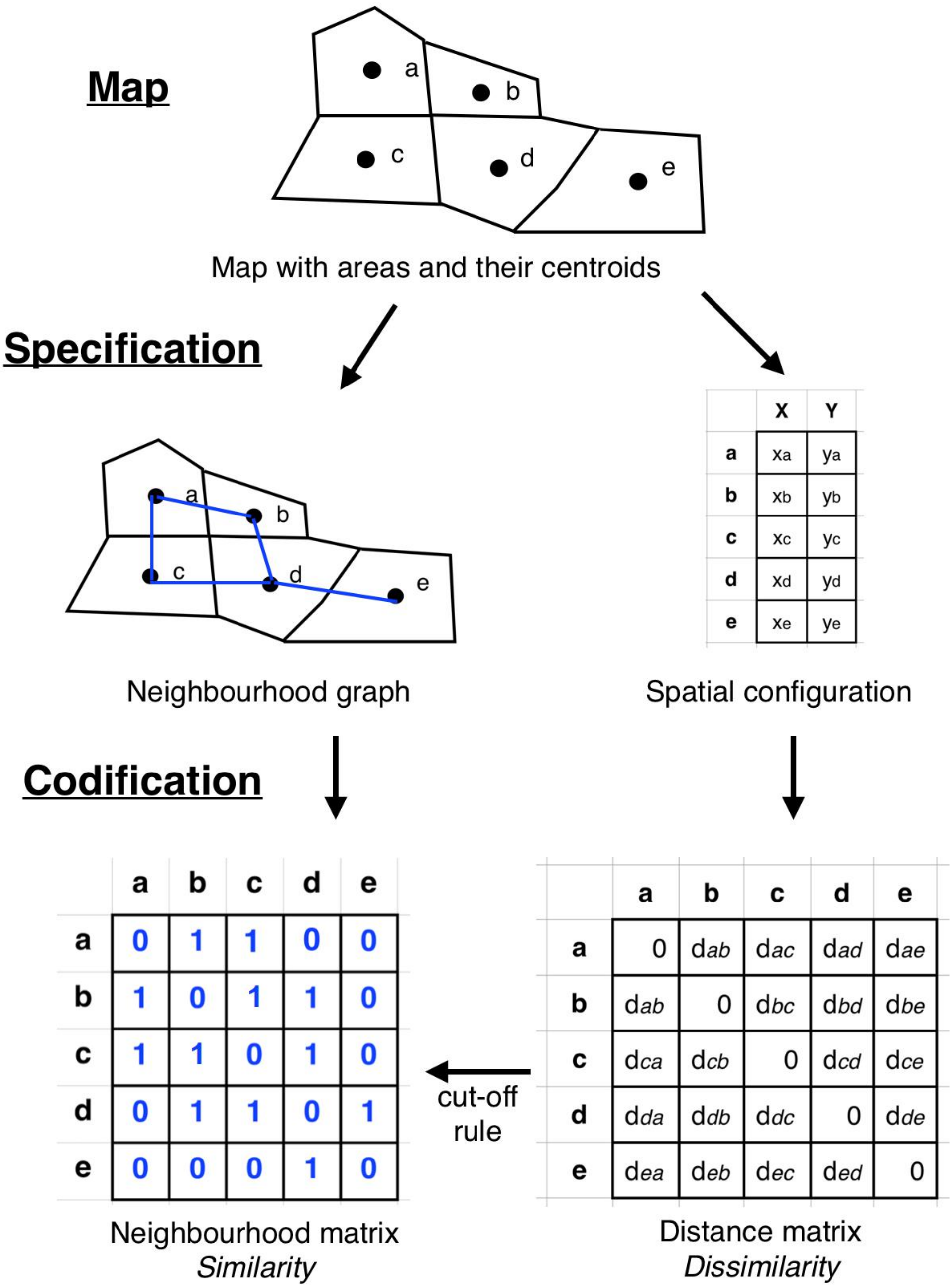
Conversely, the distance matrix measures dissimilarity between zones. The higher d_{ij} , the more different the zones. With, if an Euclidian distance is used : $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, α and β being the geographical coordinates of the observations.

The neighbourhood matrix is used in the study of areal spatial data, while the distance matrix is rather used for geostatistics (see Chapter 5: "Geostatistics"). However it is possible to move from one to the other by setting a minimum distance beyond which the observations are no longer considered as neighbouring.

The spatial dependence structure may not be geographical. Any relevant dual relationship may be used to define a neighbourhood graph. For instance:

— **at individual level**: friendship bonds, frequency of communication, citations;
— **at company level**: head office-subsidiary ties, similarities in terms of markets;
— **at international level**: strategic alliances, trade flows, shared belonging to an organisation, cultural exchanges and migratory flows.

The following sections detail different neighbourhood specifications.



	X	Y
a	xa	ya
b	xb	yb
c	xc	yc
d	xd	yd
e	xe	ye

	a	**b**	**c**	**d**	**e**
a	0	1	1	0	0
b	1	0	1	1	0
c	1	1	0	1	0
d	0	1	1	0	1
e	0	0	0	1	0

	a	**b**	**c**	**d**	**e**
a	0	d*ab*	d*ac*	d*ad*	d*ae*
b	d*ab*	0	d*bc*	d*bd*	d*be*
c	d*ca*	d*cb*	0	d*cd*	d*ce*
d	d*da*	d*db*	d*dc*	0	d*de*
e	d*ea*	d*eb*	d*ec*	d*ed*	0

Figure 2.1 – Codifying spatial relations
Source: *Tiefelsdorf 1998*

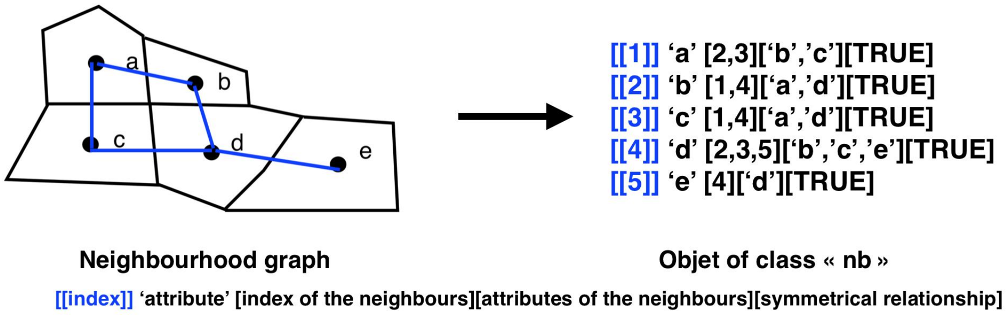
The "list of neighbours" object in R

Package *spdep* makes it possible to define the relationships between spatial objects. In R, the class of an object defines all its properties and how the statistician can use it. Neighbourhood relationships are recorded in an object of class `nb`.

Assume *n* spatial observations and *neighbours_nb* the spatial object containing the associated neighbourhood relationships. *neighbours_nb* is a list of length *n*. Each element [*i*] of the list contains a vector with the index of the neighbours of the item indexed *i*. If [*i*] does not have neighbours, the list contains only 0. The list also contains a vector of characters corresponding to the attributes of each neighbourhood zone, as well as a logical value indicating whether the relationship is symmetrical (see Figure 2.2). The main information about the object *neighbours_nb* can be derived using the function:

```
summary(neighbours_nb)
```

The documentation for package *spdep* provides more information (Bivand et al. 2013b).



[[1]] 'a' [2,3]['b','c'][TRUE]
[[2]] 'b' [1,4]['a','d'][TRUE]
[[3]] 'c' [1,4]['a','d'][TRUE]
[[4]] 'd' [2,3,5]['b','c','e'][TRUE]
[[5]] 'e' [4]['d'][TRUE]

[[index]] 'attribute' [index of the neighbours][attributes of the neighbours][symmetrical relationship]

Figure 2.2 – The list of neighbours in *spdep*

2.1.2 Defining neighbours based on distance

Once we have a set of points spread across the territory, we can calculate the distance between them. These points may be specific locations where the information has been observed, or points representative of each zone, for example their centroid. In this case, the underlying assumption is that the distribution of the variable within each zone is sufficiently homogeneous to approximate it to a single point.

Neighbourhood graphs materialise the links between the various entities. They are defined in such a way that they represent the underlying spatial structure as closely as possible. There exists many different types of neighbourhood graphs. Here we will show the graphs based on geometric concepts and closest neighbours.

Neighbourhood graphs based on geometric concepts

Delaunay's Triangulation is a geometric method that connects points into triangles such that the minimum angle of all triangles is maximised (this triangulation is aimed at avoiding "elongated" triangles), see Figure 2.3 and . Delaunay's Triangulation has interesting geometric and mathematical properties. However, the concept of neighbourhood can be refined.

The **sphere-of-influence based graph** links two points if their "circles from the nearest neighbour" overlap. The "circle of the nearest neighbour" of point P is the largest circle centred in P and that contains no other points than P (see Figure 2.4 and 2.5b). The graphs of the sphere of influence are not necessarily connected, *i.e.* all points in the study set are not necessarily interconnected.

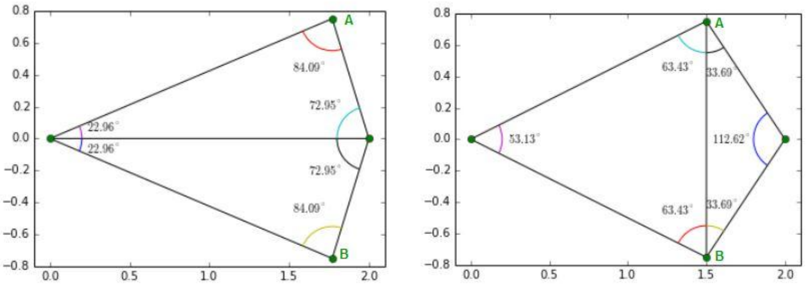


Figure 2.3 – Delaunay triangulation associated with different positions of points A and B
Source: *Gustavo [CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)], from Wikimedia Commons*

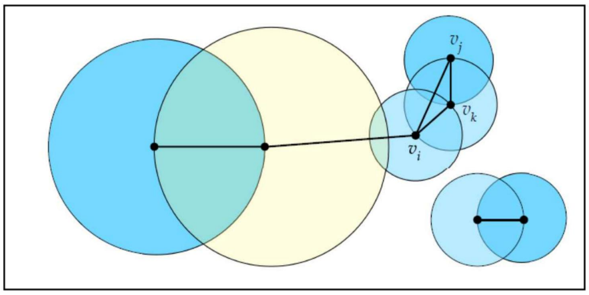


Figure 2.4 – The graph of the sphere of influence of a set of points
Source: *Toussaint 2014*

Gabriel's graph links two points p_i and p_j if and only if all other points are outside the circle with diameter $[p_i, p_j]$. Gabriel's graph removes some links of Delaunay's graph, see Figure 2.5c.

The **graph of relative neighbours** considers that two points p_i and p_j are neighbours if

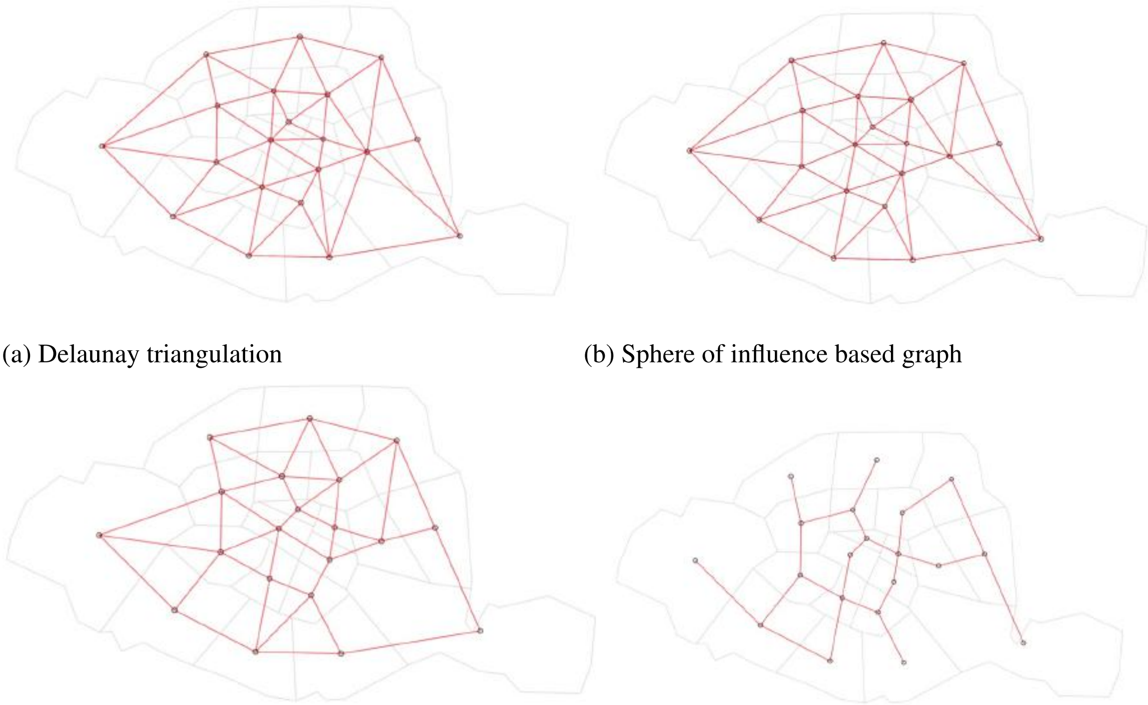
$$d(p_i, p_j) \leq max[d(p_i, p_k), d(p_j, p_k)] \quad \forall k = 1, ..., n \quad k \neq i, j \tag{2.2}$$

with $d(p_i, p_j)$ the distance between p_i and p_j. The graph of relative neighbours imposes fewer connections than Delaunay's Triangulation or the sphere of influence graph, see Figure 2.5d. Toussaint 1980 explains that it adapts better to data by requiring the fewest links.

The neighbourhood graphs shown here on Parisian districts are all sub-graphs of Delaunay's Triangulation (see Figure 2.5). They have the advantage that it leaves no unit without neighbours. However, they are only implemented in R with Euclidean distance, while other types of distances, such as the great-circle distance, can be better-suited to certain studies.

Application with R

```
library(rgdal) #To import MIF/MID files
library(maptools) #To import files Shapefile
library(tripack) #To calculate neighbours based on distance
library(spdep)

#Spatial File Import
arr75 <- readOGR("~/ArmF.TAB", "ArmF")

#Neighbours based on the concept of graph
#The input file is a matrix with geographical coordinates
#or an object from type SpatialPoints
coords <- coordinates(arr75)
IDs <- row.names(as(arr75,"data.frame"))

#Delaunay Triangulation
Sy4_nb <- tri2nb(coords, row.names=IDs)
plot(arr75, border='lightgray')
plot(Sy4_nb,coordinates(arr75),add=TRUE,col='red')

#Sphere-of-influence based graph
Sy5_nb <- graph2nb(soi.graph(Sy4_nb,coords),row.names=IDs)
plot(arr75, border='lightgray')
plot(Sy5_nb,coordinates(arr75),add=TRUE,col='red')

#Gabriel Graph
Sy6_nb <- graph2nb(gabrielneigh(coords), row.names=IDs)
plot(arr75, border='lightgray')
plot(Sy6_nb,coordinates(arr75),add=TRUE,col='red')

#Relative neighbours graph
Sy7_nb <- graph2nb(relativeneigh(coords), row.names=IDs)
plot(arr75, border='lightgray')
plot(Sy7_nb,coordinates(arr75),add=TRUE,col='red')
```



(a) Delaunay triangulation

(b) Sphere of influence based graph

(c) Gabriel graph

(d) Relative neighbours graph

Figure 2.5 – Four neighbourhood graphs of Parisian districts based on geometric concepts

Neighbourhood graphs based on nearest neighbours

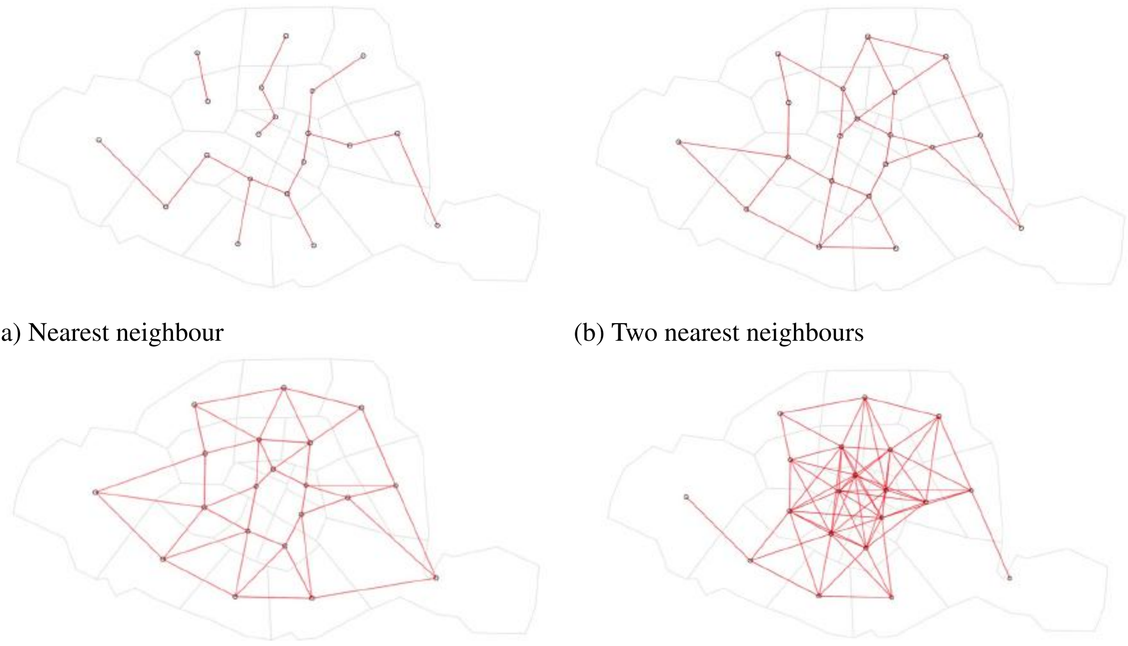
A second method consists in selecting the k closest points as neighbours (Figure 2.6). This method has the advantage that it leaves no point without a neighbour, which is not required when conducting a spatial analysis, but generally offers a better reflection of reality (a geographical zone is rarely completely isolated). However, it is sometimes difficult to identify the value of k that reflects the true underlying spatial relationships. The graphs based on the k closest neighbours are not necessarily symmetrical.

The choice can also be made to keep only the points located at a certain distance. The `nbdists` function of R can be used to calculate the vector of distances between neighbours. It makes it possible to determine the minimum distance d_{min} above which all points have at least one neighbour, then the `dnearneighb` function allows to keep as neighbours only the points between distances 0 and d_{min}. This "minimum distance" method is not adapted to irregularly spaced data, as the minimum distance required for a relatively isolated point having at least one neighbour is much higher than the distance to the closest neighbour of a point located in a dense zone. There will therefore be significant disparities in the number of neighbours, see Figure 2.6d (Bivand et al. 2013b).

Application with R - Source: *Bivand et al. 2013b*

```
#graphs based on the nearest neighbours
Sy8_nb<-  knn2nb(knearneigh(coords,k=1),row.names=IDs)
Sy9_nb<-  knn2nb(knearneigh(coords,k=2),row.names=IDs)
Sy10_nb<-  knn2nb(knearneigh(coords,k=3),row.names=IDs)

plot(arr75, border='lightgray')
plot(Sy8_nb,coordinates(arr75),add=TRUE,col='red')
```

```
#Study of the average distance of the nearest neighbours
dsts <- unlist(nbdists(Sy8_nb,coords))
summary(dsts)
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    820    1188    1678    1707    2016    3412
max_1nn <- max(dsts)

#Calculation and representation of neighbours at the minimum distance
Sy11_nb<- dnearneigh( coords, d1=0, d2=max_1nn,row .names=IDs)
plot(arr75, border='lightgray')
plot(Sy11_nb,coordinates(arr75),add=TRUE,col='red')
```



(a) Nearest neighbour

(b) Two nearest neighbours

(c) Three nearest neighbours

(d) Neighbours at a minimum distance

Figure 2.6 – Four graphs based on the nearest neighbours of Parisian districts

2.1.3 Defining neighbours based on contiguity

When the areal data consist in a partition of the entire territory, the concept of "distance between observations" can become quite ambiguous. Example 2.1 illustrates the limits of using the distance between centroids to define the notion of neighbourhood.

■ **Example 2.1 — Ambiguity of the notion of distance between centroids.** Let R_1, R_2, R_3 be three distinct zones. It can be considered that since R_2 and R_3 are separated in space, but both are adjacent to R_1, they are both closer to R_1 than to one another. However, the centroids in these zones are equidistant from each other (see Figure 2.7). Summarising the proximity between zones by the distance between the centroids results in a partial loss of the richness of the spatial relationships.

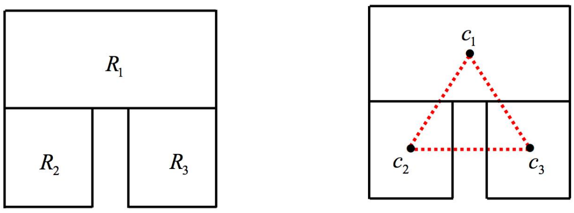


Figure 2.7 – Left: three zones - Right: distance between centroids
Source: *Smith 2016*

■

This subsection introduces various concepts of contiguity and presents the way in which package *spdep* in R makes it possible to create a list of neighbours.

In the sense of **Rook** contiguity, neighbours have at least two common boundary points (a segment). This matches the movement of the Rook in chess. For two zones to be adjacent in the sense of **Queen** contiguity, they only need to share one common boundary point. This matches the movement of the Queen in chess. Figure 2.8 illustrates these concepts in the case of a regular grid of points. When polygons have an irregular shape and surface, the differences between the Rook and Queen neighbourhoods become more difficult to grasp. It should also be noted that a very large zone surrounded by smaller zones will have a far greater number of neighbours than its neighbouring zones.

The neighbourhood in the sense of contiguity is often used to study demographic and social data, in which it may be more important to be on either side of an administrative boundary than to be located at a certain distance from one another.

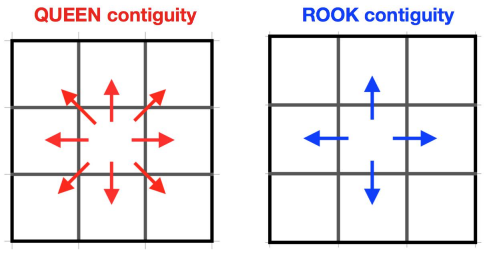


Figure 2.8 – Definition of Queen and Rook contiguity

Application with R

Construction of Queen and Rook neighbourhood graphs for Paris districts (Figure 2.9)

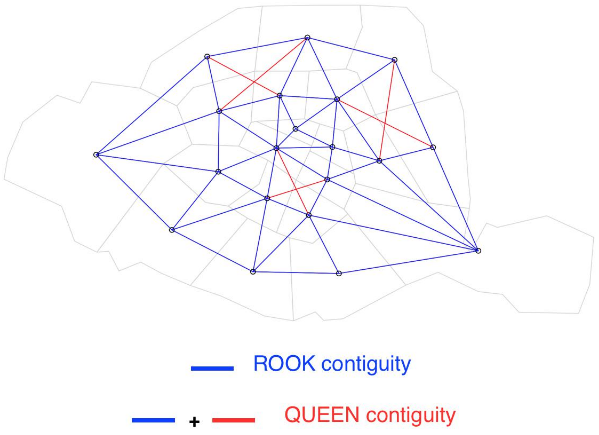


Figure 2.9 – Queen and Rook contiguity in Paris districts

```
#The input file is a SpatialPolygons file
#Extraction of list of neighbours as defined in QUEEN contiguity (by default)
arr75.nb<-  poly2nb(arr75)

#Extraction of list of neighbours as defined in ROOK contiguity
arr75.nb.ROOK<-  poly2nb(arr75, queen=FALSE)

#Visual representation of neighbours:
plot(arr75, border='lightgray')
plot(arr75.nb, coordinates(arr75),add=TRUE,col='red')
plot(arr75.nb.ROOK, coordinates(arr75),add=TRUE,col='blue')
```

2.1.4 Defining neighbours based on the optimisation of a trajectory

About the travelling salesman

Some methods such as spatial sampling (see Chapter 10 "Spatial sampling") require prior data sorting. When the latter are characterised by two variables (*i.e.* their geographical coordinates in the plan), how to choose a sorting method becomes a complex theoretical problem.

One solution consists in running a *path* along all the points, and sorting them by their order of appearance when the path is taken. The neighbours of a given point are then the points located just before or just after along the path.

Out of the set of possible paths, some have characteristics that are better suited to the desired objectives, such as, for instance, reducing sampling variance. This is the case of the *shortest path*. It minimises the sum of the distances between two consecutive points. This path, which does not set any particular constraints on the starting or arrival point, is known in the literature of graph theory as the *Hamilton path* (Figure 2.11b) associated with a graph the edges of which are weighted.

A particular and well-known case of *shortest path* is that of the travelling salesman. It represents the path which a travelling salesman must take to visit all his customers, minimising the distance travelled and managing to return home in the evenings. Such a path corresponds to a Hamiltonian cycle (Figure 2.11c).

Looking for a shortest path is a classic optimisation problem in the context of graph theory. It can be seen in particular in Euler's attempt to solve the problem of the seven Königsberg bridges [1]. It also plays a part in questions relating to Eulerian or Hamiltonian graphs [2]. Today, there are no algorithms in polynomial time that can be used to find the shortest path. When the number of points is high, the search for the optimal path requires the use of heuristics [3] resulting in a local optimum. They are available in package *TSP* in R (Hahsler et al. 2017).

When the distance is Euclidean and the number of points is reasonable, around a few hundreds, an exact solution can be found thanks to the `concorde` programme (Applegate et al. 2006). This programme can be called up directly from package *TSP* in R.

Lastly, the search for a Hamiltonian path from a distance matrix is equivalent to that of a Hamiltonian cycle, provided that a line and a column formed of 0 are added to the original matrix (Garfinkel 1985). Package *TSP* explicitly refers to this case with the `insert-dummy` function.

Other methods

The *general randomized tessellation stratified* method (GRTS , Stevens Jr et al. 2004) is popular in spatial sampling, as it makes it possible to get a spatially-balanced sample for a finite population of individuals (distinct and identifiable units of dimension 0 of a discrete population, *e.g.* trees in a forest), a linear population (continuous units of dimension 1, *e.g.* rivers) or a population of surfaces (continuous units of dimension 2, *e.g.* forests). It is based on a path built from a class of functions referred to as *quadrant-recursive* (Mark 1990), making it possible to ensure that certain two-dimensional spatial proximity relationships are still preserved in one-dimensional space.

The idea of the method is to project the coordinates on a unit square, then cut this square into four cells, each of which is cut again into four sub-cells, etc. To each cell, a value is assigned, resulting from the order in which the division was carried out, ultimately making it possible for the units to be placed on the path going through the two-dimensional space.

Figure 2.10 shows the initial stages of cutting, which can be implemented with package `spsurvey` in R (Kincaid et al. 2016). However, with the GRTS method, *large jumps* (Figures 2.11d) are created along the paths, which can affect the accuracy of the estimates.

Application with R - Source: Finding a shorter path

```
library(TSP)
library(miscTools)

#The utility software "concorde" must be downloaded at this address:

http://www.tsp.gatech.edu/concorde/downloads/downloads.htm
#and called from R

Sys.setenv(PATH=paste(Sys.getenv("PATH"),"z:/cygwin/App/Runtime/Cygwin/bin"
    ,sep=";"))
concorde_path("Z:/concorde/")

#The input data are a distance matrix
```

[1] The issue studied by Euler was: in the city of Königsberg, is it possible to take a walk in which each of the 7 bridges is used once and only once? (**euler1741solutio**).

[2] A Eulerian graph is a graph that can be travelled from a given vertex and walking along each edge exactly once before returning to the starting point vertex. It can be likened to a drawing that can be etched without ever lifting the pencil from the page. A Hamiltonian graph is a graph that can be travelled passing across all vertices and only once. A Hamiltonian graph is not necessarily Eulerian because in a Hamiltonian cycle, it is entirely possible to omit to pass through certain edges.

[3] A heuristic is a calculation method that quickly (in polynomial time) provides a feasible solution, albeit not necessarily optimal.

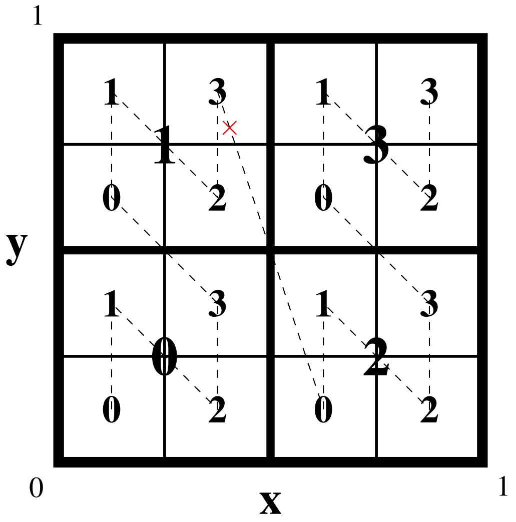


Figure 2.10 – Constructing a path with the GRTS method
Note: The value "13" is associated with the unit the position of which is a red cross, thus making it possible to position it on the path.

```
test <-as.matrix(read.csv("U:/paris.csv",header=FALSE,sep="\t"))

#rounding errors can lead to the matrix not being completely symmetrical.

tsp <-(symMatrix(test[upper.tri(test, TRUE)], nrow=nrow(test), byrow=TRUE))
#an object readable by TSP is created
tsp<-TSP(tsp)
#The concorde method is applied to this object.
tour<-solve_TSP(tsp, method = "concorde")
```

2.2 Attributing weights to neighbours

2.2.1 From a list of neighbours to a weight matrix

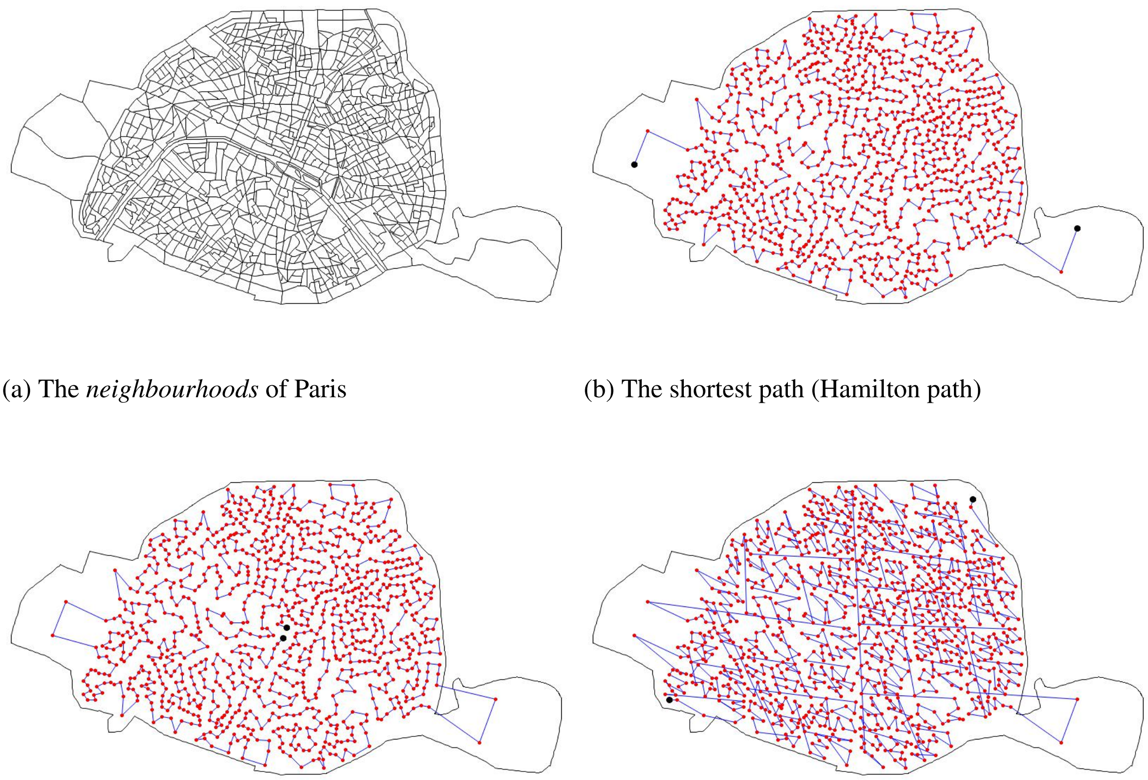
Once the neighbourhood graph has been defined and codified into a list of neighbours, the link between points i and j is transformed into the element w_{ij} of the weight matrix $\mathbf{W}$. The weight matrix $\mathbf{W}$ is the "formal expression of spatial dependency between observations" (Anselin et al. 1988).

Defining the weight matrix

— Most commonly, the weight matrix is a binary contiguity matrix (see Figure 2.12):

$$w_{ij} = \begin{cases} 1 & si \quad i \quad and \quad j \quad are \quad linked \quad in \quad space \\ 0 & otherwise. \end{cases} \tag{2.3}$$

— The weight matrices can also take into account the distance between the geographical zones, as relationships becoming smaller with distance: 1 if $d < d_0$ - 0 otherwise, $\frac{1}{d^\alpha}$, or $e^{-\alpha d}$ with α an estimated or predetermined parameter. Using a maximum distance beyond which $w_{ij} = 0$ makes it possible to limit the number of components with a value different from zero. As described in 2.1.2, when the size of the zones is heterogeneous, this method increases the risk of a considerable variability in the number of neighbours.



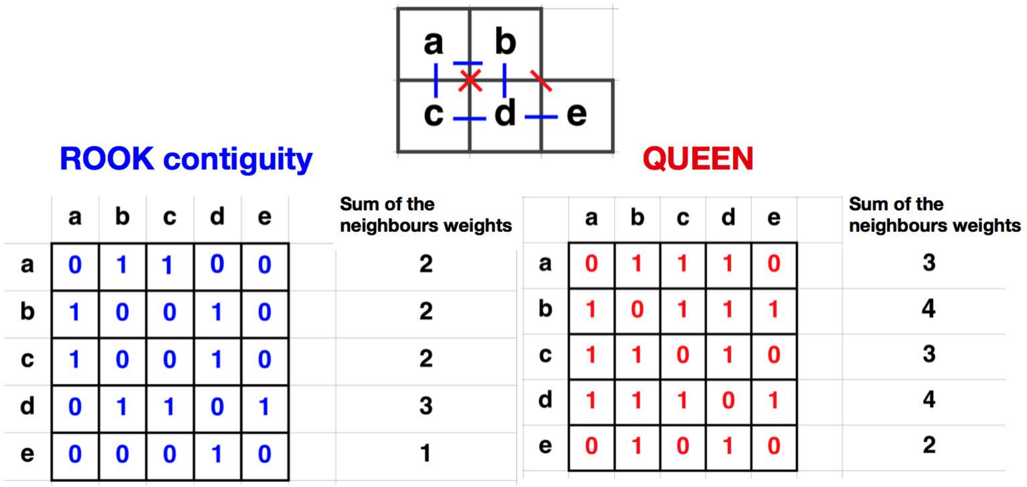
(a) The *neighbourhoods* of Paris

(b) The shortest path (Hamilton path)

(c) The Hamiltonian cycle, path of the travelling salesman

(d) Constructing a path using the GRTS method

Figure 2.11 – Looking for paths that cross through all the *neighbourhoods* of Paris



ROOK contiguity

	a	b	c	d	e	Sum of the neighbours weights
a	0	1	1	0	0	2
b	1	0	0	1	0	2
c	1	0	0	1	0	2
d	0	1	1	0	1	3
e	0	0	0	1	0	1

QUEEN

	a	b	c	d	e	Sum of the neighbours weights
a	0	1	1	1	0	3
b	1	0	1	1	1	4
c	1	1	0	1	0	3
d	1	1	1	0	1	4
e	0	1	0	1	0	2

Figure 2.12 – Binary weight matrix

— Lastly, certain matrices take the strength of relations between the zones into account. For example, weight can be defined by $\frac{b_{ij}^{\alpha}}{d_{ij}^{\beta}}$ with b_{ij} a measure of the strength of relationships between zones i and j (which is not necessarily symmetrical), such as the percentage of common boundaries, the total population, the wealth and d_{ij} the distance between the zones.

Some econometric studies are aimed at endogenising the weight matrices, but they are considered to be exogenous in most spatial econometric applications (Anselin 2013). In general, therefore, the neighbourhood weights must not be a function of the phenomenon which we are trying to explain.

The "weight list" object in R

The function nb2listw of package spdep makes it possible to convert a "list of neighbours" object into a "weight list" object. It is important to note that the "weight list" object, which corresponds to the weight matrix described above, is not a matrix $n \times n$ as represented in theory. It is a list containing the standardisation style and then for each observation: its attribute, the list of observation numbers of its neighbours, the list of the attributes of its neighbours and the list of the weights of its neighbours. Reference is often made to *sparse matrices*.

When a zone has no neighbours, the option zero.policy=TRUE makes it possible to generate a list of weights which takes value ’zero’ for observations without neighbours (if the option is FALSE, an error message is generated).

Application with R

```r
#Matrix based on contiguity
#The function nb2listw converts any object of the nb type into a weight
   list
arr75.lw <- nb2listw(arr75.nb)

#Matrix based on distance
#The mat2listw function converts a matrix into a weight list
library(fields) #to calculate the distance between two points
coords <- coordinates(arr75)
distance <- rdist(coords,coords)
diag(distance) <- 0
distance[distance >=100000] <- 0
#the weight decreases as a square of the distance, within a radius of 100
   km
dist <- 1.e12 %/% (distance*distance)
dist[dist >=1.e15] <- 0
dist.w <- mat2listw(dist,row.names=NULL)
```

Weight matrix standardisation

The sum of the weights of the neighbours of a zone is called its *degree of connection*. If the weight matrix is not standardised (*"B" coding scheme*), the degree of connection will depend on the number of its neighbours, which creates heterogeneity between the zones. According to Tiefelsdorf 1998, four types of standardisation can be distinguished:

— Line standardisation (*"W" coding scheme*): for a given zone, the weight ascribed to each neighbour is divided by the sum of the weights of its neighbours: $\sum_{j=1}^{n} w_{ij} = 1$. This standardisation makes the interpretation of the weight matrix easier, because $\sum_{j=1}^{n} w_{ij}x_j$ represents the average of variable x on all neighbours of observation i. Each weight w_{ij}

can be interpreted as the fraction of spatial influence on observation i ascribable to j. In contrast, such standardisation implies a certain degree of competition between neighbours: the fewer neighbours a zone has, the greater their weight. Moreover, when weights are inversely proportional to the distance between the zones, row standardisation makes them difficult to interpret.

— Global standardisation (*"C" coding scheme*): weights are standardised so that the sum of all weights is equal to the total number of entities. All weights are multiplied by $\frac{n}{\sum_{j=1}^{n}\sum_{i=1}^{n} w_{ij}}$.

— Uniform standardisation (*"U" coding scheme*): weights are standardised so that the sum of all weights equals 1: $\sum_{j=1}^{n}\sum_{i=1}^{n} w_{ij} = 1$.

— Standardisation by variance stabilisation (*"S" coding scheme*): let $\mathbf{q}$ be the vector defined by: $\mathbf{q} = (\sqrt{\sum_{j=1}^{n} w_{1j}^2}, \sqrt{\sum_{j=1}^{n} w_{2j}^2}, ..., \sqrt{\sum_{j=1}^{n} w_{nj}^2})^T$.
Let matrix $\mathbf{S}^* = [diag(\mathbf{q})]^{-1}\mathbf{W}$.[4] From $\mathbf{S}^*$, we calculate $Q = \sum_{j=1}^{n}\sum_{i=1}^{n} s_{ij}^*$ from which we deduce the standardised weight matrix: $\mathbf{S} = \frac{n}{Q}\mathbf{S}^*$.

Standardisation by variance stabilisation was introduced by Tiefelsdorf in order to reduce the heterogeneity in the weights due to differences in size and the number of neighbours between zones. Line standardisation gives more weight to observations bordering the study zone, with a small number of neighbours. On the contrary, with global or uniform standardisation, the observations in the centre of the study zone, with a large number of neighbours, are subject to more external influences than the border zones. This heterogeneity can have a significant impact on the results of spatial autocorrelation tests.

The weight of the standardised matrix based on the "S" coding scheme varies less than those of the standardised matrix based on the "W" scheme. The sum of the weights of the lines varies more for the "S" scheme than for the "W" scheme, but less than for the "B", "C" and "U" schemes (Bivand et al. 2013b).

Whether the coding scheme is in row, global, or by variance stabilization, the sum of all elements in the matrix is always n, which enables the spatial autocorrelation statistics using the matrix to be comparable to each other.

Application with R

```
#The style option makes it possible to set the type of standardisation
arr75.lw <- nb2listw(arr75.nb,zero.policy=TRUE, style="W")
names(arr75.lw)
## [1] "style"      "neighbours" "weights"
summary(unlist(arr75.lw$weights))
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.1250  0.1667  0.1833  0.1961  0.2500  0.3333
```

2.2.2 Importance of the choice of weight matrix

When trying to test the importance of economic or social relationships between certain variables, the geographical location of the observations is a key parameter. First of all, observations in the same geographical zone are subject to the same external parameters (climate, pollution, etc.) Secondly, neighbouring observations mutually influence one another. Spatial econometrics models take these various interactions into account. These models use neighbourhood specification *via* weight matrix

[4] $diag(\mathbf{q})$ is a diagonal matrix with the components of $\mathbf{q}$ on its main diagonal

W. Within the scientific community, opinions diverge on the influence of the definition of the weight matrix on results.

Bhattacharjee et al. 2005 note that: "The choice of weights is often arbitrary [...] and the result of the studies varies considerably depending on the definition of the spatial weights". A poor specification of **W** would lead to false conclusions. Having said that, as different weight matrix construction methods can be applied, "[...] it is possible that one method leads to relevant results, though the risk of a poor specification will always weigh on the chosen model". (Getis et al. 2004).

The aim is that the weights w_{ij} reflect interactions between observations as accurately as possible. The underlying assumptions can be based on economic or sociological models. For example, zero weight beyond a certain distance will be justified by the fact that the influence of an employment area on its environment is constrained by the mobility of individuals, which is itself limited by their travelling time. However, Harris et al. 2011 emphasise that the concept of 'distance' is itself unclear. Distance is often defined by a geometric distance between two representative points of the study zones. But distance can also be the transport time between two regions (minimum time, or time taking the least expensive route), or for instance be proportional to interactions between zones. According to Harris et al. 2011, "the consequence of using measures connected with contiguity or distance to weight the observations of neighbouring regions is that a spatial interaction structure is imposed without any means of verifying its reliability, such that it may be poorly specified."

Harris et al. 2011 show some alternative approaches to weight matrix construction. These methods aim at minimising the *ad hoc* hypotheses in matrix specification. However, no method gets rid of it completely.

Not all researchers are as pessimistic: LeSage et al. 2010 consider that the belief that weight matrix has a crucial influence on results is due to errors in interpreting the coefficients of spatial econometrics models, or to errors in model specification. In their words, this belief is "the biggest myth in spatial econometrics". They argue that if we look at the average effect of explanatory variables on dependent variables, the differences in weight matrix specification do not have a significant influence on results. However, Lesage et al. 2009 acknowledge that much remains to be done toward better characterising the concept of equivalence between matrices.

References - Chapter 2

Anselin, Luc (2013). *Spatial econometrics: methods and models*. Vol. 4. Springer Science & Business Media.

Anselin, Luc and Daniel A Griffith (1988). « Do spatial effects really matter in regression analysis? » *Papers in Regional Science* 65.1, pp. 11–34.

Applegate, David et al. (2006). *Concorde TSP solver*.

Bhattacharjee, Arnab and Chris Jensen-Butler (2005). « Estimation of spatial weights matrix in a spatial error model, with an application to diffusion in housing demand ». CRIEFF Discussion Papers.

Bivand, Roger S, Edzer Pebesma, and Virgilio Gomez-Rubio (2013b). « Spatial Neighbors ». *Applied Spatial Data Analysis with R*. Springer, pp. 83–125.

Garfinkel, R.S. (1985). « Motivation and modelling (chapter 2) ». *E. L. Lawler, J. K. Lenstra, A.H.G. Rinnooy Kan, D. B. Shmoys (eds.) The traveling salesman problem - A guided tour of combinatorial optimization,* Wiley & Sons.

Getis, A and J Aldstadt (2004). « On the specification of the spatial weights matrix ». *Geographical Analysis* 35.

Hahsler, Michael and Kurt Hornik (2017). *TSP: Traveling Salesperson Problem (TSP)*. R package version 1.1-5. URL: `https://CRAN.R-project.org/package=TSP`.

Harris, Richard, John Moffat, and Victoria Kravtsova (2011). « In search of 'W' ». *Spatial Economic Analysis* 6.3, pp. 249–270.

Kincaid, Thomas M. and Anthony R. Olsen (2016). *spsurvey: Spatial Survey Design and Analysis*. R package version 3.3.

LeSage, James P and R Kelley Pace (2010). « The biggest myth in spatial econometrics ». *Available at SSRN 1725503*.

Lesage, James and Robert K Pace (2009). *Introduction to spatial econometrics*. Chapman and Hall/CRC.

Mark, David M (1990). « Neighbor-based properties of some orderings of two-dimensional space ». *Geographical Analysis* 22.2, pp. 145–157.

Smith, Tony E. (2016). *Notebook on Spatial Data Analysis*. http://www.seas.upenn.edu/ ese502/notebook.

Stevens Jr, Don L and Anthony R Olsen (2004). « Spatially balanced sampling of natural resources ». *Journal of the American Statistical Association* 99.465, pp. 262–278.

Tiefelsdorf, Michael (1998). « Modelling spatial processes: The identification and analysis of spatial relationships in regression residuals by means of Moran's I (Germany) ». PhD thesis. Université Wilfrid Laurier.

Toussaint, Godfried T (1980). « The relative neighbourhood graph of a finite planar set ». *Pattern recognition* 12.4, pp. 261–268.

— (2014). « The sphere of influence graph: Theory and applications ». *International Journal of Information Thechnology and Computer Science* 14.2.