Geospatial Analytics 24-25

Notes

University of Pisa

M.Sc. in Data Science and Business Informatics

Contents

Chapter 1

Introduction

A geographic information system (GIS) is a computer system used to capture, store, query, analyze, and display geospatial data. **Geospatial data** describes both the location and the characteristics of spatial features: for example, if we want to describe a road, we may refer to its location and its features (length, name, speed limit, etc.). Other than single entities, geospatial data can also describe trajectories, specifying the sequence of locations constituting it.
The following chapter will give an overview of the basic concepts used in GISs and spatial data analysis.

1.1 Geographic Coordinate Systems

When using a GIS, any map layers used together must align spatially; to make sure this is true, we need to use some common spatial reference system for all maps. GIS users normally work with locations expressed on a plane using a coordinate system expressed in x- and y-coordinates, while the actual, real-life locations represented by them are on Earth's surface (which is ellipsoidal). A **map projection** is used to convert the Earth's surface to a plane.

The system used to locate points on Earth is called **geographic coordinate system**. This system is defined by two coordinates: **longitude** and **latitude**. They are angular measures which measure the angle at which the point can be found with respect to the **prime meridian** and the **equator**; longitute represents the angle east or west from the prime meridian, while latitude represents the angle north or south of the equatorial plane.

Meridians are lines of equal longitude. The prime meridian passes through Greenwich, England, and corresponds to 0°. **Parallels** are lines of equal latitude. The equator is the line corresponding to 0° latitude.

In a **plane coordinates** system, longitute and latitude correspond to x and y coordinates respectively. Logitude takes positive values in the easter hemisphere, and negative values in the western hemisphere; latitude takes positive values north of the equator, and negative values south of the equator.
Longitute and latitude values may be expressed in different ways:

- **Decimal degrees** (**DD**): represented by a single decimal value;
- **Degrees-minutes-seconds** (**DMS**): represented by a set of three values, corresponding to degrees, minutes, and seconds. 1 degree corresponds to 60 minutes, and 1 minute corresponds to 60 seconds;
- **Radians** (**rad**): similar to DD, but expressed in radians instead of decimal values: 1 degree is equal to 0.01745 rad.

As mentioned before, planet Earth can be approximated as an **ellipsoid**: this shape is obtained by rotating an ellipse by its shortest axis. Indeed, Earth is wider along the equator (its major axis) than it is between the poles (its minor axis). Another parameter that describes an ellipsoid is the **flattening** (f), calculated as $f = \frac{maj-min}{maj}$, and it describes the difference between the two axes.

A **datum** is a mathematical model of the Earth which is used as the reference to calculate the geographic coordinates of a point (or even the elevation, if we consider vertical datums). A datum is defined as: the pair *longitude, latitude* of coordinates of an initial point wich will be the origin, an ellipsoid, and the separation of the ellipsiod and the Earth at the origin.

Distances on the Earth's surface are not straight lines; they are instead represented by **geodesics**: through any two points (not antipodal), there is exactly one "great circle" that connects them. The two points separate the great circle in two parts: the shorter of the two is their geodesic distance. Since the Earth is nearly spherical, geodesic distances are correct with an error up to 0.5%. The geodesic distance between points A and B is calculated as:

$$\cos(d) = \sin(lat_A)\sin(lat_B) + \cos(lat_B) + cos(lon_A - lon_B),$$

where d is the angular distance between the two points.

1.2 Trajectories, Tessellations, Flows

Let u be an individual. A **trajectory** $T_u = \langle p_1, p_2, \ldots, p_{nu} \rangle$ is a time-ordered sequence composed by the spatio-temporal points visited by u. A spatio-temporal **point** is a pair $p = (t, l)$, where t is the time, and $l = (x, y)$ is the point visited at that time.

Given an area A, a **tessellation** is a set of geographical polygons with the following properties:

- It contains a finite number of polygons called **tiles**:
$$\mathbb{G} = \{g_i : i = 1, \ldots, n\}$$
- The tiles are non overlapping:
$$g_i \cap g_j = \emptyset \, , \forall i \neq j$$
- The union of all tiles completely covers the tessellation:
$$\bigcup_{i=1}^{n} g_i = A$$

A tessellation can be **regular** or **irregular** depending on the shape of its tiles. Regular tessellation may use equilateral triangles, squares, hexagons; irregular tessellation may use buildings, census cells, administrative units. A **spatial join** is used to associate a point with the tile it belongs to. Since the tiles are non overlapping and cover the entire area, each point belongs to one and only one tile. **Voronoi tesselations** are a particular type of tessellation that partition the plane into regions (called **cells**), each closer to a secific point (called **seed**) out of a set. Each of these cells is defined as the set of points that are closest to the seed of the cell itself than any other seed in that area.

Given a tessellation, the **flow**
$$y(g_i.g_j)$$
represents the number of people/objects moving between g_i and g_j. A trajectory refers to a single entity, while a flow refers to the total amount of entities moving between two points. Flows can be derived from a set of trajectories, but the inverse is not true.

1.3 Raster and Vector Data Models

The raster and vector data models are two ways to represent geographic information in GISs. In both cases, data can be stored in several **thematic layers**, each of which contains a set of objects of the same nature. For example, a layer may contain information about buildings, another about streets, another about rivers, and so on.

The **raster data** model divides the space into a regular grid of square cells with a given size (which defines the **resolution**). This format is often used for images, where each element corresponds to a pixel. Depending on how much information is assigned to a cell, data can be **single-band** (one attribute per cell), or **multi-band** (several attributes per cell). Raster data is typically sourced from satellites.

The **vector data** model uses discrete objects (points, lines, polygons) to represent spatial features. Each object can have its own properties and relationship with the others. A **point** is a zero-dimensional object with a *location* property (expressed as x,y coordinates). A **line** in a one-dimensional object with two properties: *location* and *lenght*. It be either straight or curved. A **polygon** is a two-dimensional object with three properties: *location*, *area*, and *perimeter*. These objects are expressed differently depending on the data format used by the software/platform.

Objects in a layer are sometimes also called *spatial features*; for this reasin, the variables associated to them are called *attributes* (and not features). To represent geometric objects in a GIS, we can use one of the following models:

- **Geo-relational data model**, where objects and attributes are stored separately, and associating each object to the corresponding attributes requires a join operation (at the advantage of possibly saving space if a certain attribute(s) is (are) only possessed by few objects);

- **Object-relational data model**, where objects and attributes are stored together in a single table, making retrieval much faster (but possibly increasing the amount of space needed to store everything).

In principle, vectors can model everything; raster data is a discretized view of the same information. Raster data is better suited for “dense” data; it can be more efficient in those cases where a raster representation may need a very high number of objects, but precision is not a concern. Vector data can also be converted to raster data, and vice versa. **Rasterization** is the process of transforming vector data into raster data, and produces a discrete approximation. **Vectorization** is the inverse process: it may be difficult at times, and many algorithms and methods have been developed to perform it.

At times, vectors and raster information can be used together in multi-layer data. Some information is better modeled with one format than the other: for example, street networks or locations of interest are often encoded as vector layers, while things like land usage are encoded as raster layers.

1.4 Spatial Operations

The most important spatial operations are:

- **Intersection**: returns all the points in common with the operands.
- **Union**: returns the union of the two operands. In some tools, They are kept as separate objects, meaning that the result is always a multipolygon. As an alternate operation, the same tools offer the **dissolve** operation, which instead merges the two objects into a single one.
- **Difference**: returns all the points in the first operand which are not in the second.
- **Buffering**: creates a buffer, i.e., an expanded area, around the object. The result is equivalent to replacing each point in the geometry with a circle with a given radius.
- **Spatial join**: like in relational databases, joins merge the information of two objects. THe join can be **inner** (the output contains only pairs in common with both objects), or **outer/left/right** (the output also contains non matching objects with the *NULL* value in place of the missing attributes).

1.5 Spatial Patterns and Spatial Correlation

Point Pattern Analysis (**PAA**) is the study of point patterns, i.e., the spatial distribution of points in an area. Spatial distributions are typically categorized into three types:

- **Uniform (discrete)**: points are evenly distributed in the area;
- **Random**: points are distributed according to a random process;
- **Clustered**: points appear to be grouped (clustered) in some areas.

A basic form of point pattern analysis consists in determining summary statistics such as mean center, standard distance, and standard deviational ellipse.
Mean center is the average of the x and y coordinate values:

$$\bar{s} = \left(\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n} \right)$$

Standard distance measures the variance between the average distance of the features to the mean center:

$$d = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \mu_x)^2 + (y_i - \mu_y)^2}{n}}$$

Similar to standand distance, **standard deviational ellipse** measures the standard distances for each axis:

$$d_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu_x)^2}{n}}$$
$$d_y = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \mu_y)^2}{n}}$$

Average Nearest Neighbor (**ANN**) is an algorithm that can be used to study patterns. For each point, its nearest neighbor is found as the point with the smallest distance to it. The average of all points' nearest neighbor distance is calculated as d_{obs}, and normalized with regards to the expected average if the pattern were random (d_{exp}), obtaining a ratio:

$$R = \frac{d_{obs}}{d_{exp}}$$

If $R = 1$, the pattern is random. If $R < 1$, the pattern is clustered, because the distances are smaller than expected; if $R > 1$, the pattern is uniform (or at least more dispersed than random).

Ripley's K-function is another popular method for analyzing point patterns. Usually, its normalized version, called **L function**, is used. Given n points in an area of size A, and a distance d, the L function is calculated as follows:

1. Compute all $n * (n-1)$ distances between each pair of points;
2. Compute ϕ, the fraction of distances that are $\leq d$;
3. Compute
$$L(d) = \sqrt{\frac{A}{\pi}\phi}\,.$$

$L(d) = d$ for random distributions. If $L(d)$ is higher, the data is more clustered; if it is lower, the data is more dispersed. Different values of d can be explored to understand patterns at different spatial granularities.

Another important aspect is **density based analysis**. Density measurements can be either global or local. **Global density** is simply calculated as the ratio of observed points and the study region's area:

$$\hat{\lambda} = \frac{n}{A}$$

Density can also be measured at different locations of the study region. **Local density** is computed over a single tessellation cell; the chosen resolution will affect the resulting density calculation. **Kernel density** is another method of calculating density per-cell which considers also the points found in its neighborhood. Usually, given a cell c, the 8 adjacent cells are considered as the neighborhood N_c, and so the kernel density becomes:

$$\textit{Kernel density }(c) = \hat{\lambda}(c \cup N_c)$$

A variant is **weighted kernel density**, which assigns to each point a weight inversely proportional to the distance from the cell's center. Different weight functions can be used; a common one is the Gaussian function.

Autocorrelation is the correlation of the values of a same variable measured at different points in time (**temporal autocorrelation**)/space (**spatial autocorrelation**). An example of spatial autocorralation may be checking how much the temperature values in the points of a layer are influenced by the neighboring values. According to **Tobler's first law of geography**, "everything is related to everything else, but near things are more related than distant things": this is the fundamental assumption in spatial analysis.

Popular measures of spatial autocorrelation are:

- **Moran's I**, which calculates the autocorrelation between values of each point against all other points in its neighborhood:

  $$I = \frac{\sum_{i=1}^{n} \sum_{j:x_j \in N_{x_i}} w_{ij}(x_i - \mu_x)(x_j - \mu_x)}{s^2 \sum_{i=1}^{n} \sum_{j:x_j \in N_{x_i}} w_{ij}}$$

  where s^2 is the variance of the x values, and w_{ij} is a weight, typically defined as the inverse of the distance between the two points. Positive values mean positive correlation, negative values mean negative correlation;

- **Geary's C**:

  $$C = \frac{n - 1 \sum_{i=1}^{n} \sum_{j:x_j \in N_{x_i}} w_{ij}(x_i - x_j)^2}{2(\sum_{i=1}^{n} \sum_{j:x_j \in N_{x_i}} w_{ij}) * \sum_{i=1}^{n}(x_i - \mu_x)^2}$$

  The higher it is, the more different are nearby values (less correlation), the lower it is the closer they are (more correlation).

Both measures can also be interpreted as the average of local I/C values calculated across neighborhoods. These local values can be studied individually as well.

1.6 Spatial Interpolation and Regression

Spatial interpolation refers to the process of using points with known values (called **control points**) to estimate values at others. For example, we could estimate the temperature at a point with no recorded data by approximating it from known temperatures at nearby points. Ideally, control points should be well distributed across the study area, although this situation is rare in real-world applications since a study area oftentimes also contains data-poor areas.

Interpolation methods can be divided in two groups: **deterministic** and **stochastic**. The first group assumes that the known values are exact, with no assessment of errors for predicted values. The second group considers the presence of some random error in the known data and offers some assessment of prediction error with an estimated variance.

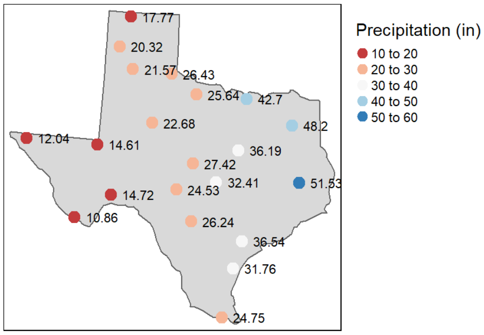

Figure 1.1: An example of samples corresponding to yearly precipitation recorded in different sites in Texas.

1.6.1 Deterministic Methods

Proximity Interpolation

Proximity interpolation (also known as **Thiessen interpolation**) is one of the simplest and oldest interpolation methods. The goal is to assign to all unsampled locations the value of the closest sampled location, producing a Voronoi tessellation over the study area. All the points within the same cell have the same value. A problem of this approach is that surface values change abruptly across the perimeter of adjacent cells, which is not realistic.

Inverse Distance Weighted Interpolation

Inverse Distance Weighted (IDW) interpolation calculates an average value using nearby weighted locations. The weight of each sample location is inversely proportional to the distance; the value at location j is given by:

$$\hat{Z}_j = \frac{\sum_i Z_i/d_{ij}^n}{\sum_i 1/d_{ij}^n}$$

Here, d_{ij} is the distance between points i and j, and n is an hyperparameter that controls the irrelevance of a point as the distance increases/decreases: the larger n is,

the less far away samples influence the interpolated value. For $n \to \infty$, the result is equivalent to proximity interpolation.
Values returned by this method are always within the range of the known values: $[Z_{min}, Z_{max}]$.

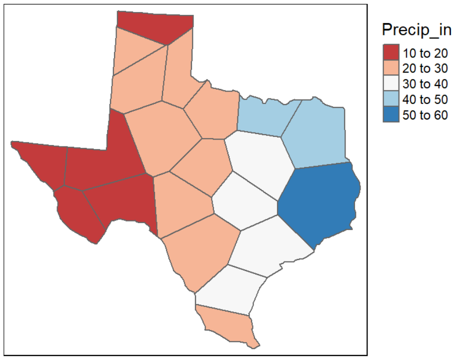

Figure 1.2: Interpolated values obtained by proximity interpolation.

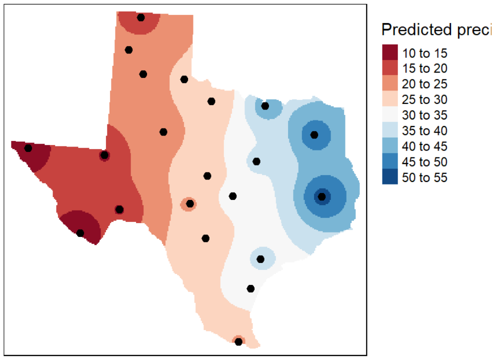

Figure 1.3: Interpolated values obtained by IDW interpolation.

1.6.2 Stochastic Methods

Trend Surface Interpolation

Trend surface analysis approximates points with known values using a polynomial equation. The same equation can then be used to predict values at other points. Depending on the order of the polynomial, the approximation can be more or less complex:

- A 0^{th} order surface is described by $Z = z$, where c is the average value of all samples;
- A 1^{st} order surface is described by $Z = aX + bY + z$, where X, Y are the coordinate pairs and c is a constant;
- A 2^{nd} order surface is described by $Z = aX^2 + bY^2 + cXY + dX + eY + z$;

and so on. Changing the order allows the model to better capture the complexity of the data, but using a value that is too high may result in overfitting the data, meaning that the model is too dependant on the known information and does not provide a useful prediction.

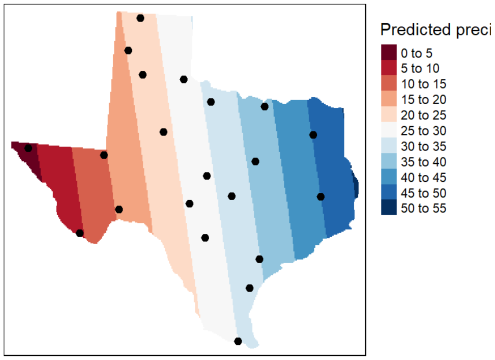

Figure 1.4: Interpolated values obtained by a 1^{st} order trend surface. The model is too rigid.

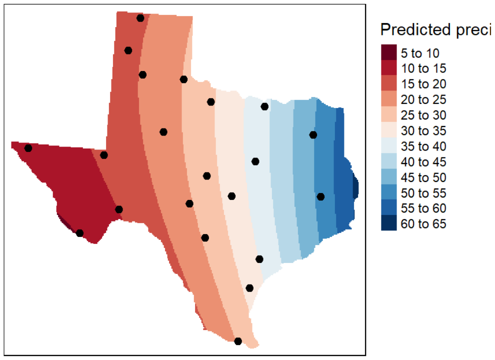

Figure 1.5: Interpolated values obtained by a 2^{nd} order trend surface. The model is better than before, but still not a good fit.

Kriging

Kriging differs from other methods in that it can assess the quality of prediction with estimated prediction errors. It assumes that the spatial variation of an attribute is neither random nor deterministic; instead, it is a combination of some spatially correlated component, a “drift” (assumed for now to be null), and a random error term.

The first step is **de-trending the data**, so that mean and variance of the data are constant across the whole study area. This can be done by using any trend model and subtracting the predictions from the data. From this point on, the method will focus on residuals, i.e., the remaining variability in te data that is not explained by the global trend. The second step is **constructing a (semi)variogram**. For each pair of points i and j, their semivariance is calculated as:

$$\gamma = \frac{(Z_i - Z_j)^2}{2}$$

The variogram is obtained as the plot of all the semivariances, using the x axis to represent distances, and the y axis to represent the semivariances. Usually, it is simplified by binning over the distance values and averaging the semivariances within each bin, obtaining a **sample experimental variogram**. The third step is to fit an **experimental variogram model** to find the parameters that best describe the spatial variance; there are many model variants, with the main ones being the Gaussian, linear, and spherical ones. Finally, **interpolation** can be performed. The general equation for estimating the residual at a point j is:

$$z_j = \sum_{i=1}^{s} z_i W_i$$

where s is the number of sample points used in the estimation, and W_i is the weight assigned to point i. These weights are derived by solving a set of simultaneous equations.

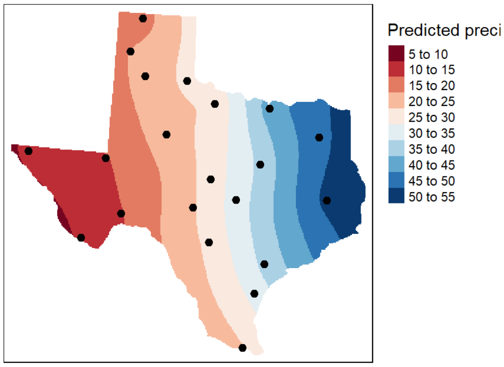

Figure 1.6: Interpolated values obtained by kriging.

1.7 Spatial Associations and Trend Detection

Bibliography

[1] Kang-Tsung Chang. *Introduction to geographic information systems.* Mcgraw-hill Boston, 2018.