Spatial Interpolation Notes

December 19, 2020

1 Introduction

We want to use interpolation because it is reasonable to assume that spatially distributed variables are also spatially correlated. It is not always true, but often worth exploring as part of an analysis. There are multiple different methods to interpolate data that depend on different underlying assumptions. These methods are described below.

Information from ARCGIS and Applied Spatial Data Analysis With R (2013) unless listed otherwise

There are two main methods used to interpolate data and estimate a surface for geospatial data:

- Inverse distance weighting (IDW) and spline methods
- Kriging

IDW and splines are deterministic interpolation methods as they are directly based on the surrounding values or smoothed formulas. Kriging is different as it uses autocorrelation and takes position into account in the statistical models. Kriging uses a certain number of neighbouring points, or all points within a specified radius (cf kNN).

The general formula for IDW and kriging is:

$$\hat{Z}(s_0) = \sum_{i=1}^{N} w(s_i) Z(s_i)$$

where: $Z(s_i)$ = the measured value at the ith location
$w(s_i)$ = an unknown weight for the measured value at the ith location
s_0 = the prediction location
N = the number of measured values

The difference between IDW and kriging is that in IDW, $w(s_i)$ only depends on distance to prediction location. In kriging $w(s_i)$ also depends on autocorrelation i.e. spatial relationship between prediction locations.

2 Inverse distance weighting (IDW)

IDW determines cell values using a linearly weighted combination of surrounding values. The weights are function of the inverse distance. The general form for the IDW function is:

$$\hat{Z}(s_0) = \frac{\sum_{i=1}^{n} w(s_i) Z(s_i)}{\sum_{i=1}^{n} w(s_i)}$$

where: $Z(s_i)$ = the measured value at the ith location
$w(s_i) = ||s_i - s_0||^{-p}$
$||\cdot||$ = Euclidean distance
p = an inverse distance weighting power, defaulting to 2

The value of p determines how much closer values are preferred. As p increases, IDW approaches a one-nearest-neighbour interpolation model. p can be selected using cross-validation.

Another way to control IDW interpolation is through selecting the number of neighbouring observations to include. This can improve speed of interpolation, and may be used when there is reason to believe that distant points have little correlation. There are two approaches for varying the number of points used for interpolation:

1. Varying search radius
    - The number of points to include is fixed, and the radius changes to include that set number
    - Depends on the density of observations fluctuating
    - The maximum radius can also be set, in which case all points will be included if that max radius is reached before n
2. Fixed search radius
    - Set a radius and minimum number of points
    - If $n <$ minimum number of points at set radius, the radius increases until the minimum is reached.

In addition to these two approaches, barriers can be created to limit the searches for neighbouring points, i.e. only search for this side of a river.

3 Kriging

One of the key benefits of kriging is that in addition to using autocorrelation, it is able to estimate uncertainty in the interpolation. It can do this because it is based on a spatial arrangement of empirical observations, rather than a presumed model of spatial distribution. Although kriging preferentially weights closer observations, its use of autocorrelation means that clusters are not over-fit i.e. lowering bias as each point in a cluster provides less information than a single point.

The kriging predictor is an "optimal linear predictor" and an exact interpolator. This means that prediction error is each interpolated value is calculated to minimize the prediction error for that point. It also means that the interpolated value for sampled points is equal to the actual value, and all interpolated values will be the Best Linear Unbiased Predictors (BLUPs).

Kriging is only helpful where there is at least moderate spatial autocorrelation. If there is not, then simpler methods like IDW, will generally perform as well as kriging.

3.1 Assumptions in kriging

Information for assumptions from Columbia

For kriging to be used, there are a number of assumptions/conditions to be met. These conditions can be checked in exploratory data analysis.

1. Assumption of intrinsic stationarity
    - Means that the joint probability distribution does not vary across the study space, so the same parameters (e.g. mean, range and sill etc) are valid across the space

- Means one variogram is valid across the space

2. Assumption of isotropy
    - Uniformity in all directions (semivariance identical in all directions)

By making these assumptions, we are assuming that the samples are randomly generated by the function $Z(s)$ with a mean (m) and residual ($e(s)$).

$$Z(s) = m + e(s)$$

where: $E\left(Z(s)\right) = m$

The assumption of *intrinsic stationarity* and *isotropy* can be relaxed to create models where the mean varies spatially. In instances like this, the measured values can be assumed to be randomly generated by a linear function of known predictors $X_j(s)$.

$$\begin{aligned} Z(s) &= \sum_{j=0}^{p} X_j(s)\beta_j + e(s) \\ &= X\beta + e(s) \end{aligned} \tag{1}$$

3.2 Creating a prediction map with kriging

There are two steps:

1. Create the variograms and covariance functions to estimate the spatial autocorrelation values that depend on the model of autocorrelation (fitting a model).
2. Predict the unknown values

3.2.1 Variography (spatial modelling/structural analysis)

There are often too many pairs of spatial points to calculate and plot the distance for each pair. Instead, spatial distances are put into lag bins i.e. all points in the range $40m < h \leq 50m$ of point A, and calculate the semivariance. The semivariance is equal to half the variance of the differences between all possible points spaced a constant distance apart. Assuming *isotropy* and *intrinsic stationarity*, we can generalise the distances between points and use the distance $||h||$ rather than the vector $\mathbf{h}$, i.e. use bins.

$$\hat{\gamma}(\tilde{h}_j) = \frac{1}{2N_h} \sum_{i=1}^{N_h} \left(Z(s_i) - Z(s_i + h)\right)^2, \ \forall h \in \tilde{h}_j$$

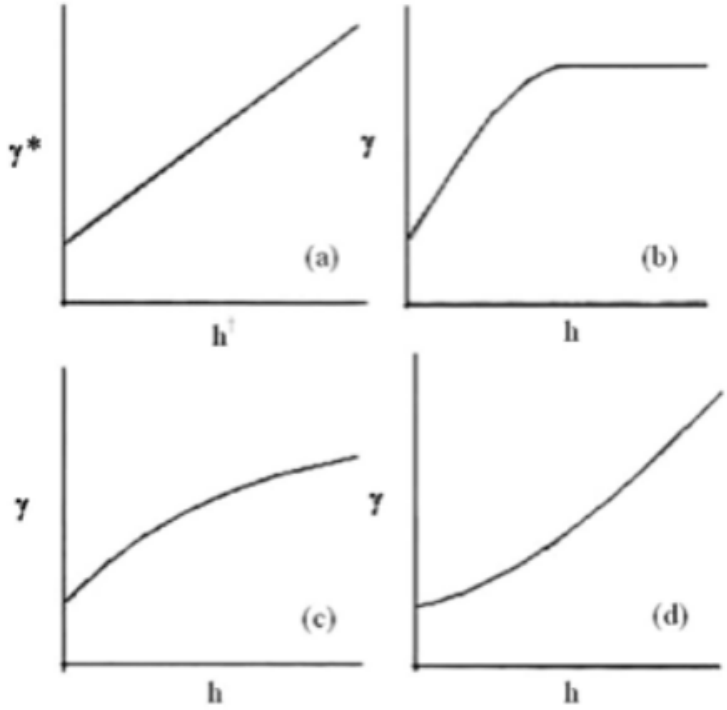
where: $Z(s_i)$ = the measured value at the ith location
$\hat{\gamma}(\tilde{h}_j)$ = sample variogram
N_h = sample data points
$\tilde{h}_j$ = distance bins (intervals)

Plotting the distance vs semivariance produces an empirical semivariogram. Closer items should be more similar, therefore lower semivariance. The opposite is true for further points.

A model is fit to the empirical semivariogram (cf regression). Different types of models can be fit the the semivariogram, and the optimal model can be selected using metrics like RMSE, MLE, and Bayesian methods:

- Spherical (most common)
- Circular
- Exponential
- Gaussian
- Linear

Figure 1: Different types of models used in spatial modelling (Poilou 2008). a) Linear semi-variogram; (b) spherical semi-variogram; (c) exponential semi-variogram; and (d) power semi-variogram



There are a number of key points on the figures:

- Range
  - The Range is the point at which the semivariance first levels off
  - Items within the range are autocorrelated (distance matters)
  - Items outside the range are not autocorrelated (distance no longer changes the semivariance)
- Sill
  - The Sill is the height at which the semivariance levels off to
- Nugget
  - The minimum value of semivariance ($\gamma(h = 0)$)
  - Theoretically there is no semivariance when $h = 0$, but in reality it is present due to measurement error or spatial sources of variation at distances smaller than the sample interval (or both)
- Partial Sill
  - Amount of semivariance between Sill and the Nugget

3.2.2 Predictions

Now a model has been fit to the semivariance and autocorrelation can be observed, predictions can be made within the domain. Kriging differs from IDW as it uses the semivariogram to calculate the weights. There are a number of methods used in kriging:

1. Ordinary kriging
    - Assumes the constant mean is unknown
2. Universal kriging
    - Assumes there's a prevailing trend, relaxing the assumption of stationarity for the mean, but maintaining a constant variance
    - Trend is modelled with polynomial function, and subtracted from observed
    - Semivariogram is modelled on the residuals to produce autocorrelations
3. Block kriging
    - Estimates averaged values over gridded "blocks" rather than single points
    - These blocks often have smaller prediction errors than are seen for individual points
4. Covariate kriging
    - Additional observed variables (which are often correlated with each other and the variable of interest) are used to enhance the precision of the interpolation of the variable of interest at each location
5. Poisson kriging
    - Used for incidence counts and disease rates

3.3 Limitations

Information for limitations from Columbia

There are a number of limitations of kriging.

1. Since the weights of the kriging interpolator depend on the modeled variogram, kriging is quite sensitive to misspecification of the variogram model
2. Similarly, the assumptions of the kriging model (e.g. that of second-order stationarity) may be difficult to meet in the context of many environmental exposures
    - Some newer methods (e.g. Bayesian approaches) have thus been developed to try and surmount these obstacles
3. In general, the accuracy of interpolation by kriging will be limited if the number of sampled observations is small, the data is limited in spatial scope, or the data are in fact not amply spatially correlated
    - In these cases, a sample variogram is hard to generate, and methods such as land-use regression may prove preferable to kriging for spatial prediction

4 Natural Neighbour

Natural neighbour is a local method that examines samples near the point of interest and evaluates the relative overlap with their areas. The relative overlaps are then used to create the weights for interpolation. Because of this, it is also known as "area-stealing" (Sibson) interpolation. Natural neighbour interpolation therefore does not infer trends that are not already present in the data, and the surface passes through the points, and is smooth in between.

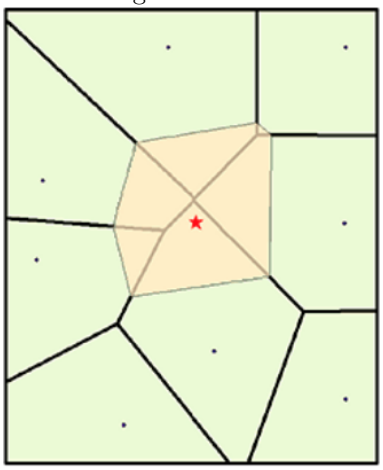
The areas are called Voronoi (Thiessen) polygons. Voronoi polygons are created by examining the space around points and drawing the boundary so that every place inside the boundary is closest to the polygon's point than any other. Formally this is written as:

$$R_k = \{x \in X \mid d(x, P_k) < d(x, P_j),\ \forall j \neq k)\}$$

where: R_k = Voronoi polygon of point k
P_k = Point k
P_j = Neighbouring point j

An example of this can be seen in the figure below.

Figure 2: Natural neighbour method of interpolation



5 Splines

Splines are a smoothing function that pass through all the input points and attempt to create a smooth surface between them. As such, it is best for gently varying surfaces e.g. pollution concentrations. The surface is fit to a specified number of neighbouring input points. The basic spline is also known as a thin plate interpolation. There are two conditions that minimum curvature splines must follow:

1. The surface must pass through all data points
2. The surface must have minimum curvature i.e. minimize the cumulative sum of squares of the second derivative terms of the surface at each point

One possible issue with thin plate interpolation is that there may be rapid change in first derivatives around each data point. Increasing the number of points used for interpolation can help to smooth the surface as the cell is influenced by a greater number of more distant points. Splines create rectangular regions of equal size, with the same number in the $x-$ and $y-$ directions. Each region must contain at least 8 points, but different densities resulting from data that is not uniformly distributed can lead to regions containing different numbers of points.

Generally, the spline formula is:

$$S(x,y) = T(x,y) + \sum_{j=1}^{N} \lambda_j R(r_j)$$

where: N = total number of points to be used in interpolation
λ_j = coefficients found by the solution of a system of linear equations
r_j = the distance from the point (x,y) to the jth point

There are two spline types, which define the terms $T(x,y)$ and $R(r_j)$ differently.

5.1 Regularized splines

A regularized spline creates a smooth and gradually changing surface, allowing values outside those observed in the data.

$$T(x,y) = a_1 + a_2 x + a_3 y$$

where: a_i = coefficients found by the solutions of a system of linear equations

and,

$$R(r) = \frac{1}{2\pi}\left\{\frac{r^2}{4}\left[\ln\left(\frac{r}{2\tau}\right) + c - 1\right] + \tau^2\left[K_0\left(\frac{r}{\tau}\right) + c + \ln\left(\frac{r}{2\pi}\right)\right]\right\}$$

where: r = the distance between the point and the sample
τ^2 = the Weight parameter
K_0 = the modified Bessel function
c = a constant equal to 0.577215

In regularized splines, the Weight parameter (τ^2) specifies the weights attached to the third derivatives terms during minimization. Larger weights result in smoother surfaces and smooth first-derivative surfaces. Typical values range between 0 and 0.5.

5.2 Tension splines

A tension spline creates a less smooth surface with values more tightly constrained by the sample data range.

$$T(x,y) = a_1$$

where: a_1 = a coefficient found by the solutions of a system of linear equations

and,

$$R(r) = -\frac{1}{2\pi\varphi^2}\left[\ln\left(\frac{r\varphi}{2}\right) + c + K_0(r\varphi)\right]$$

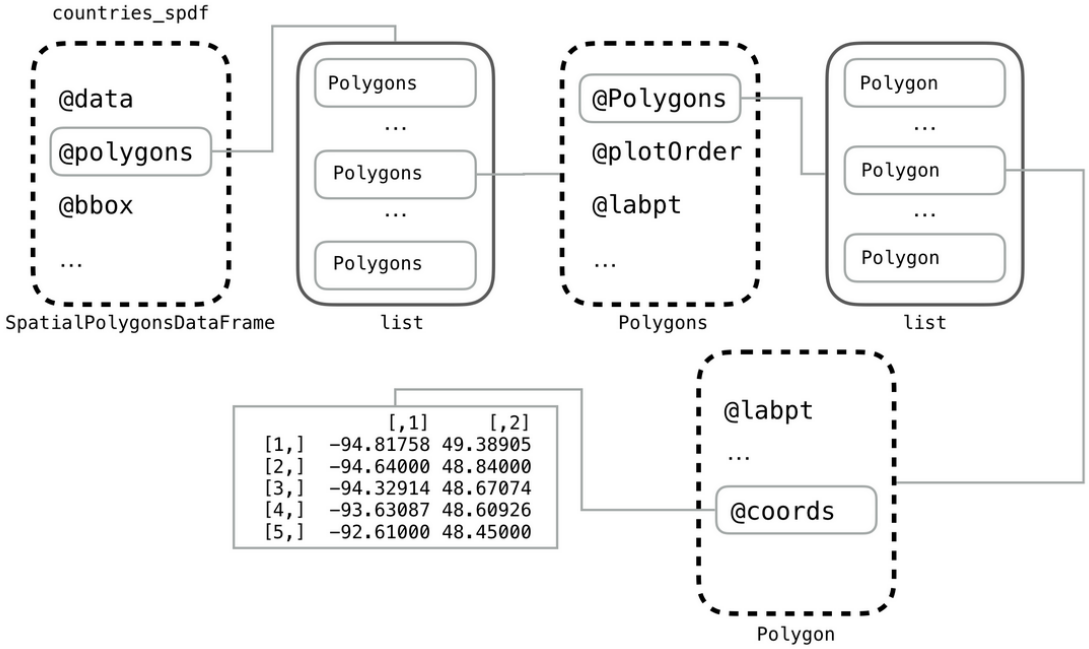
where: r = the distance between the point and the sample
φ^2 = the Weight parameter
K_0 = the modified Bessel function
c = a constant equal to 0.577215

The tension method differs from regularized splines as it attaches the Weight parameter (φ^2) to first-derivative terms, not third-derivative terms. Larger values of φ^2 lower the tension and result in a coarser surface as the first-derivative surface is not smooth, passing through all the points. $\varphi^2 = 0$ results in a basic thin plate surface. Typical values range between 0 and 10.

6 Datacamp: Visualizing geospatial data in R

- *ggmap* package very useful for quickly producing static spatial plots
  - *ggmap:get_map(long, lat)* pulls basemap based lat long
  - *ggmap(*ggmap*, base_layer = ggplot(df, aes(long, lat))) + geom_point()* allows you to plot layers over map and retain same aes() e.g. for faceting
- Different types of spatial data
  - Point data
  - Line data - assumes points connected by straight lines
  - Polygon data
    - Data associated with enclosed area of points
    - *ggplot2::geom_poly()*
  - Raster (grid) data
    - Regular grid specified by origin and steps in x and y axis, and data is associated with cells in grid
    - *ggplot2::geom_tile(aes(fill = *var*))* used to create raster
- Polygon data
  - Difficult to described
    - Order of joining up points matters
    - Polygons may be broken up e.g. by river therefore needing multiple polygons to describe it
- *sp* data structures better that *dataframes* for storing spatial data as don't have to repeat info like groups and order for polygons, and contains information about the coordinate sytem itself, which is useful when working with multiple systems/for sharing
- *spdf* is an **S4** data type
  - Useful adaptation of *sp* structure as also contains dataframe
  - Items are **slots** that are accessed with the @ symbol e.g. *spdf@polygon*
    - Each **slot** contains a list that is *another* **S4** object (e.g. Polygons) (see Fig. 3)
    - Can pull information as with normal dataframes using $ symbol e.g.
      - *is_nz = countries_spdf$name == "New Zealand"*
      - *nz = countries_spdf[is_nz,]*
- *tmap* package designed to plot spatial data, rather than requiring dataframe format, like *ggplot2*
  - *tm_shape()* adds basemap
  - *tm_raster()* creates choropleth
  - Can save interactive *leaflet* map using *tmap_save(filename = *.html)*
- *raster* package better to work with raster data than *sp* and *ggplot2*

Figure 3: *spdf* data structure



  - Creates an **S4** object
  - More efficient as stores data in matrix like format, where each value is associated with a cell in the raster grid
    * Multiple matrices act as layers to provide more information
      · *Multi-band/multi-layer* rasters
    * Reduces reproducing the same grid
  - *rasterVis::levelplot()* good for quickly visualizing rasters

- *classInt::classIntervals()* useful to bin continuous variables for choropleths
- *rgdal::readOGR* used to read in shape files (polygons)
- *proj4string()* allows you to define the coordinate reference system (CRS) and projection when no present, or print it where it is present
- *rgdal::spTransform()* used to transform CRS
  - *tmap* does the transformation automatically

7 Datacamp: Spatial analysis with sf and raster in R

8 Datacamp: Spatial statistics in R

9 Datacamp: Interactive maps with leaflet with R