GIS Exercise - April 2011

Maria Antonia Brovelli,
Laura Carcano, Marco Minghini

ArcGIS exercise 4 - Spatial Interpolation

Introduction:

Interpolation is the procedure of predicting the value of attributes at unsampled sites from measurements made at point locations within the same area or region. It is used to convert the data from point observations to continuous fields.

Before producing the final surface, we should have some idea of how well the model predicts the values at unknown locations. And so crossvalidation and validation help to make a decision as to which model provides the best predictions.

Crossvalidation and validation use the following idea—remove one or more data and then predict their values using the data at the rest of the locations. In this way, you can compare the predicted value to the observed value and obtain useful information about your previous decisions about the interpolation you have used.

In this exercise we will see different interpolation techniques. By specifying different parameters of the interpolation and examining the quality of interpolation, we will compare these different techniques.

Data:  

*Lidar1.dbf*
1. Visualization of the data in 2D

- **Add data** -> select the file Lidar1.dbf
- Right click on the layer name “Lidar1” -> **Display XY data** – put X field=N1, Y field=N2, Z field=N3; change the reference system clicking on Edit... -> ... -> Monte Mario Italy1.prj

Not all the data are displayed, and so we have to set the number of data to be treated, assigning a maximum sampling number larger than the total number of the data contained in the dataset

- Right click on the layer name “Lidar1 Events” -> **Properties** -> **Quantities** -> **Quantity numbers: Classify** -> **Sampling** – put a number higher than 38000 (ex. 40000); and change the colours of the field using a color ramp over the N3 fields, with for example 5 classes.

**Expected result:**

![Image of expected result]
Tasks:

Apply the different interpolation methods we have seen (with different parameters) on the Lidar1 terrain dataset (the terrain points are those with the parameter N5=0, the others are points belonging to buildings); use cross-validation and validation to identify the best method to adopt.

2. Assign to the entire dataframe the reference system
   
   - Right click on the dataframe name -> Properties -> Coordinate System ->
     Predefined - Projected coordinate systems – National Grids – Europe – Monte Mario Italy1.prj

3. Extract terrain points and create the dataset lidar1_terrain.dbf

   The attribute N5 identifies which points belong to terrain (N5=0) and which belong to buildings (N5=999).
   
   - Right click on the layer name “Lidar1 Events” -> Open attribute table -> Select by attribute -> write “N5”=0

   Extract these selected data
   
   - Attribute table: Options -> Export… -> Export the selected data, name it as “lidar1_terrain.dbf” and select “dBASE Table” as file type

4. Load data lidar1_terrain.dbf, display them and assign them a coordinate system:

   - Right click on the layer name “lidar1_terrain” -> Display XY data – put X field = N1, Y field = N2, Z field = N3; change the reference system clicking on Edit... -> … -> Monte Mario Italy1.prj

   - Right click on the layer name -> Properties -> Quantities -> Quantity numbers: Classify -> Sampling – put a number higher than 22008 (ex. 25000); and change the colours of the field using a color ramp over the N3 fields, with for example 5 classes.
5. Subdivide the data set “lidar1_terrain Events” into two subsets (corresponding respectively to 90% - “lidar1_terrain Events_training1” and 10% - “lidar1_terrain Events_test1”)

- Geostatistical Analyst -> Subset Features... – put as Input features = lidar1_terrain Events, Output training feature class = “Lidar1 Events_training”, Output test feature class = “Lidar1 Events_test”; Size of training feature subset = 90; Subset size units = PERCENTAGE_OF_INPUT
Expected Result:

Check now the distributions of the two subsets and compare them, using the general QQ plot:

- *Geostatistical Analyst -> Explore Data -> General QQ Plot; Data source #1 = “lidar1_terrain Events_training”, Attribute = N3, Data source #2 = “Lidar1_terrain Events_test”, Attribute = N3. Handling coincidental sample: choose Include all.*

Draw some conclusion about the distribution of the two subsets, thus the applicability of the two data sets in cross-validation and validation.
6. Detect and remove (if present) outliers (for instance using a global polynomial interpolation and by examining the residuals)

- **Geostatistical Analyst -> Geostatistical Wizard -> Global Polynomial Interpolation; Source dataset = “lidar1_terrain Events_training1”, Data field = N3. Handling coincidental sample: choose Include all.**

Validation:

- Right click on the layer name “Global polynomial interpolation [lidar1_terrain Events_training]” -> Validation/Prediction -> Input geostatistical layer = “Global polynomial interpolation”; Input point observation locations = lidar1_terrain Events_training; Field to validate on = N3; Output statistics at point locations = Validation_training.shp.

- Right click on the layer name “Global polynomial interpolation [lidar1_terrain Events_training1]” -> Validation/Prediction -> Input geostatistical layer = Global polynomial interpolation; Input point observation locations = lidar1_terrain Events_test; Field to validate on = N3; Output statistics at point locations = Validation_test.shp.
Keep record of the Mean (M) and the Root-Mean-Square (RMS) values of the Training group and the Test group by changing the power degree gradually, with the result table, decide the optimal interpolation degree.

<table>
<thead>
<tr>
<th>power</th>
<th>Training Mean</th>
<th>Training RMS</th>
<th>Test Mean</th>
<th>Test RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>4.622286</td>
<td>0.044778</td>
<td>4.600193</td>
</tr>
<tr>
<td>2</td>
<td>0.000000</td>
<td>4.379869</td>
<td>0.077456</td>
<td>4.364094</td>
</tr>
<tr>
<td>3</td>
<td>0.000000</td>
<td>2.552742</td>
<td>0.521131</td>
<td>2.552598</td>
</tr>
</tbody>
</table>

Make the Global Polynomial Interpolation on the whole lidar1_terrain.dbf database with the optimal degree

- **Geostatistical Analyst** -> **Geostatistical Wizard** -> **Global Polynomial Interpolation**;
  - *Source dataset*= “lidar1_terrain Events”, *Data field*= N3. *Handling coincidental sample*: choose *Include all*.

- Right click on the layer name “Global polynomial interpolation [lidar1_terrain Events\_training]” -> **Validation/Prediction** -> **Input geostatistical layer** = Global polynomial interpolation; *Input point observation locations* = “lidar1_terrain Events”; *Field to validate on* = N3; *Output statistics at point locations* = “Residuals.shp”.

The program save the file .shp and the other files attached to the shape (ex. the file .dbf). You can open the file .dbf with Excel and check the residuals.

**Residual (Error) = Observation (Measurement)-Trend (Prediction)**
From now on we will work on the errors. This means that the field we are interested in hereafter will be *Error* of the dataset *Residuals.dbf*.

**Visualize the Errors:**

- Right click on the layer name “Residuals” -> *Properties* -> *Quantities* -> *Quantity numbers: Classify* -> *Sampling* – put a number higher than 22008 (ex. 25000); and change the colours of the field using a color ramp over the ERROR fields, with for example 5 classes.

**Expected Result:**

7. **Interpolate the “Residuals” dataset using different interpolation methods and different parameters, trying to obtain the best surface interpolation** -> the programs gives in output the statistics of the cross-validation of the “Residuals” data
(1) Inverse Distance Weighting (IDW)

IDW is an exact interpolator and there are very few decisions to make regarding model parameters. To predict a value for any unmeasured location, IDW uses the measured values surrounding the prediction location; IDW assumes that each measured point has a local influence that diminishes with distance: greater weights are given to the points closest to the prediction location.

- **Geostatistical Analyst -> Geostatistical Wizard -> Inverse Distance Weighting -> put these options: Source dataset = residual_training; Data field = Error; Handling Coincidental Samples -> Use Mean**
By modifying the parameters in the control panel, we can change the neighborhood search strategy.

We may judge the interpolation quality by examining the RMSPE of the cross-validation and the validation: the smaller the RMSPEs are, the better the interpolation is.

**2) Local polynomial interpolation**

Local Polynomial (LP) is a moderately quick deterministic interpolator that is smooth (inexact). It is more flexible than the Global Polynomial method and with more parameter decisions.

While Global Polynomial interpolation fits a polynomial to the entire surface, Local Polynomial interpolation fits many polynomials, producing surfaces that account for more local variation.

- **Gesostatistical Analyst -> Geostatistical Wizard -> Local polynomial interpolator** -> put these options: Source dataset = residual_training; Data field = Error; **Handling Coincidental Samples** -> Use Mean
By modifying the parameters in the control panel and the power of the interpolation, one can alter the neighborhood search strategy.

Interpolation quality may be judged by examining the RMSPE of the cross-validation and the validation: the smaller the RMSPEs are, the better the interpolation is.

Modify the parameters, observe the RMSPEs in consequence, try to fix an optimal combination of the parameters in order to get the optimal local polynomial interpolation.

(3) Radial Basis Functions
Radial Basis Functions (RBF) are deterministic exact interpolators. RBF methods are a special case of splines. The RBF techniques are much more flexible than Inversed Distance Weighting method, but there are more
parameter decisions.

- **Gesostatistical Analyst -> Geostatistical Wizard -> Radial basis function** -> put these options: Source dataset = residual_training; Data field = Error; **Handling Coincidental Samples** -> **Use Mean**

Similar to the previous interpolations, once a specific kernel function is selected, by modifying the parameters in the control panel and the power of the interpolation, one can alter the neighborhood search strategy.
Interpolation quality may be judged by examining the RMSPE of the cross-validation and the validation: the smaller they are, the better the interpolation is.

Modify the parameters, observe the RMSPE in consequence, try to fix an optimal kernel function with a combination of the parameters in order to get the optimal Radial Basis Functions interpolation.

These are the results obtained with a certain combination of parameters; the results can be different:

<table>
<thead>
<tr>
<th>Method</th>
<th>MPE</th>
<th>RMSPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completely Regularized</td>
<td>-0.002</td>
<td>0.612</td>
</tr>
<tr>
<td>Inv. Multiquadratic</td>
<td>-0.008</td>
<td>0.644</td>
</tr>
<tr>
<td>Multiquadratic</td>
<td>-0.005</td>
<td>0.580</td>
</tr>
<tr>
<td>Tension Spline</td>
<td>-0.000</td>
<td>0.753</td>
</tr>
<tr>
<td>Thin Plate Spline</td>
<td>-0.010</td>
<td>0.635</td>
</tr>
</tbody>
</table>

Observation

Compare the interpolation results obtained by the previous methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>MPE</th>
<th>RMSPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inv. Distance (p=2.8262)</td>
<td>0.006</td>
<td>0.605</td>
</tr>
<tr>
<td>Global Interp (poly=1)</td>
<td>0.000</td>
<td>4.635</td>
</tr>
<tr>
<td>Global Interp (poly=3)</td>
<td>-0.008</td>
<td>2.557</td>
</tr>
<tr>
<td>Local Interp (poly=1)</td>
<td>0.009</td>
<td>0.693</td>
</tr>
<tr>
<td>Local Interp (poly=3)</td>
<td>0.003</td>
<td>0.709</td>
</tr>
</tbody>
</table>
(4) Kriging

Kriging is a moderately quick interpolator that can be exact or smoothed depending on the measurement error model. The flexibility of kriging can require a lot of decision-making.

The Kriging interpolation process supplies the possibility of remove the global trend (de-trend) before interpolation process, so we may use directly the “lidar1_terrain” data.

- Geostatistical Analyst -> Geostatistical Wizard -> Kriging -> put these options: Source dataset = “lidar1_terrain Events”; Data field = N3; Handling Coincidental Samples -> Use Mean
  - Step 2/6: Transformation type = None; Order of trend removal = Third; there are different kriging type: simple kriging assumes constant mean known in the study area, ordinary kriging when mean is constant but unknown, universal kriging assumes that unknown mean is changing smoothly in the study area and is considered as a trend.
- Step 4/6: Semivariogram/Covariance modeling is a key step between spatial description and spatial prediction. The empirical semivariogram and covariance provide information on the spatial autocorrelation of datasets. For this reason and to ensure that kriging predictions have positive kriging variances, it is necessary to fit a model (in other words, a continuous function or curve) to the empirical semivariogram/covariance. There are many semivariogram models (circular, spherical, exponential, gaussian,...)

Semivariogram parameters:

- RANGE and SILL
- LAG SIZE and NUMBER OF LAGS

there exists a rule of thumb for determining the lag size and the number of lags:
Lag Size*Number of Lags = 1/2 * maximum distance between pairs

We can also visualize the covariance function:

The steeper the curve is close to the origin, the more influence the closest neighbors will have on the prediction; consequently, the less smooth the output surface will be.

Vary the function to model the empirical semivariogram, try with different Nugget values and observe the results.

The criteria to judge the quality of the Kriging interpolation are:

✓ MPE (Mean Prediction Error) -> should be near zero;
✓ MSPE (Mean Standardized Prediction Error) -> should be near zero
✓ RMSPE (Root Mean Square Prediction Error) -> the smaller the RMSPE, the better the interpolation is.
The uncertainty of the prediction can be measured:

- ASE = RMSPE → good
- ASE > RMSPE → variability overestimate
- ASE < RMSPE → variability underestimate
- RMSSPE = 1 → good,
- RMSSPE < 1 → variability overestimate
- RMSSPE > 1 → variability underestimate

Model the semivariogram with different theoretical functions, using the criteria and try with different parameters, obtain the best interpolation.

**One possibility:**

- **Spherical model, nugget = 0.079:**
  - MPE = -0.00185
  - RMSPE = 0.5489
  - ASE = 0.5485
  - RMSSPE = 1.006

- **Circular model, nugget = 0.08 (lag size 12, lag # 8)**
  - MPE = -0.001785
  - RMSPE = 0.5493
  - ASE = 0.5474
  - RMSSPE = 1