Accounting for Spatial Autocorrelation

Michael L. Treglia

Material for Lab 7 of Landscape Analysis and Modeling, Spring 2015

This document, with active hyperlinks, is available online at:https://github.com/mltConsEcol/TU_LandscapeAnalysis_Documents/blob/master/Assignments/Lab7_Mantel_SpRegress.Rmd

Due Date: Thursday, 12 March 2015

PLEASE WRITE YOUR NAME ON YOUR ANSWER DOCUMENT

Questions

1) What is the default number of permutations used for calculating the Mantel test statistic?

2) Based on a Mantel test, is there a significant relationship between temperature and boreality? Provide Mantel's r and the p-value.

3) Based on a partial Mantel test, is there a significant relationship between temperature and boreality when controlling for spatial effects? (Provide Mantel's r value and the p-values).

4) What happens to the p-values of Mantel tests (and Partial Mantel Tests) if you halve the number of permutations?

5) Plot a Moran's I correlogram for the residuals of boreality as a function of wetness from a linear model that does not account for spatial autocorrelation. Provide some interpretation (Does there seem to be a lot of autocorrelation? Don't just look at significance, but look at magnitude of the y-axis too)

6) Plot a Moran's I correlogram for the residuals of boreality as a function of wetness from a linear model that *does* account for spatial autocorrelation. Provide some interpretation (Does there seem to be a lot of autocorrelation? Don't just look at significance, but look at magnitude of the y-axis too)

7) Run the analyses again, but this time use temperature as a predictor variable for boreality. Present a Variogram for residuals of an uncorrected model (use the GLS function and plot the line, as in the example for the model 'B1.gls').

    a) Does there appear to be a clear sill? If so, what is the range?
    b) What would you estimate as the value for the nugget?

8) For temperature as the predictor variable for boreality, what type of correlation function provides the best-fit model?

9) Present a variogram for the best-fit spatially corrected model. Does there still seem to be a problem with autocorrelation?

10) Sometimes there can still be autocorrelation in residuals after applying the terms to deal with autocorrelated error. What else could be causing these systematic errors? (Think about assumptions of working with spatial data!)

Introduction

Though it is informative to identify whether there is spatial autocorrelation for individual variables (e.g., with Moran's I or Geary's c), as we did in Lab 6, in ecology and evolutionary biology, we are often interested in the relationships of multiple variables. For example, how does temperature affect tree size or the ecological community of species at a site? As we discussed in lecture, the effect of spatial autocorrelation can obscure actual patterns in the data, thus we must deal with this appropriately.

We will focus this lab on an example from Chapter 7 of the book *Mixed Effects Models and Extensions in Ecology with R* by Alain Zuur et al, as part of Highland Statistics. You can check out the book's website, here: http://www.highstat.com/book2.htm. This group has a few books on analyzing ecological data with R - I recommend looking into their books for topics you may use in your research, and definitely check out their published papers too.

You can download R code and data used in the book at the book's website. We will focus on the "Boreality" dataset (Boreality.txt).

The focal dataset is centered on characterization of forest communities in part of British Colombia, Canada. The dependent variable is a metric of how many species present are associated with boreal forests; other variables include x- and y-coordinates, the data used to compute the boreality index, and remotely sensed measures of temperature ('T61'), Greenness ('GRN') and Wetness ('Wet'). Herein, we will only focus on a transformed metric of the Boreality index, wetness, and location (x- and y-coordinates). Here, we will look at the relationships between wetness and boreality, but you will also look at the remotely sensed temperature for the assignment. You will examine the data using Mantel tests, linear regression, and generalized least squares (GLS) regression techniques.

Necessary R Packages

The packages we will use in this lab are:

- ecodist
- gstat
- nlme
- sp
- pgirmess

(See previous lab materials for instructions on installing and loading packages in R.)

Importing and Setting Up the Data

As per usual, it often helpful to set your working directory appropriately and load any required packages. Then, you can import the dataset (you will have to have the data downloaded and un-zipped from the website linked above.) To open these data, you will use the 'read.table' function in R. Previously we have used the 'read.csv' function, which is a special case of read.table, for comma-separated values files. The Boreality dataset is stored in a tab-delimited format. While there are lots of arguments available for read.table, the only additional one you'll need to use after the filename is 'header=TRUE', indicating that the first row of information is the column names. We'll name the imported data 'Boreality'.

```
Boreality <- read.table("./Boreality.txt", header=TRUE)
```

To calculate the Boreality value we'll use for analyses, run the following line, creating a new variable, 'Bor'.

```
Boreality$Bor<-sqrt(1000*(Boreality$nBor+1)/(Boreality$nTot))
```

Running Mantel Tests

Though we have discussed the problems with Mantel tests in lecture, and they are well documented in the literature (e.g., Guillot and Rousset 2014), you may wish to explore the analyses with data.

As discussed in class, Mantel tests are based on distance or similarity matrices, characterizing physical distance between sample units, and similarity (or dissimilarity) between values for variables at among sample units. Thus, we will calculate these using the 'ecodist' package, designed for calculating and working with distance/similarity metrics.

We will calculate a geographic distance matrix for the dataset (based on the Pythagorean theorem), and then euclidean distance for 'Bor' and 'Wet' (simply the difference in values across sample units). More sites with more similar values have smaller distances. *For the assignment you will also need to calculate euclidean distance for the Temperature variable, 'T61', on your own.*

For characterizing differences based on multiple variables at once (representing a multivariate dissimilarity), you would likely want to use Mahalanobis distance - see this useful page on StackExchange: http://stats.stackexchange.com/questions/62092/bottom-to-top-explanation-of-the-mahalanobis-distance.

We will calculate distance/similarity metrics using the 'dist' function in base R (i.e., it is a default function of R), setting the method argument as 'euclidean'. You can also do this via the 'distance' function in the 'ecodist' package that we will use for the mantel test (which also allows calculation of Mahalanobis distance), but it is a bit slower in that package. For analysis of ecological communities (e.g., presence/absence and number of different species), a wide variety of similarity metrics are also available in the 'vegdist' function of the 'vegan' package.

```
geog.dist <- dist(Boreality[, c("x", "y")], method = "euclidean")
bor.dist <- dist(Boreality$Bor, method = "euclidean")
wet.dist <- dist(Boreality$Wet, method = "euclidean")
```

The function for calculating both regular and partial Mantel tests is 'mantel'. In the regular Mantel test, the main argument needed is the model being tested - basically, is one distance matrix related to another matrix (written as 'matrix1 ~ matrix2'). We can do this for all possible combinations here for exploratory purposes. Shown below is the code for testing relationships between geographic distance, similarity in boreal index, and similarity in wetness. As with other functions, it is worth checking out the possible arguments for 'mantel'.

```
mantel(bor.dist~geog.dist)
mantel(wet.dist~geog.dist)
mantel(bor.dist~wet.dist)
```

The output for each will look something like what is pasted below (the result is from 'mantel(bor.dist~geog.dist)'. The 'mantelr' value is the calculated mantel correlation between the two matrices; the first two p-values are for one-way tests (testing wither r is <= 0 or >= 0, respectively), while the third is for a two-way test (testing whether r is significantly different from expected in either direction) - generally I suggest focusing on that third one (pval3), as . The 'llim.2.5%' and 'ulim.97.5%' show the 95% confidence interval for the Mantel's r based on bootstrap resampling. This effectively provides an estimate of the potential values for r for a population, given that we typically only have a sample of data. These results show there is small but significant correlation (Mantel's r = 0.047, p=0.001) between the boreality values and the geographic distance between samples.

```
  mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
0.08336013 0.00100000 1.00000000 0.00100000 0.06979311 0.09794219
```

To carry out a partial Mantel test in the ecodist package, we simply add an extra term onto the formula - the matrix which we want to correct for. In this case, we might want to see if there's a significant effect of wetness on boreality, while controling for variation in space, so the code would be:

```
mantel(bor.dist~wet.dist+geog.dist)
```

The results are interpreted the same as above, but it should be understood that the effect of geographic distance is removed. Remember, Mantel tests have their problems, so you should check results with other techniques.

Spatial Regression

Note: R code below is adapted from the aforementioned book by Zuur et al.

In linear regression, one of the major assumptions is that measurements are 'independent and identically distributed' ('i.i.d.') - in other words, that all values within the population have the same probability of being sampled. Spatial autocorrelation violates this assumption, as samples close to one another have a higher probability of being similar in value. We can see this effect by conducting a simple linear regression with the data and looking at the residuals. If there is no spatial autocorrelation, the residuals (i.e., error from the model) will be randomly distributed in space, but if there is autocorrelation, the residuals will show spatial patterns. Here, we will focus on boreality as a function of wetness.

Setting up a Basic Linear Model and Visualizing Autocorrelation

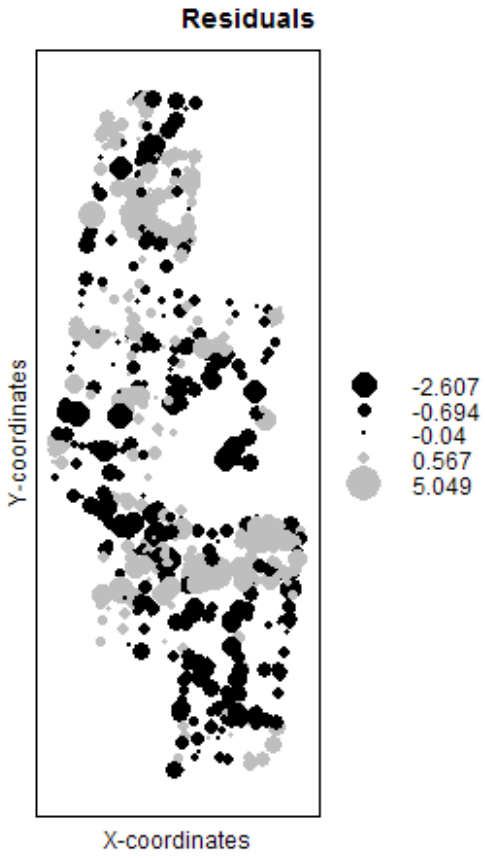
```
#Calculate the linear model
B.lm<-lm(Bor~Wet,data=Boreality)

#You can calculate the f-statistic and p-value for this using
anova(B.lm)
#Not necessarily desired; see:
# https://stat.ethz.ch/pipermail/r-help/2006-May/094765.html

#Extract standardized residuals (rstandard(LinearModel))
lm.resids <- rstandard(B.lm)

#Create SpatialPointsDataFrame of residuals and x/y coords
lm.resids.spdf<-data.frame(lm.resids,Boreality$x,Boreality$y)
coordinates(lm.resids.spdf)<-c("Boreality.x","Boreality.y")

#Create bubble plot (sp package)
bubble(resids.spdf,"resids",col=c("black","grey"),
       main="Residuals",xlab="X-coordinates",
       ylab="Y-coordinates")
```

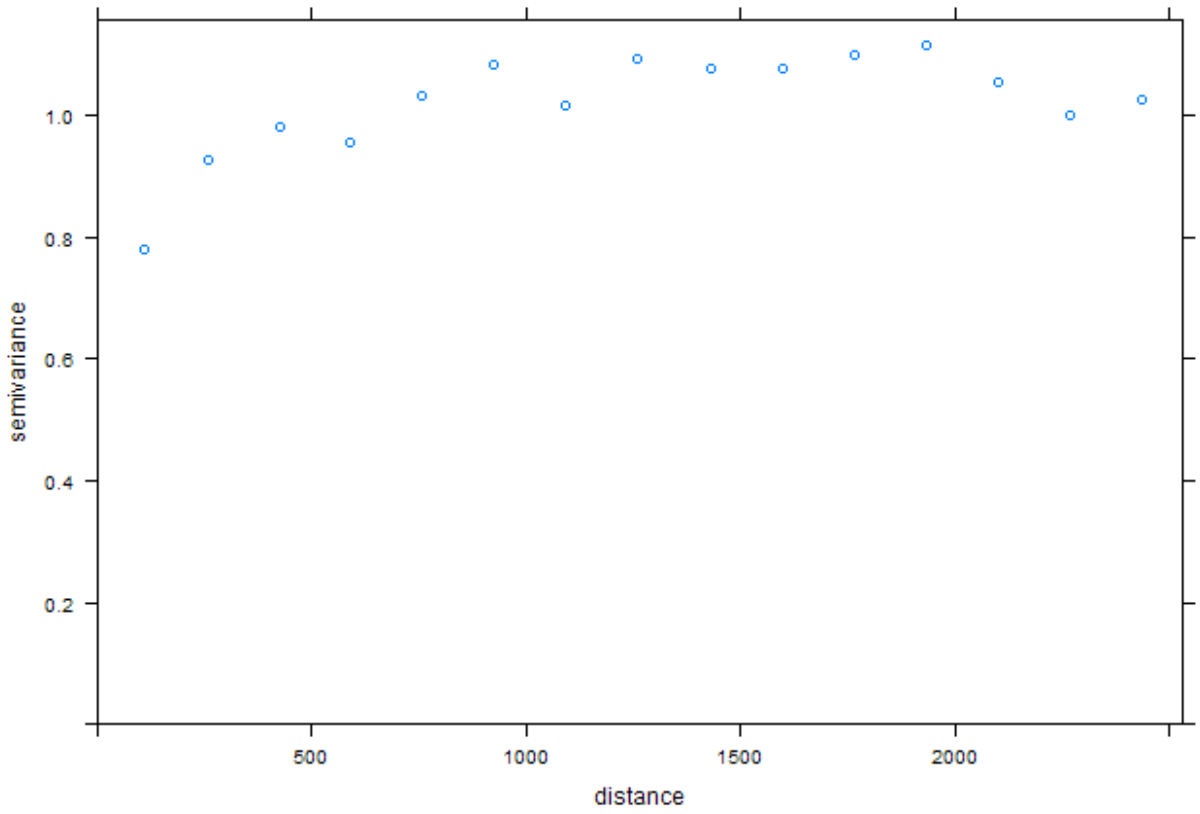


As you can see, there appears to be clustering of high and low residual values.

We can visualize and spatial autocorrelation in the residuals as we did last week using the 'pgirmess' package and the function 'correlog' (Look back to Lab 6)

We can also calculate and plot a variogram of the residuals, using the 'variogram' function in the 'gstat' package.

```
lm.residVario <- variogram(lm.resids ~ 1, lm.resids.spdf)
plot(lm.residVario)
```



You can see there seems to be an increasing semivariance with increasing distance up to ~500-1000 units.

Though the above variogram assumes isotropy, you can explore the variogram in different directions using this code:

```
plot(variogram(lm.resids ~ 1, lm.resids.spdf, alpha = c(0, 45, 90,135) ))
```

Correcting for Spatial Autocorrelation: Generalized Least Squares

So, how can we correct for this? One way that has become well developed and implemented in R via the 'nlme' package using the generalized least squares regression (function 'gls'), as discussed in lecture. This will let you control for spatially-associated error based on variogram functions fit to the data.

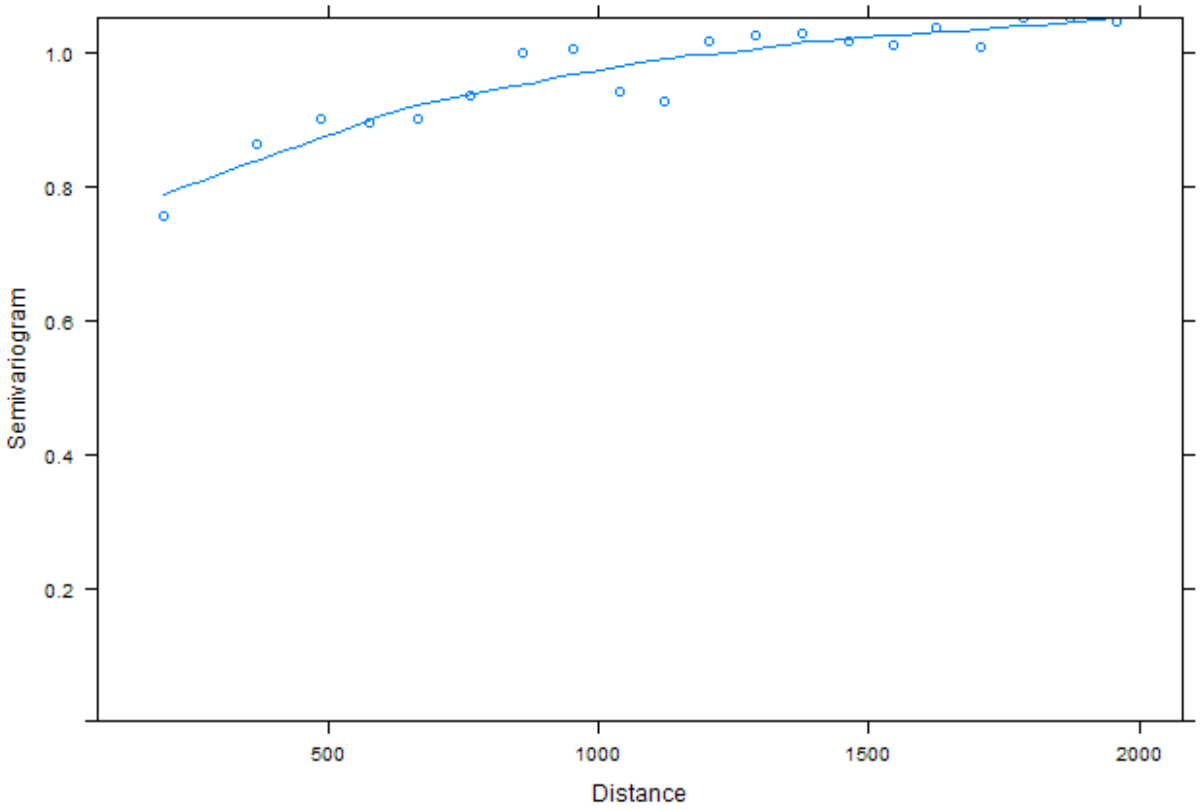
First, lets fit a basic gls model without taking into account the error structure.

```
# Don't forget to open the nlme package
library(nlme)

# We will assign the forumla for this model to 'f1' so we don't need ot keep
# retyping it
f1 <- formula(Bor ~ T61)

# Run the model using function 'gls'
B1.gls <- gls(f1, data = Boreality)
```

```
# this is a shortcut way to plot a variogram of the residuals from a gls
# model; resType='pearson' uses standardized residuals
plot(Variogram(B1.gls, form = ~x + y, robust = TRUE, maxDist = 2000, resType = "pearson"))
```



To add the spatial correlation structures, we can use add the 'correlation' argument to the nlme function. Herein, we can define the type of variogram model to use, indicate whether to use a nugget. To view the options, you can look up help for 'corClasses', and specifications for each option you can use help for each too. The main ones used for spatial autocorrelation are: corExp, corGaus, corLin, corRatio, and corSpher.

```
help(corClasses)

#show specifications for 'corSpher' option
help(corSpher)
```

It is not always clear which correlation structure is best to use, so you can run the model multiple ways and compare results based on fit metrics. These models are fit using maximum-likelihood estimators (MLEs), which is an iterative procedure. A good description of this is found at website for the statistical software, STATA: http://www.stata.com/support/faqs/statistics/convergence-of-maximum-likelihood-estimators/. Sometimes models will not converge with an MLE,and other estimators could be attempted or more iterations could be used; for now, if a model fails to converge, simply leave it out and move on.

Note: These models will take a minute or two to run, especially on older/slower computers.

```
B1Sph <- gls(f1,correlation=corSpher(form=~x+y,nugget=T),data=Boreality)
B1Lin <- gls(f1,correlation=corLin(form=~x+y,nugget=T),data=Boreality)
B1Rat <- gls(f1,correlation=corRatio(form=~x+y,nugget=T),data=Boreality)
```

```
B1Gau <- gls(f1,correlation=corGaus(form=~x+y,nugget=T),data=Boreality)
B1Exp <- gls(f1,correlation=corExp(form=~x+y,nugget=T),data=Boreality)
```

With the models run, we can select the best fit a few different ways. One of the most common ways is to use Information Theoretic metrics, like Akaike's Information Criterion (AIC) and Bayes' Information Criterion (BIC), which are metrics of how likely a model is given the data; since adding parameters to models will generally increase fit to the data, they penalize models with more parameters (e.g., variables or terms) in attempt to identify the most parsimonious model. The lower the AIC and BIC values, the better and more parsimonious the model; models with fit differences > 7 are considered to be considerably different, while models with fit differences < 2 are usually considered to be about equivalent.

It is important to keep in mind that for these metrics, models should have the same dependent variable and be based on the same sample of data. A great reference for this material is *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach* by Kenneth P. Burnham and David R. Anderson. Google around for other resources too, and check this paper: Aho, K., D. Derryberry, and T. Peterson. 2014. Model selection for ecologists: the worldviews of AIC and BIC. *Ecology* 95:631-636..

In this exercise we will only evaluate AIC values, and select the model with the lowest as our best fit. You can obtain this using the 'AIC' function on all of the models you wish to compare. (remove any that failed to run because of convergence issues).

```
AIC(B1.gls,B1Sph,B1Lin,B1Rat, B1Gau, B1Exp)
```

The B1Exp model has the lowest AIC - it has a much better fit than the model that does not take into account the correlation structure. We can look at the same plots as earlier to see how much of a difference this makes.

```
# Extract standardized residuals (rstandard(LinearModel)) This is a bit
# different than earlier, as the model is a 'gls' model from the nlme
# package
gls.resids <- residuals(B1Exp, type = "normalized")

# Create SpatialPointsDataFrame of residuals and x/y coords
gls.resids.spdf <- data.frame(gls.resids, Boreality$x, Boreality$y)
coordinates(gls.resids.spdf) <- c("Boreality.x", "Boreality.y")

# Create bubble plot (sp package)
bubble(gls.resids.spdf, "gls.resids", col = c("black", "grey"), main = "Residuals",
    xlab = "X-coordinates", ylab = "Y-coordinates")


# Create Moran Correlogram
library(pgirmess)
gls.moran <- correlog(coordinates(gls.resids.spdf), gls.resids.spdf$gls.resids,
    method = "Moran", nbclass = NULL)
plot(gls.moran)

# There still appears to be some significant autocorrelation, but it is
# greatly reduced (look at the values on the y-axis!).

# Look at the variogram of the residuals, normalized for the correlation
# structure
Vario1E <- Variogram(B1Exp, form = ~x + y, robust = TRUE, maxDist = 2000, resType = "normalized")
plot(Vario1E, smooth = FALSE)
```

Again, we can obtain a p-value for this via the anova command to evaluate whether there is an effect of wetness on boreality while controlling for spatial autocorrelation.

```
anova(B1Exp)
```

Notice, the F-value is considerably reduced compared to the prevoius model; the model is still significant, but it is a better fit model and the F-statistic is much closer to 0 (and closer to failing to reject the null hypothesis, that there is no effect of wetness on boreality).