Using R with ArcGIS to perform point clustering, classification, and density estimation using the *MCLUST* package

Before proceeding to the example, you must have the following installed on your computer:

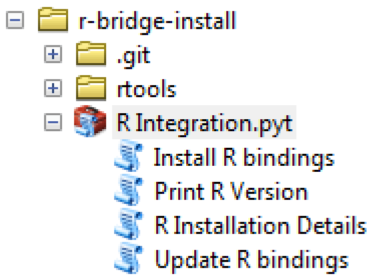
Prerequisites

1. ArcGIS 10.3.1 or ArcGIS Pro 1.1 (don't have it? try a 60 day trial)
2. R Statistical Computing Software, 3.1.0 or later
    - 32-bit version required for ArcMap, 64-bit version required for ArcGIS Pro (Note: the installer installs both by default).
    - 64-bit version can be used with ArcMap by installing Background Geoprocessing and configuring scripts to run in the background.
3. R ArcGIS Bridge

Setup Instructions

ArcGIS 10.3.1

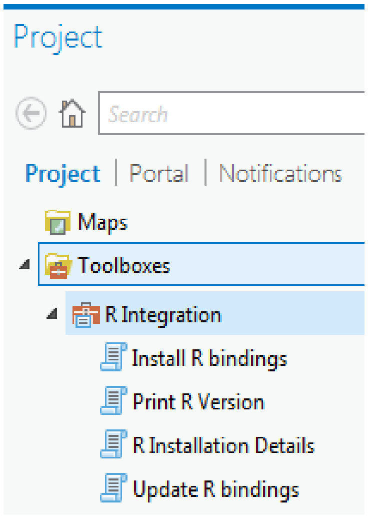
- In the Catalog window, nagivate to the folder containing the Python Toolbox, `R Integration.pyt`. *Note*: You may have to first add a folder connection to the location that you extracted the files or downloaded via Git.
- Open the toolbox, which should look like this:



- Run the `Install R bindings` script. You can then test that the bridge is able to see your R installation by running the `Print R Version` and `R Installation Details` tools, and running the included sample tools.

ArcGIS Pro 1.1

- In the Project pane, either navigate to a folder connection containing the Python toolbox, or right click on *Toolboxes* > *Add Toolbox* and navigate to the location of the Python toolbox.
- Open the toolbox, which should look like this:



- Run the `Install R bindings` script. You can then test that the bridge is able to see your R installation by running the `Print R Version` and `R Installation Details` tools, and running the included sample tools.

Background information

Cluster analysis helps to identify groups of points on the map that are separated geographically from other groups of points. Cluster analysis can be used in a variety of applications such as:

- Cholera outbreaks clustered around water wells (as in the first application of GIS by Dr. John Snow in the 1850s), or
- A cluster of petty crimes that occurred in the past week in a particular city.

The fundamental idea is that if many points are located close together, they are likely related in some way, and we say that they are part of the same cluster.
Many clustering algorithms have been developed over the years. For example, the simplest clustering algorithm, nearest neighbor clustering, works by repeatedly finding the two nearest pairs of points and combines them into a group. The process stops when the required number of groups is reached or the distance between the remaining pair of points becomes larger than a specified distance. This and many other clustering algorithms do not involve any statistical measure, and they do not perform well when clusters are not well separated.

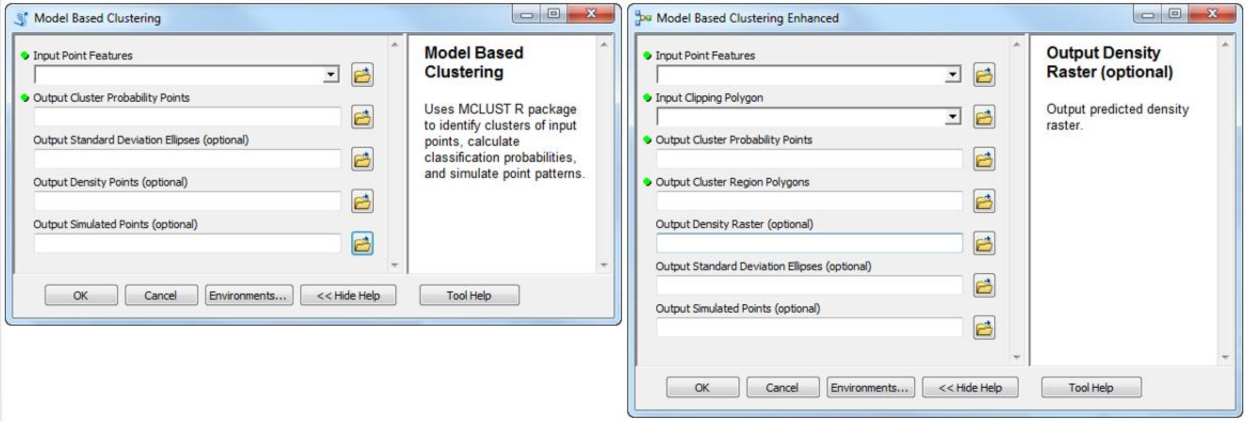
By using a statistical clustering method, we can ask additional questions such as:

- What is the optimal number of clusters?
- How likely is it that a point belongs to a particular cluster?
- How should we deal with outliers and noise?
- Which cluster model parameters should be used?

Probabilistic model-based clustering implemented in the *mclust* package [2] written by Chris Fraley and Adrian Raftery addresses these issues by assuming that the observed data are a mixture of several different populations with different ellipsoidal areas, shapes, orientations and possible noise. The goal is to identify and separate these different populations into individual clusters. The technique also provides the probability that each point falls into a particular cluster. The algorithm works by creating many candidate models, and the final model is chosen using the Bayesian information criterion (BIC).

Understanding the Model Based Cluster tools

The Model Based Clustering tools analyze the spatial locations of the points in order to identify spatial clusters. Because the model predictions are only reliable in areas that are completely surrounded by points, a clipping polygon is required by the enhanced Model Based Clustering tool and analysis will only be performed within this clipping polygon.


Model Based Clustering
Input Point Features
Output Cluster Probability Points
Output Standard Deviation Ellipses (optional)
Output Density Points (optional)
Output Simulated Points (optional)
OK
Cancel
Environments...
<< Hide Help
Tool Help
Model Based Clustering Enhanced
Input Point Features
Input Clipping Polygon
Output Cluster Probability Points
Output Cluster Region Polygons
Output Density Raster (optional)
Output Standard Deviation Ellipses (optional)
Output Simulated Points (optional)
OK
Cancel
Environments...
<< Hide Help
Tool Help

Model Based Clustering

Uses MCLUST R package to identify clusters of input points, calculate classification probabilities, and simulate point patterns.

Output Density Raster (optional)

Output predicted density raster.

Figure 1. The Model Based Clustering and Model Based Clustering Enhanced tools.

These tools both call the *mclust* R package, however, the enhanced version has some additional parameters. The different parameters are briefly described below;

Input Points Features: The point feature class with observed point locations.
Input Clipping Polygon: A polygon border around the observed point locations. Analysis will only be performed within this polygon feature class (Enhanced version only).
Output Cluster Probability Points: Output point feature class that contains the probability that each point belongs to each of the identified clusters.
Output Standard Deviation Ellipses: The standard deviational ellipse for the regions of each identified cluster. These are useful for quantifying the likely boundaries of each identified cluster (not in Enhanced version).
Output Density Points: Output gridded point feature class with the predicted density at each point (not in Enhanced version).
Output Simulated Points: Output point feature class that is simulated form the fitted clustering model. If the model performs well, these simulated points should closely resemble the input point features.
Output Cluster Region Polygons: Output polygon feature class that partitions the clipping polygon into regions based on the classification of the most likely cluster. This output shows the estimated regions of each identified cluster (Enhanced version only).
Output Density Raster: Output predicted density raster (Enhanced version only).

Case study

The location of a tree in the forest depends on many factors, such as the positions of other trees, soil characteristics, slope, and past forest management practices. Forest stands are usually heterogeneous, and their characteristics vary in space.

Ideally, forest researchers would take a census of every tree in a forest, but manual collection of the tree characteristics in the entire forest is expensive and nearly impossible. The limited number of samples that can be collected manually is rarely sufficient to reliably summarize the characteristics of the forest. Instead, researchers often identify small areas of the forest that they believe are representative of the entire forest. These small areas are exhaustively studied, and from them many inferences are made about the entire forest. Other small, representative areas are then studied to see if these inferences hold up. This methodology can be used, for example, to estimate the total number of trees in the forest and estimate the sustainable yield of the forest.

Cluster analysis is a small but important component of this process. The number of tree clusters, their orientations, sizes, and associated uncertainties are all interesting results that can be used to better understand the characteristics of the entire forest. Once a clustering model is generated, it can then be used to simulate tree patterns that have the same general structure as the trees in the study area, and these simulations will hopefully represent how trees cluster together in areas of the forest that were not studied.

The data for this example comes from an application of aerial photo interpretation to tree species identification. Figure 2 illustrates the process of individual tree identification from combined LIDAR and multispectral airborne imagery. A semiautomatic method allowed researchers to recognize nearly all dead trees, and about 89 percent of the living spruce trees.

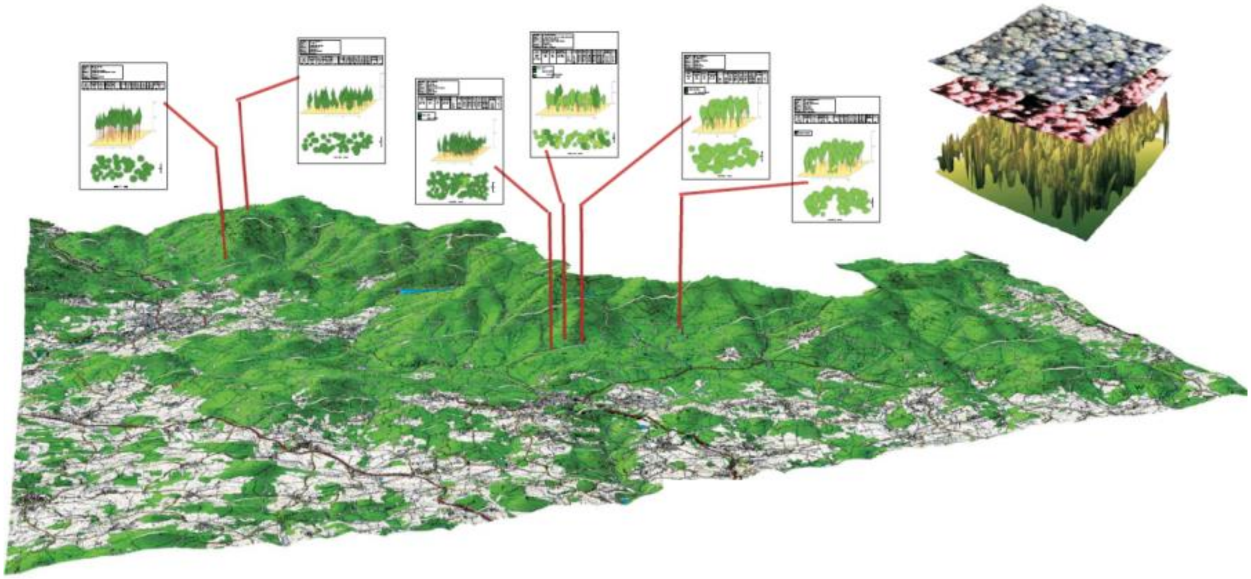


Figure 2. The individual tree identification from combined LIDAR and multispectral airborne imagery [4].

Figure 3 shows living spruce trees collected in the Bavarian Forest National Park using the above approach. These points are used as input to the Model Based Clustering tools. The clipping polygon was created by buffering the tree locations.

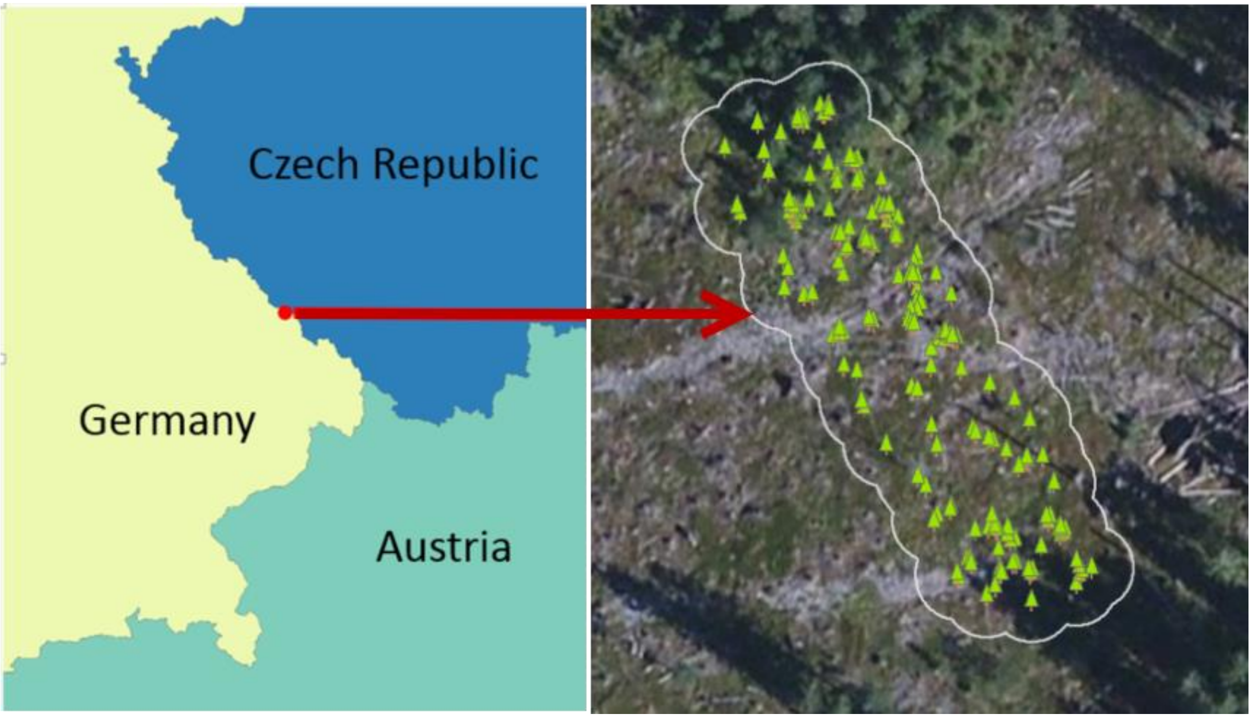


Figure 3. Locations of spruce trees in a small area within the Bavarian Forest National Park [4].

Some of the output from the Model Based Clustering tools are illustrated in Figure 4.

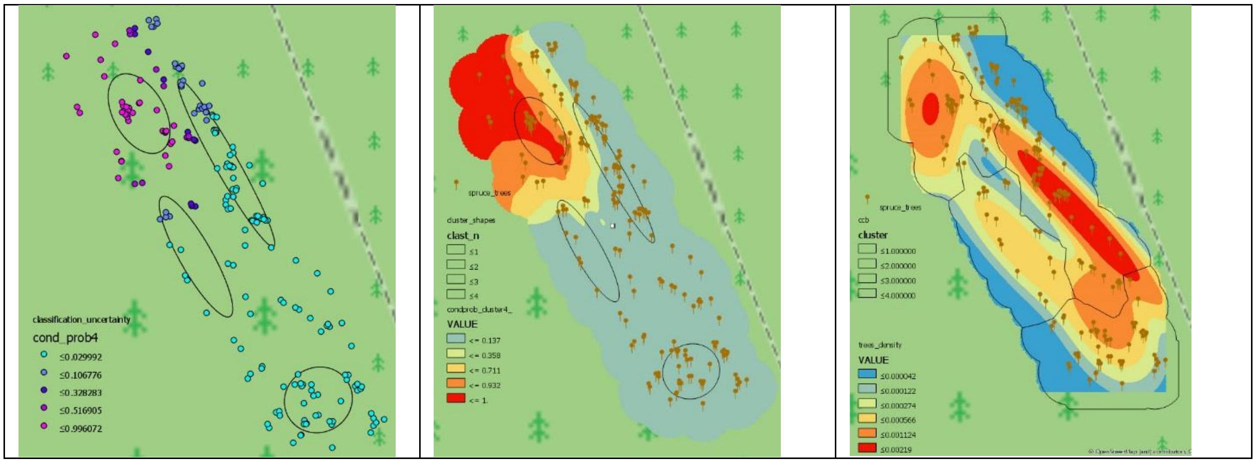


Figure 4. Outputs from the Model Based Clustering tool. Left: probability that each tree falls into the fourth cluster (analogous maps could be created for the other three clusters), and the standard deviational ellipses of each cluster (black ellipses). Center: The ellipses and the conditional

probability that points belong to one particular cluster (created by interpolating the probabilities). Right: density of the points and the estimated regions of each cluster (black lines).

The main use of the output maps is exploratory analysis of forest tree locations, including comparison with typical and well-studied tree patterns.

Figure 5 shows additional output from the Model Based Clustering tools, three simulations of the tree locations and their densities produced by running the tool with *Output Simulated Points* instead of the actual trees locations. Note that since the tree clustering model is probabilistic, the fitted model each time generates a different number of trees.

If the model was fitted effectively, we may find such or very similar trees distributions in the nearby areas of the forest that were not exhaustively studied, and these simulations can be used to plan for management of the entire forest.

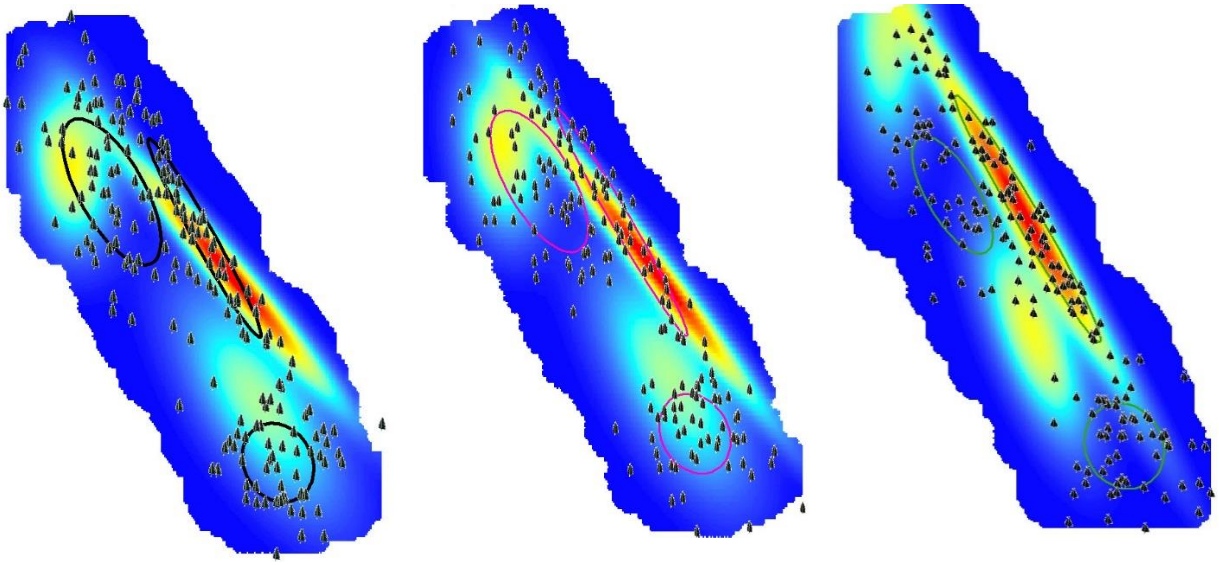


Figure 5. Three simulations of the tree locations and the densities of the simulated trees.

Further reading

[1] Fraley, C. and Raftery, A.E. (2003). Enhanced model-based clustering, density estimation and discriminant analysis software: MCLUST. Journal of Classification, 20, 263-286.
[2] Fraley C. and Raftery A.E. (2007). Model-based methods of classification: Using the mclust software in chemometrics. *Journal of Statistical Software*, 18, paper i06.
[3] Koch B., Heyder U., and Weinacker H. (2006) Detection of Individual Tree Crowns in Airborne Lidar Data. Photogrammetric Engineering & Remote Sensing, Vol. 72, No. 4, pp. 357–363.
[4] Krivoruchko, K. (2011) *Spatial Statistical Data Analysis for GIS Users*. ESRI Press, 928 pp:

- Cluster detection methods in regional data, chapter 11.
- Cluster analysis, chapter 13.

- Cluster analysis using the mclust02 package, chapter 16.
- Point pattern analysis in forestry, chapter 6.