The Implementation of Autocorrelation-Based Regioclassification in ArcMap Using ArcObjects

Christoph MAYRHOFER

Abstract

Conventional methods for cartographic classification are often solely based on underlying attributive values. There are numerous algorithms to determine the resulting classes, such as *Jenks Optimal* classification, but they still do not account for the spatial patterns that are inherent to spatial data. This can cause a visual disruption of areas that would normally be considered a cluster, thus making the overall message of a map harder to grasp. With a method called "*Autocorrelation-Based Regioclassification"*. TRAUN & LOIDL (2012) introduced an alternative approach that takes spatial properties into account and classifies data values in respect to their statistical and spatial properties. This paper builds upon their method and shows how their approach has been implemented in ArcMap as an Add-In. Additionally, some improvements to the original method are described as well as a method that allows the representation of overlapping classes, which result from the spatial classification.

1 Introduction

Conventional classification methods that assign polygons to a certain class by their data value often result in visually fragmented maps. In order to produce logically consistent maps that allow identifying the general patterns of spatial data, it is beneficial to have contiguous polygons of very similar values to be within a common class. This results in rather compact objects (visual clusters). The crisp class boundaries of attributive classifications however, do not account for spatial proximity. Therefore, it is possible that the map representation of a "statistical cluster" with adjacent polygons that have values close to a class break will be very fragmented and does not result in a visual cluster.

Autocorrelation-Based Regioclassification reduces this problem since it also accounts for the spatial distribution of data values. Polygons that are adjacent and have similar values, but are yet in different classes will be adjusted by this method to be within a common visual class. This method is only applicable to choropleth maps with metric data and can be used to emphasize the general patterns of spatial data. The reduction of visual complexity avoids excessively fragmented patterns and thereby improves the interpretability of maps.

This section briefly describes the method introduced by TRAUN & LOIDL (2012): In addition to the attributive component, which is normally used to classify data, they include a spatial component. The amount of spatial autocorrelation (i.e. the statistical relation of values and their neighboring values) is used to determine the degree to which the

Jekel, T., Car, A., Strobl, J. & Griesebner, G. (Eds.) (2012): GI_Forum 2012: Geovizualisation, Society and Learning. © Herbert Wichmann Verlag, VDE VERLAG GMBH, Berlin/Offenbach. ISBN 978-3-87907-521-8.

spatial component should influence the classification process. If the dataset is characterized by values that are generally surrounded by similar values, it inherits a high spatial autocorrelation and the spatial component will be weighted stronger. If the values in the dataset are distributed completely arbitrarily in a spatial sense, the classification will be solely based on the attributive values as it would be with a conventional classification method. The approach is self-calibrating, since the weighting between the spatial and attributive components is directly derived from the data itself.

The statistical measure that is used to determine the spatial autocorrelation is called Moran's I and will be thoroughly explained in section 3.1. This measure can be calculated for each feature and will then be called Local Moran's I. The mean of all these values is defined as the Global Moran's I, which is equivalent to the regression line of coordinate points that are defined by the local value as the x-axis and the mean of the surrounding values as the y-axis. Fig. 1 shows a Scatterplot with the mentioned setup. The slope of the thick dashed diagonal line represents the Global Moran's I. The vertical dotted lines represent the class breaks of a conventional, non-spatial classification.

The data points will now be orthogonally projected on the regression line as described in section 3.2 and the resulting frequency distribution on the regression line is used to obtain the spatial classes (diagonal dotted lines). Therefore, all data points that are within one of the diagonal class areas will be assigned to the same visual class (= color).

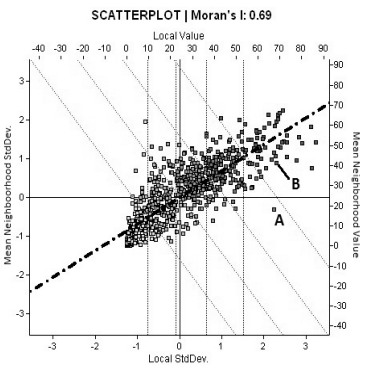


Fig. 1: Scatterplot

Example: Point A and B have similar local (own) values and are both within the highest class (>53) when a non-spatial optimal classification is applied. With the spatial approach, A will only be assigned to the second highest class (the diagonal line above A is the break to the highest class), while B remains in the highest class. This is caused by the fact, that B has an above average value (local value) which is represented by its x-coordinate and is surrounded by other high values (neighborhood value), which are averaged to define its y-coordinate. A, on the other hand, has a high local value, but is surrounded by low neighboring values, which results in a class "downgrade" using the spatial method.

Fig. 1 also shows that A will now be in a lower class than B, while other features – even with a lower attributive value – remain in, or are upgraded to the highest class. This results in overlapping classes, which is a concept that is not yet well established and often even opposed by many cartographers who insist on the principle of non-overlapping classes.

In order to satisfy the need to clearly associate each feature with a distinct value range, TRAUN & LOIDL (2012) suggest the addition of another visualization layer for the value domain. While the spatially classified colors of the polygons result in a smoother visual appearance of the map, small labels indicate the non-overlapping distinct classes that the polygon actually belongs to. This method is described in section 3.5. The resulting map of this approach is illustrated in Fig.12 at the end of this paper.

2 Methods and Sample Data

The program has been developed as an ESRI ArcMap Add-In, which allows users to extend the functionality of any ArcMap installation with the new classification approach.

There are three main components involved in the development process:

- *An Integrated Development Environment (IDE):* This software is used to write and compile the source code. Microsoft Visual Studio 2008 was used for this project.
- *ArcObjects:* ESRI provides developers with the "ArcObjects SDK" (Software Development Kit). ArcObjects is a collection of components which allow control of all ArcGIS functionalities (e.g. attribute manipulation, classification, processing tools). There are numerous, well documented interfaces included. The main source of information about each of these is the ESRI Online Help (ESRI 2012a).
- *The programming language:* The integrated Python environment of ArcMap could not be used, since it does not allow advanced user interface customization. (PENNSYLVANIA STATE UNIVERSITY 2012). Therefore, it is necessary to switch to a more sophisticated programming language that is able to use ArcObjects, such as VB.NET, C++, C# or Java. C# was used for this project due to personal preference.

The same dataset as used in the paper by TRAUN & LOIDL (2012) was used in order to directly compare the programmatically created statistics with their results. The dataset features the percentage of African American people in 755 counties of the southeast United States as reported by the US Census Bureau in 2000.

3 Application Architecture

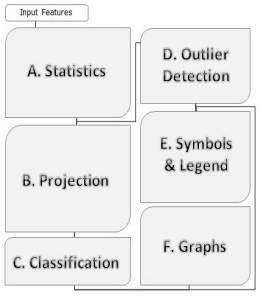


Fig. 2: Module Structure

The application can be divided into six distinct modules. Fig. 2 shows the module structure and their dependency. Each section of this chapter elaborates on the functionality of one of these modules.

Module A, B and C are exactly implemented as described in TRAUN & LOIDL (2012). Module D adds some additional statistical analysis as proposed in the discussion and outlook section of their paper.

Module E describes how the new concept of overlapping classes can be implemented in ArcMap by using a special algorithm in combination with non-overlapping categorical classification. Module F illustrates a workaround that allows to create complex graphs by using layer stacks.

3.1 Module A: Statistics

The first module calculates the local Moran's I (I_i), which indicates the spatial autocorrelation of a local value (i.e. percentage of Afro Americans in a certain county) and its neighbors (e.g. all adjacent features). It is defined as (ANSELIN 1995):

$$I_i = \frac{x_i - \overline{X}}{\sigma_i^2} \sum_{j=1, j\neq i}^{n} \omega_{i,j}(x_i - \overline{X}), \quad \text{with} \quad \sigma_i^2 = \frac{\sum_{j=1, j\neq i}^{n} \omega_{i,j}}{n-1} - \overline{X}^2$$

In order to understand and to implement this equation it can be split into its components:

σ_i^2 is the variance among the values of all counties excluding the one that the I_i is currently being calculated for. As a first step, the variance can be expressed as $(\sigma_i^2 = \sigma \cdot \sigma)$ by expanding σ_i^2 to both factors of the multiplication.

$$I_i = \frac{x_i - \overline{X}}{\sigma} \cdot \frac{\sum_{j=1, j\neq i}^{n} \omega_{i,j}(x_i - \overline{X})}{\sigma}$$

$\frac{x_i - \overline{X}}{\sigma}$ is the local value expressed in standard deviations and will be represented as "x" in all of the following flow charts, such as in Fig. 3 calculates the sum of values in respect to a weight matrix. This matrix describes how much each feature (i.e. neighboring county) influences the I_i of the currently calculated feature. In case of a first order neighborhood (contiguity), each adjacent county will have the same influence assigned in the weight matrix, and all other counties will be neglected (weight = 0). $j \neq i$ assures that the local value itself is excluded from this calculation.

Therefore, $\frac{\sum_{j=1, j\neq i}^{n} \omega_{i,j}(x_i - \overline{X})}{\sigma}$ is the mean value of the neighbors expressed in standard deviations. This component will be "y" in the flowcharts.

$$\sum_{j=1, j\neq i}^{n} \omega_{i,j}(...)$$

Fig. 3 shows how the described equations are implemented in the program:

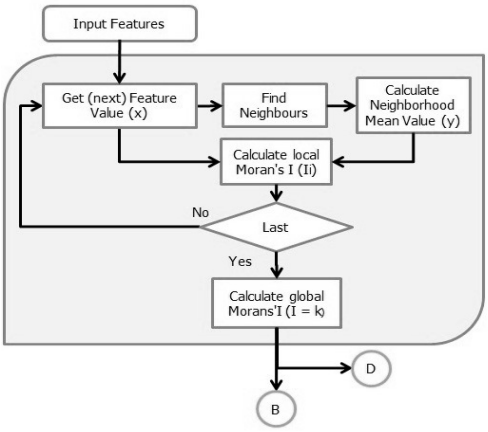
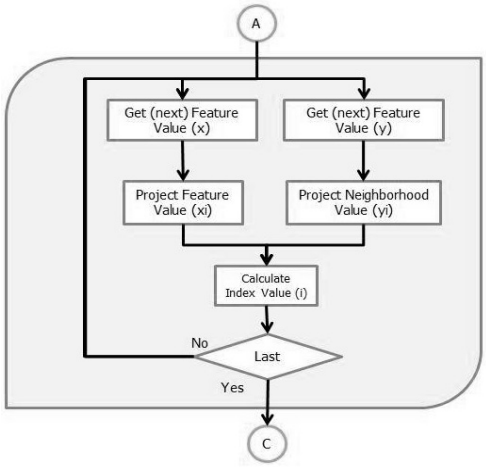


Fig. 3: Module A

Note: All fields that mention the word value (e.g. Get next Feature Value) refer to the value expressed as standard deviations and not to the actual number. All features (counties) will be looped in order to calculate the local Moran's *I*. The *x* component of the above equations is already known. The neighbors of the county will be identified with the help of spatial filters. Their values are then used to calculate the *y* component, which is finally multiplied with *x* to determine I_i. After the local Moran's *I* has been calculated for every county, it is possible to determine the global Moran's *I*, which is defined as the mean of all I_i (HARRIS & JARVIS 2011).

3.2 Module B: Projection



The coordinate points x and y can now be projected onto the global regression line as shown in Fig. 4:

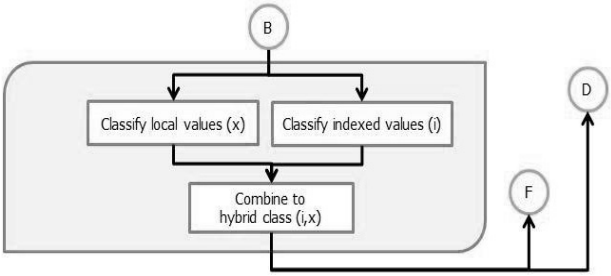
$$xi = \frac{\mathrm{k}y_i + x_i}{1+k^2} \quad yi = \frac{\mathrm{k}(\mathrm{k}y_i + x_i)}{1+k^2}$$

Finally each feature will be assigned an index value which reflects its distance from the origin along the regression line. This is accomplished by using the Pythagorean Theorem.

Fig. 4: Module B

3.3 Module C: Classification

This module classifies the original attributive values and the respective index value that is derived in module B. The user interface (cp. section 4) allows users to choose from different classification methods (e.g. Jenks, quantile, standard deviation). However, Jenks Optimal Classification has been chosen as the default since an optimal classification method generally produces the best result for multimodal data distributions (JENKS1977).



Each feature will then be assigned a non-spatial class which depends on its x value and a spatial-class dependent upon its i value. Both classes are finally combined to a hybrid class (e.g. i-class:3, x-class: 4, hybrid-class: 34).

Fig. 5: Module C

In case that there are more than nine classes, leading zeros will be added to avoid confusions (e.g. hybrid class 119 could be interpreted as i:11/x:9 or i:1/x:19, but class 0119 is distinct).

3.4 Module D: Outlier Detection

One of the major downturns of the classification approach to this point is the disregard of significant outliers. Thus, the visual class of a county with a significantly low value would

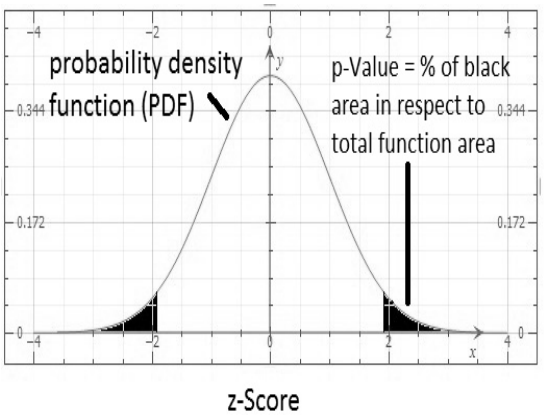
be upgraded if it is surrounded by other counties with high values. While this result is welcome for most features in order to create visually smoother maps, it also reduces the important contrast between these significant outliers and their neighbourhood, making it harder to spot them. *"The basic method described herein is especially useful where the main interest lies in the communication of general spatial patterns. It is not the method of choice when the map's objective is to highlight spatial outliers – polygons that differ significantly from their neighbors ..."* (TRAUN & LOIDL 2012, p. 14). They propose to include further statistical analysis to identify and exclude the outliers as a measure to solve this problem. ArcMap provides a tool that is built upon the work of ANSELIN (1995) on cluster and outlier analysis. It can be found in the ArcMap Toolbox (Spatial Statistics Tools → Mapping Clusters → Cluster and Outlier Analysis). The tool is programmed in Python and also uses the local Moran's I. The calculation of this figure as shown in module A accounts for the biggest part of the overall processing time. Therefore, the speed of the developed program would drastically decrease, if the provided ArcMap tool had been used for the outlier detection. Thus, the functionality of the python code was completely reprogrammed in C# using the equations provided by ESRI (2012b) and then altered to directly use the local Moran's I values from module A. The method of outlier detection as proposed by ANSELIN (1995) classifies features as outliers according to the following principles:

Z-scores are a common measure to determine statistical significance. They are calculated for the local Moran's I of each feature and are a mere representation of the features in standard deviations:

$$z_{I_i} = \frac{I_i - \mu}{\sigma}$$

The graph in Fig. 6 shows a probability density function (PDF), which reflects the normal distribution and is defined as:

$$y = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$



The function can be used to calculate the p-value of each county. A p-value quantifies the probability that a county will lie within a certain value range. The z-score of each county is used as the x-value. The black area in Fig. 6 represents the p-value of a county with a z-score for the I_i of 1.96. This distinct threshold is used since it results in a p-value of 0.05 = 5%. This is a commonly used level to declare a value as statistically significant (HARRIS & JARVIS 2011/EBDON 1985).

Fig. 6: Probability density function

The p-values can be calculated by integrating the PDF over the interval defined by z:

$$p(z) = 1 - \int_{-z}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz$$

Unfortunately it is not possible to directly implement integrals in program code, since integrals require infinite precision. Integration can be described as the process of dividing the area under the function into small rectangles, which are then used to calculate the area. The smaller the rectangles are, the more precise the area can be quantified.

One way to calculate the area programmatically would be to use finite sizes for the rectangles and to manually define the number of segments used to approximate the area. This approach requires a significant amount of calculation steps when high precision is desired.

ESRI (2012a) states that their spatial outlier tool calculates p-values by *"numerical approximation"*, but they do not reveal which exact method is used to approximate the area. There are numerous functions available with varying complexity to result in different levels of precision. Since 0.05 has been chosen as the threshold level, an approximation with a precision to the third decimal place is sufficient (i.e. max. error of 10^{-3}). ABRAMOWITZ & STEGUN (1964) provide several numerical approximations for the PDF. Their function 26.2.16 is the least complex (= fastest calculation) and still exceeds the necessary precision by a factor of 100. It has therefore been implemented in the code:

$$p(z) = 1 - 2\phi(z)\,(a_1 t + a_2 t^2 + a_3 t^3) + \varepsilon(z)$$

$$\text{with} \quad \phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} \quad t = \frac{1}{1 + pz} \quad |\varepsilon(z)| < 10^{-5}$$

$$a_1 = 0.4361836 \;\; a_2 = -0.1201676 \;\; a_3 = 0.937298 \;\; p = 0.33267$$

The p-value can now be used in combination with the local Moran's I to identify outliers. A county will be marked as an outlier if it is characterized by a negative I_i in combination with statistical significance ($p < 0.05$).
The I_i criterion makes sure that only counties which contrast their neighbours (e.g. low value, surrounded by high values) are considered as an outlier. The p-value criterion evaluates the statistical significance of that constellation.

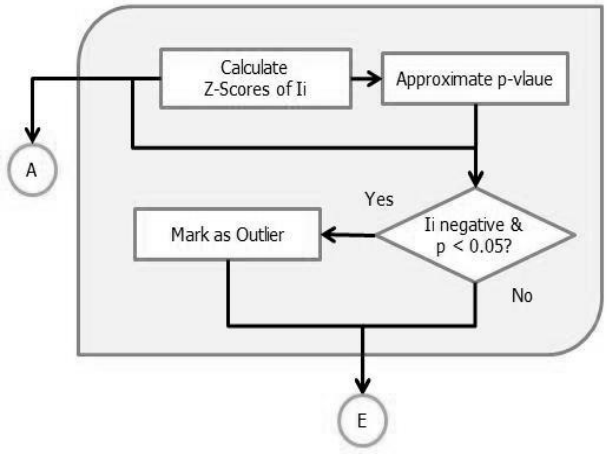


Fig. 7:
Module D

3.5 Module E: Symbols & Legend

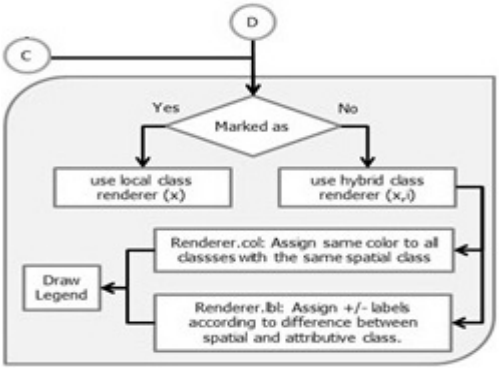


Fig. 8: Module E

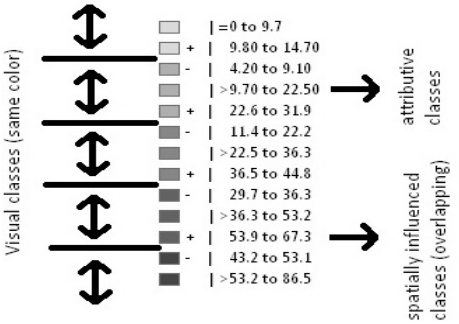


Fig. 9: Legend

Since ArcMap is not genuinely designed to support overlapping classes, it is one of the major challenges to implement the visualization concept. In order to achieve a satisfactory result with the provided tools, it was necessary to develop an algorithm that assigns the colors and labels to each feature and class manually:

Firstly, all features are sorted by their hybrid class. This makes sure that all features with the same spatial class are grouped and sorted by their attributive class within these groups. Then, all features will be looped and every hybrid class that occurs for the first time will be added to the class renderer (the component that later visualizes the features according to certain rules). All hybrid classes within the same group (same spatial component) will be assigned the same color.

This results in the class setup as depicted in Fig. 9. Additionally, all classes that overlap are supplemented by labels that indicate the difference between the spatial and attributive class (e.g. x-class:5, i-class:4 → label: +). This indicates that the value within that spatial (color) class would normally be in a higher class.

3.6 Module F: Graphs

Additional tools are in need to support the understanding of the underlying methods, since this approach is rather new and has not yet been established as a common classification concept. According to TRAUN & LOIDL (2012), a scatterplot and a histogram series may be used to assist in the visual interpretation of the classification results. Unfortunately, the graphing functionality of ArcMap relies strictly on the underlying data. Each object that needs to be drawn must be defined by a "data source" (ESRI 2012c). There is no interface to define independent objects (e.g. a line that is defined by a function rather than by data points). It is necessary to create separate tables as a data source for each object that should be drawn in the graph, in order to be able to create more complex graphs such as needed in this case. These tables include the class intervals that will later be represented as semi-transparent polygons, class breaks, which then will be dotted lines and independent objects (e.g. coordinate cross, regression line). All these tables are then combined to a layer stack when represented in the graph. The result of this workaround is illustrated in Fig. 11.

4 Graphical User Interface (GUI)

A significant amount of the development effort was made to design an intuitive GUI that allows users – including those who are not very familiar with the concept of this method – to apply spatial classification.

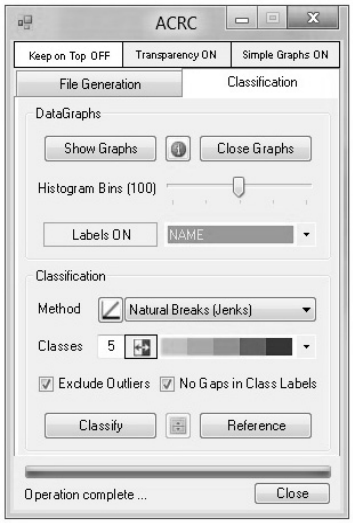


Fig. 10: GUI, Controls

The controls are separated into two sections. The first set of controls is used to calculate the statistics and invokes modules A and D. Since the calculation of bigger datasets (>1000 features) can take several minutes, it is possible to save to the calculated statistics or to load a previously created statistics file. The second set of controls (Fig. 10) is used to interact with the graphs and classification parameters.

Fig. 11 shows the interface that is used to investigate the classification result. The three windows are displayed simultaneously and the underlying data values are linked. This allows direct interaction with the graphs. It is possible to select features on the map which will then be highlighted in the graphs. The same accounts for the selection of features from the graphs, which will then be highlighted on the map. This compilation of the three sources of information supports the interpretation of the resulting maps as proposed by TRAUN & LOIDL (2012).

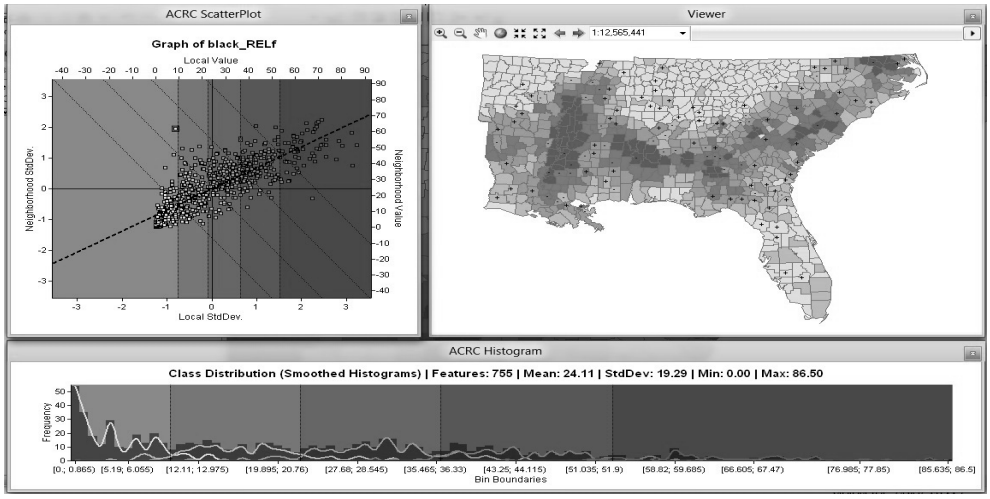


Fig. 11: GUI Visualization

5 Conclusion and Outlook

The development of this tool has shown that the unconventional approach of spatial classification and the idea of overlapping classes can be implemented in a software environment (ArcMap) that is not genuinely designed to support these concepts. Spatially aware classification has proven to reduce the visual complexity of maps which allows users to identify the overall patterns of a map more easily as Fig. 12 demonstrates. The classification results are less fragmented than with a non-spatial classification, while outliers remain unaltered and become even more apparent due to the increased contrast within the overall smoother map appearance.

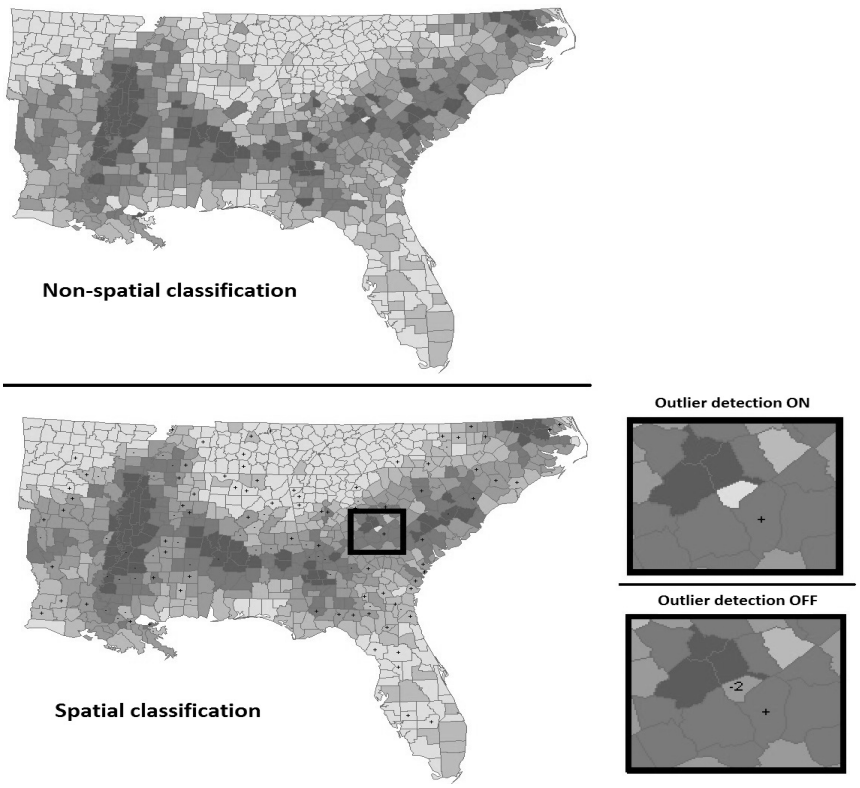


Fig. 12: Comparison of a non-spatial classification (Jenks) and the classification result of the developed tool. Additionally, the bottom right images compare the classification with the implemented outlier detection to the original method.

The improvement compared to a Jenks Optimal Classification can also be quantified by MACEACHREN'S (1982) "complexity index for choropleth maps". This measure represents the ratio of polygon edges that serve as class boundaries in respect to the overall number of polygon edges. The dataset in this case study consists of 1310 class boundary edges / 2289 total edges (index = 0.57) when Jenks classification is applied, while only 1105 class

boundary edges result from Autocorrelation-Based Regioclassification (index = 0.48). The consideration of outlier analysis – as suggested by TRAUN & LOIDL (2012) – could be implemented and eliminates one of the major issues of their spatial classification approach.

However, the outlier analysis only results in a visual exclusion of the concerning features from the spatial classification. The values of the features themselves still influence the classification of their neighbours. This method has been used in respect to calculation speed. Nevertheless, further research needs to be done to evaluate the justification of weighting speed over (minor) improvements of the statistical accuracy.

The tool offers settings to change the concept of neighbourhood and the classification method. While the original method uses a first order contiguity in combination with Jenks Optimal Classification (which are the default settings), it is now possible to use any other combination (e.g. IDW and Equal Interval), which results in derivative classification concepts. However, the results of other settings have not been statistically analyzed, which is also subject to further research.

References

ABRAMOWITZ, M. & STEGUN, I. (Eds.) (1964), Handbook of Mathematical Functions, with Formulas, Graphs and Tables, Probability functions (chapter 26), p. 932. National Bureau of Standards. New York, Dover.

ANSELIN, L. (1995), Local Indicators of Spatial Association – Lisa. Geographical Analysis, 27 (2), 93-115. Ohio State University Press

EBDON, D. (1985), Statistics in Geography. 2nd Edition, Blackwell Publishing, Oxford, 150-163.

ESRI (2012a), ArcObjects Online Help for the .NET Framework. http://help.arcgis.com/en/sdk/10.0/arcobjects_net/ao_home.html (22.01.2012).

ESRI (2012b), How Cluster and Outlier Analysis: Anselin Local Moran's I works. http://resources.esri.com/help/9.3/ArcGISDesktop/com/GP_ToolRef/spatial_statistics_tools/how_cluster_and_outlier_analysis_colon_anselin_local_moran_s_i_spatial_statistics_works.htm (05.01.2012).

ESRI (2012c), ArcObjects Online Help ISeriesProperties Interface. http://help.arcgis.com/en/sdk/10.0/arcobjects_net/componenthelp/index.html#/ISeriesProperties_Interface/001200000qp2000000 (12.01.2012).

HARRIS, R. & JARVIS, C. (2011), Statistics for Geography and Environmental Science. Chapter 9, Exploring Spatial Relationships. Prentice Hall, New Jersey, 241-248.

JENKS, G. F. (1977), Optimal data classification for choropleth maps. Lawrence: University of Kansas.

MACEACHREN, A. M. (1982), Map complexity: comparison and measurement. The American Carographer, 9 (1), 31-46.

PENNSYLVANIA STATE UNIVERSITY (2012), GIS Programming and Automation. Chapter 4.7, Limitations of Python scripting with ArcGIS. https://www.e-education.psu.edu/geog485/node/152 (15.01.2012).

TRAUN, C. & LOIDL, M. (2012), Autocorrelation-Based Regioclassification – A self-calibrating classification approach for choropleth maps explicitly considering spatial autocorrelation. International Journal of Geographical Information Science, iFirst, 1-17.