Dr. Eick

COSC 6335*“Data Mining”*

ProblemSet2 Fall 2020

Outlier Detection and Collocation Mining

Deadlines; Task 5 is due Sunday, October 25, 2020 at 11p

Task 6 is due Thursday, November 5 at 11p

Last updated: October 27, 7p

*Task 5: Design, Implementation and Evaluation of an Outlier Detection Techniques*

*for a Spatial Dataset (Individual Task; not peer reviewed)*



Complex8\_RN15 Data Set Visualization

The Complex8\_RN15 available at: <https://drive.google.com/open?id=1_geQDIMQUNHhc3d7zRxfOrOreTT3HBhU>

The goal of this task is to design and implement a bivariate, spatial outlier detection techniques of your own preference. You will apply the technique of your choice to the 2-dimensional dataset called *Complex8\_RN15*[[1]](#footnote-1), is the variation of the Complex8 dataset with 15% Gaussian noise added to the original Complex8 dataset which can be found at: <http://www2.cs.uh.edu/~ceick/DM/Complex8.csv>; however, the original dataset should only be used for visualization purposes. Finally, you will evaluate your technique and write a short report that contains a description of your outlier detection method, experimental results (see below), and your evaluation of the approach you developed. Your outlier detection technique should take the dataset and create a copy of the dataset that contains an additional column/attribute called *ols (“outlier score”)* which contains numbers that indicates how much your outlier detection method believes that the particular object is an outlier—the smaller the value of the ols attribute the less likely the object is believed to be an outlier. For the project, you can use any R-library or any other software library to accomplish the project tasks; just acknowledge what external software you used in your report.

The Complex8\_RN15 dataset has attributes x, y, class; for example, after applying your outlier detection technique to 5 examples of the dataset the result produced by the method you are supposed to implement could look as follows:

563.225,56.748,0,0.24

564.887,58.119,0,0.41

565.434,68.061,0,0.12

565.926,79.953,0,0.33

566.762,69.405,0,0.11

This depicted result indicates that the second example is the most likely outlier, the fourth example is the second most likely outlier,…, and that the last example is the least likely outlier.

**Task5 Sub-tasks:**

Task a: Visualize the Complex8\_RN15 dataset; visualize the third attribute using different colors, like supervised scatterplots we used in ProblemSet1. 3 pts

Task b: Design and implement an outlier detection techniques for the Complex8\_RN15 dataset. Maybe enhance the techniques based on feedback obtained in Task c! As explained earlier your implementation should add a column/attribute *ols* to the dataset and fill this column with numbers. 18 pts

Task c: Evaluation[[2]](#footnote-2)

a. Apply our outlier detection to the Complex8\_RN15 dataset obtaining a new file X; your outlier detection method is only applied to attributes x and y of the dataset, and ignores the attribute named class and adds an ols-column to the dataset. 2 pts

b. Sort X in descending order based on the values of attribute ols (the example with the highest ols value/the example that is the most likely outlier should be the first entry in X)! 1 pts

c. Visualize the first 7% of the observations in X, just displaying their x and y value and the class using a different color for each class, in a display and the remaining 93% of the observations in a second display. In general, the first display visualizes the outliers and the second display visualizes the normal observations in the dataset. 1pts

d. Visualize the first 14% of the observations in X, just displaying the x and y value and the class using a different color for each class and the remaining 86% of the observations in a second display. 1pts

e. Visualize the first 21% of the observations in X…remaining 79%... 1pts

f Interpret the 6 displays you generated in steps c-e; particularly, assess how well does your outlier detection method worked—intuitively observations that are quite far a way of the 8 natural clusters of the original Complex8 dataset should be outliers. Also try to characterize which points are picked as outliers as the first 7%, 14%, and 21%, respectively 9 pts

g. Create a histogram for the ols values of the top 21% entries in file X. Briefly, interpret the obtained histogram! 5 pts

Task d: Write a report which contains a 2-5 paragraph summary, describing how your outlier detection technique works and how it was implemented. Include the results of tasks a and c in the report. Also include if you enhanced your approach based on feedback to get better results a brief description of what you enhanced. If your outlier detection technique needs the selection of parameter values before it can be run, describe how you selected those parameter values. Moreover, mention in an additional paragraph what (if any) external software packages your used in the project! 12 pts

There will also 8 pts allocated for the quality of your outlier removal technique and up to 6 extra pts for very sophisticated approaches.

Submit the code of the implementation of your outlier detection techniques in a separate file! Finally, we only want one project report (either in Word or pdf format)! Please upload your submission on Blackboard and name the submission file as T5\_<student last name>. Be prepared to demo your outlier detection technique for Task c, if requested.

Task6: Association Analysis—Finding Collocation Patterns Involving Building Types in a City

Peer-Reviewed Group Project

Very Preliminary Draft

This project centers on mining collocation patterns in Point of Interest (POI) Datasets where examples are assumed to have the form: <longitude, latitude, category or class>. For example, the City of Chicago has such a POI dataset that contains every building in Chicago which can be found at: <https://data.cityofchicago.org/Buildings/Building-Footprints-current-/hz9b-7nh8> Figure 1 visualizes such a building dataset that contains the buildings and building type information of three neighborhoods of a city named after the lost city of *Zinj[[3]](#footnote-3)*. The Zinj.kml dataset can be found at: <http://www2.cs.uh.edu/~ceick/DM/Zinj.kml>

The dataset is a kml file that can be opened using Google Earth[[4]](#footnote-4), Notepad, and probably using ArcGIS; the dataset likely needs some preprocessing, extracting the relevant information, before it can be used for Task6!



Fig. 1: Example Building Dataset.

In particular, this task centers on analyzing collocation patterns with respect to the following six[[5]](#footnote-5) building types in the dataset: single house, garage, commercial building, light buildings, collective house, and schools. Questions we might be interested to answer include:

1. Are buildings randomly distributed or is there some clustering?
2. Are buildings of the same building type collocated, anti-collocated or randomly distributed?
3. Are building belonging to different building types collocated, anti-collocated or are their locations unrelated—for example, you will try to answer the question if garages are collocated with commercial buildings.

https://www.youtube.com/watch?v=hiGT\_0badNI

The main objective of Task6 is to give conclusive answers to questions 2 and 3—many suitable approaches exist to accomplish this goal; consequently, I expect a lot of variation with respect to methods different groups apply to answer the last two questions.

In this project, we consider commercial buildings and collective houses as *primary building types* and light buildings, single houses, garages, and schools as *secondary building types*. You will analyze question 2 only for primary building types, and question 3 only for collocation pattern involving primary building types with other building types (could be primary or secondary). You might analyze the distance characteristics of these associations using K-functions[[6]](#footnote-6) and other curves or statistics[[7]](#footnote-7) that are constructed by computing k-nearest neighbor distances and/or using techniques you developed on your own. You might also create a dataset Z[[8]](#footnote-8) in which buildings are placed at random—but preserving the building type distribution in Zinj—and apply your developed methods to both Z and Zinj to assess if building type locations are collocated, random, or anti-collocated. Another additional challenge is the large size of the dataset Zinj and the large complexity of creating building type distance distribution curves. Be creative in proposing sampling or approximate computing approaches, if creating the necessary summaries turns out to be too time consuming! Normalization of analysis results to make them comparable is a another challenge of the project.

Summarize the methods you used and the project results in (3-5 single-spaced pages) report. Also describe the software you produced in the project in an appendix, and submit the source code as an additional attachment. You should also be able to demo the methods you used or developed to answer questions 2 and 3.

Remark: Possibly the mentioned dataset Zinj will be replaced by another dataset and the dataset specific parts of the Task6 specification will be updated accordingly. However, the main objectives and goals of Task6 will remain the same!

1. The outlier detection method is only applied to the first 2 numerical attributes and the third attribute is just used for visualization purposes; for that reason, it is called a spatial outlier detection method; e.g. it can detect outliers for object locations described by (longitude, latitude)-pairs. [↑](#footnote-ref-1)
2. No other evaluations are required! [↑](#footnote-ref-2)
3. The lost city of Zinj is part of the movie Congo: <https://archive.org/details/Congo_The_Movie_Lost_City_Of_Zinj_1996_U> [https://en.wikipedia.org/wiki/Congo\_(film)](https://en.wikipedia.org/wiki/Congo_%28film%29) [↑](#footnote-ref-3)
4. Install Google Earth and then just open the Zinj.kml file and you should see the houses in the dataset. [↑](#footnote-ref-4)
5. The dataset Zinj also contains buildings belonging to other building types which you will ignore in your analysis. [↑](#footnote-ref-5)
6. See for example: <http://dhf.ddc.moph.go.th/abstract/s22.pdf> or [http://wiki.landscapetoolbox.org/doku.php/spatial\_analysis\_methods:ripley\_s\_k\_and\_pair\_correlation\_function](http://wiki.landscapetoolbox.org/doku.php/spatial_analysis_methods%3Aripley_s_k_and_pair_correlation_function); finally there is an R-library *Spatstat* which contains useful techniques for Task6: <https://www.rdocumentation.org/packages/spatstat/versions/1.56-1/topics/spatstat-package> [↑](#footnote-ref-6)
7. Moran’s I-function is such a statistics: <http://svitsrv25.epfl.ch/R-doc/library/ape/html/MoranI.html> all those are in R: <http://rgm2.lab.nig.ac.jp/RGM2/func.php?rd_id=GeoXp:moranplotmap> [↑](#footnote-ref-7)
8. There is actually a real *lost city* named *Z* see: <http://www.top10listing.net/travel/top-10-lost-cities> [↑](#footnote-ref-8)