PUI2015 Extra Credit Project Proposal

Graffiti Spots Grouping<Jianhao Zhou, zhoujh30, N15871313>

I. Abstract

The study is to explore possible python approaches to optimizing current GIS steps that group graffiti cleaning spots in five boroughs. Scripts including an algorithm that can be used in the future by DSNY every two weeks to efficiently group a list of graffiti cleaning spots have been created but need further improvements.

II. Introduction

The DSNY exports a list of graffiti spots (around 450) to be cleaned every two weeks.

Current Steps (using GIS):

1. *Data Preparation (X/Y coordinates)*
2. *Network Calculation (Traveling Salesman Problem)*
3. *Manual Grouping (10 per group, separate borough spots)*
4. *Final Edits (rename boroughs & columns, combine address, export to excel)*

Current problems: time-consuming, subjective grouping, unnecessary network calculation

The study is to explore possible python approaches to optimizing current GIS steps that group graffiti cleaning spots in five boroughs.

III. Data

1. Biweekly graffiti cleaning data from DSNY (I got through my internship with EDC and use 20151014_Cleaning_DownloadSites.csv available in the folder for this analysis)

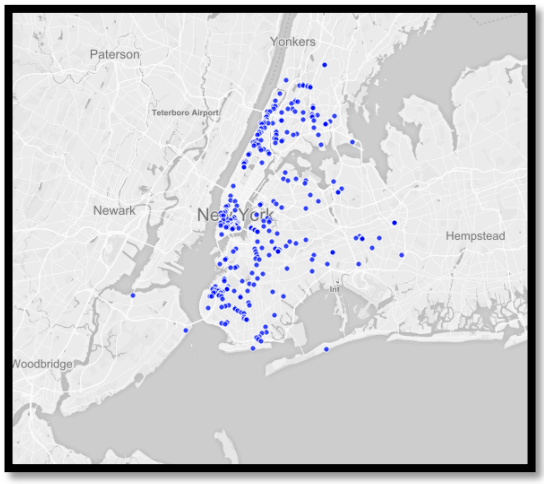

Fig1. Plot of All Graffiti Cleaning Spots in 20151014_Cleaning_DownloadSites.csv

2. NYPD Motor Vehicle Collisions data from NYC Open Data (I saved here https://db.tt/eRhRzCLi due to its large size)

III. Methodology

1. Converting x and y coordinates into latitudes and longitude

A. The conversion was conducted using pyproj package and epsg 2263 projection.

B. The conversion was for later analysis and plotting on a basemap using mplleaflet pakage.

C. The pyproj package can be installed typing ($ conda install -c https://conda.anaconda.org/jjhelmus pyproj) and mplleaflet be installed by typing ($ pip install mplleaflet) in the terminal.

2. Adding traffic accidents to create weighted distance

A. I calculated total level of injuries for each zip code by adding NUMBER OF PERSONS INJURED and three times of NUMBER OF PERSONS KILLED. The standard is arguable and thus temporary for this analysis.

B. I create a factor for each zip code using the equation *total.iloc[i] = 1 + ((total.iloc [i] – mean) / 5 / std)* to balance the weighted distance.

C. The final calculated weighted distance is an NxN matrix that cosidered the combined effect of the origin and destination zip code for point n and m.

3. Grouping by Kmeans and agglomerative clustering

A. The Kmeans and agglomerative clustering was used to group the graffiti cleaning spots into (total number of spots)/10 clusters because the ultimate goal is to group them into

clusters of equal size of 10. This was not achieved through this analysis but will be explored with further efforts.

B. I used a plenty of approaches in your notebook [1] for the Kmeans and agglomerative clustering here.

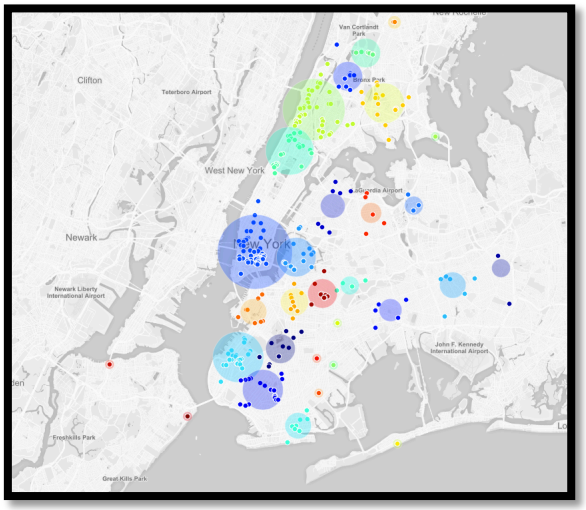

Fig1. Plot of Graffiti Cleaning Spots Using Agglomerative Clustering

IV. Conclusions

Scripts including an algorithm that can be used in the future by DSNY every two weeks to efficiently group a list of graffiti cleaning spots have been created but need further improvements.

V. Future work

1. Grouping by equal size clustering (using expectation–maximization algorithm for clustering and create extra height parameter to control cluster size)
2. Adding more weighted factors that influence the distance such as real road network distance or project emergency and difficulty.

VI. Links

Please find it in the same folder.

VII. Bibliography

[1] https://github.com/fedhere/UInotebooks/blob/master/cluster/thanksgivingClustering.ipynb

[2] http://scikit-learn.org/stable/modules/clustering.html

this is very good work, bus seen as i am familiar with it i would have liked more details on the new stuff the distance metric with traffic.
the clustering with constraints is very interesting and i look forward to hear how it works