NETWORK SCIENCE ANALYTICS APPLIED TO NYC CRIME

Final Project

CLUZEL, Arnaud
arnaud.cluzel@student-cs.fr

ERBANNI, Rebecca
rebecca.erbanni@student-cs.fr

VIAL, Jennifer
jennifer.vial@student-cs.fr

BOURIAL, Sarah
sarah.bourial@student-cs.fr

VEERAMANEN, Jithendrasai
jithendrasai.veeramaneni@student-cs.fr

January 2019

Abstract

This paper focuses on the study of NYC crime records and the identification of network patterns within the city boroughs, using graph networks, visualisation tools and prediction algorithms.

1 Introduction

This project focuses on the network structure of gangs, group-level processes of geographical activity, and applications of social network analysis to violence prevention work. It explains how social network analysis can impact the study of the geographical dimensions of gang space. Specifically, we aim at analysing the gang network in the city of New York. Indeed, gangs in New York constitute an increasing issue: in march 2018, 22 murders in the city were considered to have been caused by gang rivalry. This number keeps on increasing especially because of the internet-war known as cyber-banging (conflicts do also happen on social net-works nowadays, e.g. Twitter). Hence, analysing their network would allow us to have an overview of the links between the members and understand how they operate in order to identify the presence of established gangs and new collaborations.

2 Related Literature

Researchers paid increasing attention to the study of criminal behavior dynamics, in a two-fold approach: from a people and a place centric perspective. The people-centric perspective has mostly been used for individual or collective criminal profiling. Wang et al. (2013) suggested Series Finder, a machine learning approach to the problem of detecting specific patterns in crimes that are committed by the same offender or group of offenders. Short et.al. (2008) presented a biased random walk model established upon empirical knowledge of criminal offenders' behavior along with spatio-temporal crime information to take into account repeating patterns in historical crime data. Furthermore, Ratcliffe (2006) investigated the spatio-temporal constraints underlying supsects' criminal behavior. An example of a place-centric perspective is crime hotspot detection and analysis and the consequent derivation of useful insights. More recently, the proliferation of social media has sparked interest in using this kind of data to predict a variety of variables, including electoral outcomes (Tumasjan, 2010) and market trends (Bollen et.al., 2011). In this line, Wang et al. (2012) proposed the usage of social media to predict criminal incidents. Their approach relies on a semantic analysis of tweets using natural language processing along with spatio-temporal information derived from neighborhood demographic data and the tweets metadata. In our project, we tackle the crime hotspot identification problem by analyzing socio-spacial aspects of gangs structures and activities in New York city, and through this community building exercise draw conclusions related to the improvement of violence prevention work.

Our work would hence complement the above-mentioned research efforts and contribute to the place centric research effort in quantitative criminal studies.

3 Data

The dataset comes from NYC OpenData.(link) It shows all the crimes reported to the New York City Police Department for the year 2018. We can see that the dataset has 228,905 observations and 35 features including localization, crime committed, time of the event, and the suspects profile (ethnicity and age).

3.1 Data Cleaning

Our dataset did have some missing values: approximate date times and/or missing GPS location. The latter was only an issue when generating a new feature for neighborhoods (more on that below) as there were 269 missing neighborhood values: sexual crimes for which latitude and longitude was not given, and crimes happening in between neighborhoods. These were replaced with values `'no neighborhood [BOROUGH]'`.

3.2 Features Engineering

3.2.1 Neighborhoods

We wanted to add pricision to our network by creating a `neighborhood` feature describing the exact neighborhood a given crime took place in. To retrieve the name of the neighborhood from the latitude and longitude of the neighborhood we used the Mapbox Api. Mapbox Api is a REST (Representational State Transfer) Api which lets us extract Map (geospatial) related data. We found the request url needed to extract the name of the neighborhood on this Stack OverFlow post. We adopted the Javascript code obtained from Stack OverFlow to a Python code and we finally read the latitude and longitude from each crime, retrieved the name of its neighborhood and stored them in a `.csv` file for further use.

3.2.2 Shortest path

Shortest path algorithms are a family of algorithms designed to solve the shortest path problem i.e. find the shortest path between any two given vertices s and t. This basic approach is to rank pairs of papers (x, y) by the length of their shortest path in graph G . Such a measure follows the notion that collaboration networks are 'small worlds', in which individuals are related through short chains. In terms of gangs here, a node with the shortest path becomes one of greater importance within the bigger network of gangs in the city. This variable did not actually improve our results, so was not included in our final models.

3.2.3 Centrality Measures

Our decision to incorporate centrality measures was based on literature as Xu and Chen (2005) mention in their paper 'Criminal Network Analysis and Visualization' that:

> *Centrality deals with the roles of individuals in a network. Several centrality measures, such as degree, betweenness, and closeness can suggest the importance of a node in a network. The degree of a particular node is its number of links; its betweenness is the number of geodesics (shortest paths between any two nodes) passing through it; and its closeness is the sum of all the geodesics between the particular node and every other node in the network. An individual's having a high degree, for instance, may imply his leadership; whereas an individual with a high betweenness may be a gatekeeper in the network.* (page 106)

Degree centrality
The degree of a vertex of a graph is the number of edges incident to the vertex, with loops counted twice. Degree centrality is the number of ties that a node has and can be interpreted in terms of the immediate risk of a node for catching whatever is flowing through the network - here, the occurrence of a type of crime. The degree centrality of a vertex v, for a given graph $G := (V, E)$ with $|V|$ vertices and $|E|$ edges, is defined as $C_D(v) = \deg(v)$.

Eigenvector centrality
Eigenvector centrality is a measure of the influence of a node in a network. It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the

score of the node in question than equal connections to low-scoring nodes. For a given graph $G := (V, E)$ with $|V|$ vertices, let $A = (a_{v,t})$ be the adjacency matrix, i.e. $a_{v,t} = 1$ if vertex v is linked to vertex t, and $a_{v,t} = 0$ otherwise. The relative centrality score of vertex v can be defined as: $x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{v,t} x_t$ where $M(v)$ is a set of the neighbors of v and λ is a constant.

Betweenness centrality
Betweenness centrality quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. It was introduced as a measure for quantifying the control of a human on the communication between other humans in a social network (in our case, replace human by gang). In this conception, vertices that have a high probability to occur on a randomly chosen shortest path between two randomly chosen vertices have a high betweenness. The computation is as follows:

$$Betweeness(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where, σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that goes through v.

3.3 Data Visualisation

3.3.1 Map

We used the folium library to build an interactive map plotting every single record of a given crime type in its specific location. Folium is a powerful Python library that helps to create several types of Leaflet maps. The goal here was to create a map with location markers that are clustered (`clusteredmarker = True`) if close together. The tileset used in here is OpenStreetMap which is default. Link to example map on harassment crime can be downloaded here and opened for visualisation locally. As one can notice, crime occurence tends to be clustered at various levels (state, region etc), the lowest one being in different neighborhoods depending on the crime type. We thereby decided to study our dataset from a geographical perspective, focusing on the location of concentrated activity relating to every crime.

3.3.2 GIF

The aim here was to look for a way to destroy gang networks though NYC neighborhoods using disease to avoid crime propagation and network cascade. We built a GIF to picture our targeted attack of the neighborhood gangs network. (see 4.1.2) The .gif file can be downloaded here.
The metric we used here was betweeness centrality because (as seen above) it is suitable to describe gang networks: indeed some nodes have higher centrality, translating a more prominent association with other gangs in the network.
From the neighborhood proximity graph we build, we attack 20% of the nodes that are in the Giant Connected Component (where all the intertwined gangs sit). Every time the node with highest betweeness centrality is removed and the resulting graph is printed until we aggregated all the network alterations to make a .gif file. An interesting observation was that as nodes and the edges connecting them to the GCC were deleted, we can see the emergence of separate sub-clusters which correspond to some specific boroughs for e.g. Staten Island-based gangs gets isolated by this process.

4 Models

4.1 Network Graphs

4.1.1 Bipartite Graph

Our initial thought was to build a bipartite network to connect all crimes to their location (neighborhood). The result was fairly condensed and lacked interpretability, hence pushing us to try out alternatives.

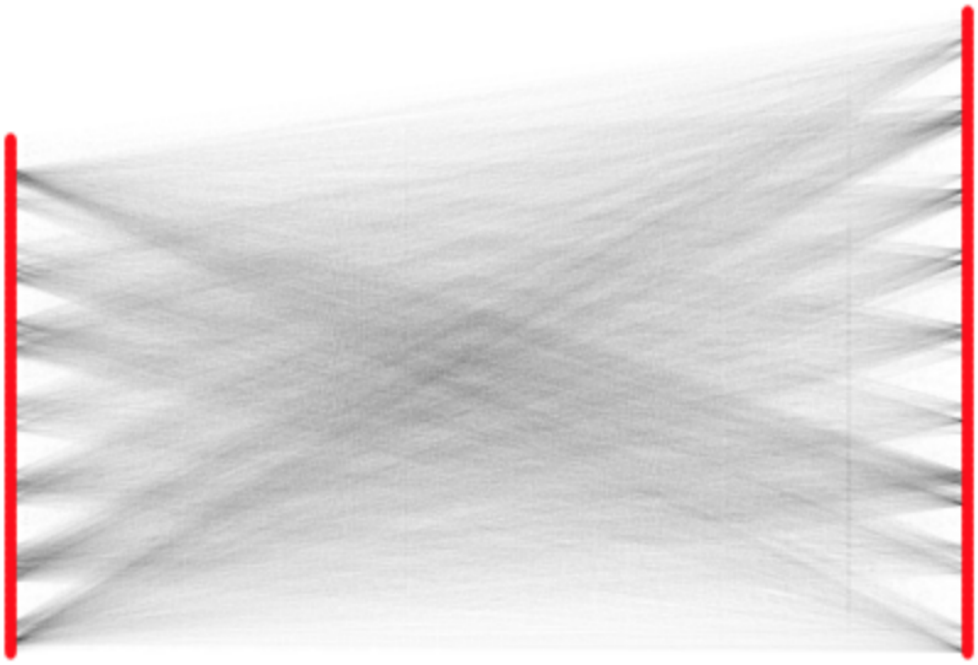

Figure 1: Crime - location bipartite graph

4.1.2 Neighborhood Proximity Graph

We here applied the definition of proximity graphs: these are simply graphs in which two vertices are connected by an edge if and only if the vertices satisfy particular geometric requirement i.e. a particular spatial distance. In our setting, we considered two neighborhoods (nodes) to be connected whenever the same level of crimes (all crimes included) occurred in both neighborhoods.

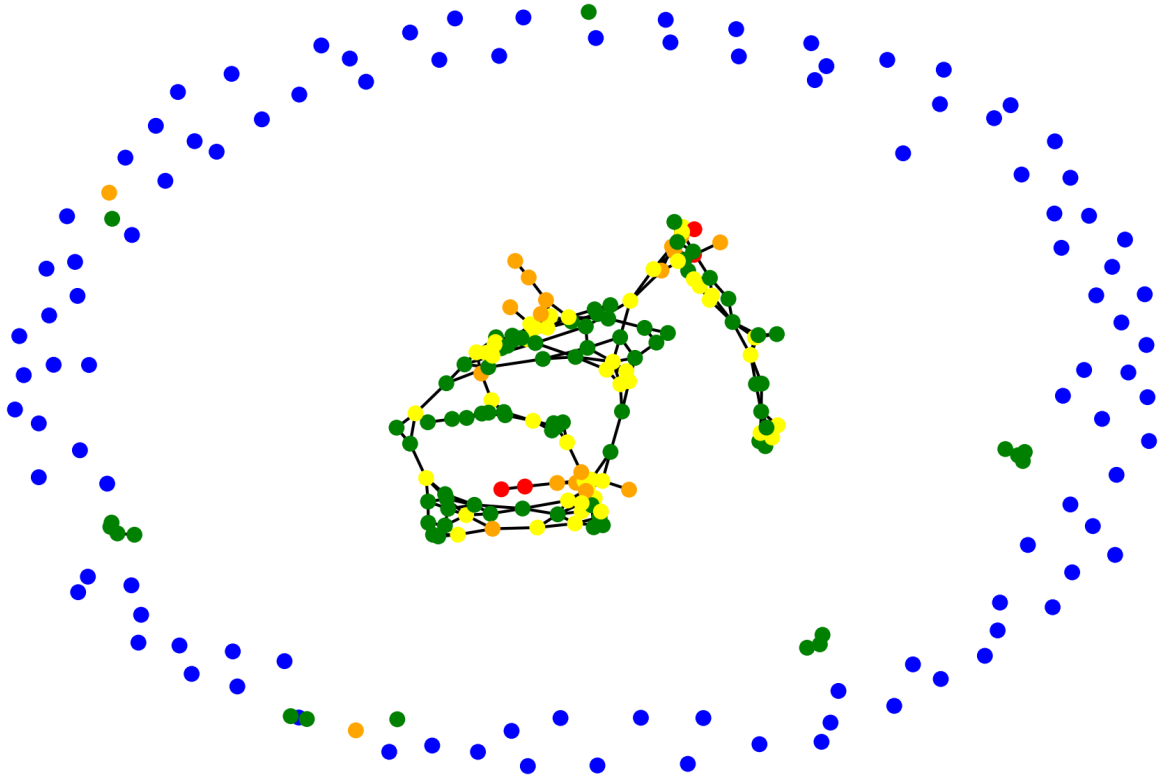

Figure 2: Neighborhood proximity graph for all crimes combined

The outside area in blue depicts what we deem to be 'safe zones' i.e. neighborhoods with low overall crime rates.

4.1.3 Type of Crime Graph

Similarly, we here created a graph based on the type of crime mostly occuring in each neighborhood i.e. crimes are the nodes and neighborhoods the edges. Similarly to our map, we can see how some types of crimes tend to concentrate in a given area, thereby translating the potential presence of an organised crime association.

Figure 3: Type of Crime Graph

4.1.4 Gang Graph

We here attempted to infer a graph translating the NYC gang network according to every gang's location from the two previous graphs i.e. using the neighborhood proximity of all crimes and the concentration of every crime within a neighborhood. Two nodes were connected when both were thought to share a gang in their streets.

4.1.5 Modularity Graph

Just like clusters, modularity is a heuristic to measure clustering, scaled between -1 and 1. The difference is that it looks at edge densities in given clusters/communities/groups compared to edge densities. The modularity algorithm we implemented measures the density of edges inside communities to edges outside communities i.e. it looks for the nodes that are more densely connected together than to the rest of the network. We used the Louvain Method for community detection (Louvain Modularity) which is a greedy optimization algortithm to extract communities from large networks.
This method attempts to optimize the modularity of a partition of the network. The optimization is performed in two steps. First, the method looks for "small" communities by optimizing modularity locally. Second, it aggregates nodes belonging to the same community and builds a new network whose nodes are the communities. These steps are repeated iteratively until a maximum of modularity is attained and a hierarchy of communities is produced.
Through this measure, the colors indicate different communities determined by this algorithm and basically shows which gangs collaborate more between each other than to the rest of the network.

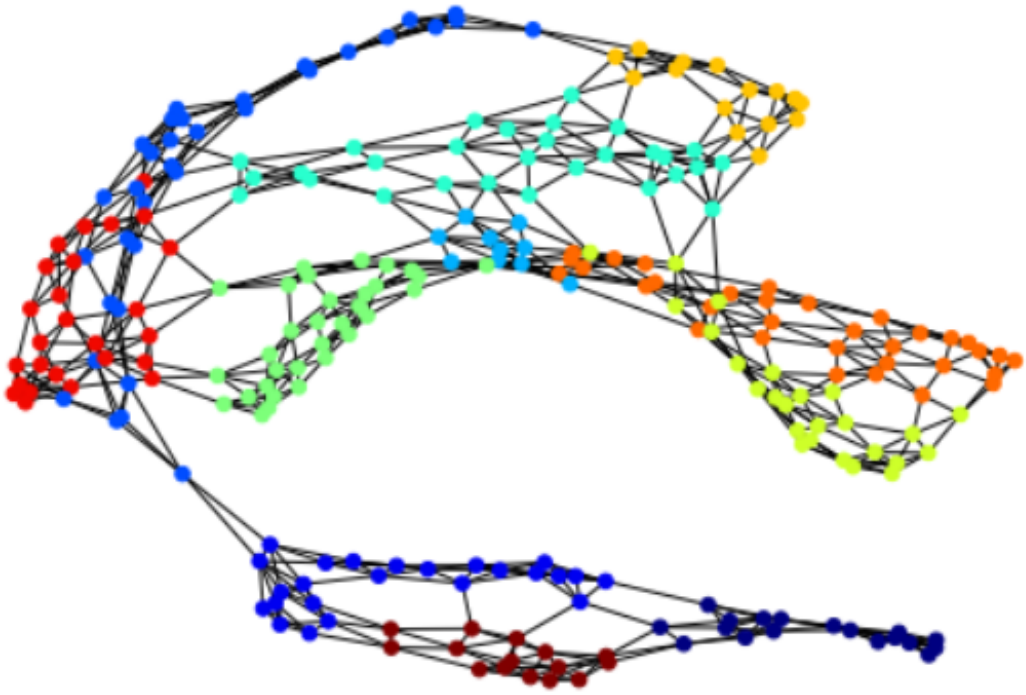

Figure 4: Modularity Graph

4.2 Predictive Analysis

The point here was to predict the presence of gangs in NYC based on the assumption that a concentration of similar types of crimes within a specific area can translate the presence of an organised crime group. Overall, this was a binary classification model, as the output could be either 'gang' or 'no gang'.

4.2.1 Random Forest

We here used a combination of Random Forest with Adaboost classifier. We use random forests as a weak learner of AdaBoost for selecting the high weight instances during the boosting process to improve accuracy, stability and to reduce overfitting problems.

4.2.2 Extreme Gradient Boosting

We also used an XgBoost model (depth: 4) with GridSearch to iterate through every parameter combination and configure optimal parameters for a given model. XgBoost is an implementation of the gradient boosting framework where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. The objective of the XGBoost model is given as:
$Obj = L + \Omega$, where L is the loss function which controls the predictive power, and Ω is regularization component which controls simplicity and overfitting. Since this is a binary classification problem, our loss function is the Log Loss function given below:

$$logloss = -\frac{1}{N}\sum_{i=1}(y_i log(p_i) + (1 - y_i) log(1 - p_i)$$

5 Evaluation

We based our model evaluation on performance metrics of K-fold cross-validation (k = 10) accuracy. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. This method shuffles the dataset randomly; then splits the dataset into k = 10 groups. For each unique group, it takes the group as a test data set and takes the remaining groups as a training data set. It then fits a model on the training set and evaluates it on the test set before retaining the evaluation score and discarding the model. It finally summarizes the skill of the model using the sample of model evaluation scores.Our results are summarized below:

Model	Validaton Accuracy
Random Forest with Adaboost	98.14%
XgBoost with KFold	98.14%

Our test accuracy is indeed similar between the two models and very high. Overall, our test set contained 431 test data points out of which 423 were correctly classified and 8 were wrongly classified. For all the misclassified data points, the input variables are exactly the same but the outputs are different, i.e., for degree centrality = -0.588345209510245, eigenvector centrality = -0.24071332516948 and betweenness centrality = -0.348897651407531, majority of the times, there is no gang present but sometimes there is gang present, so all the models are learning that there is no gang present for that input. So, we can make these two assumptions:
(i) The features used do not explain completely if there is gang.
or
(ii) There is some noise in the data.
Since we have very less data points on which we trained our models and all the models tried to learn the relation perfectly - the relation between the input and output variables is not that complex so with less difficulty all the models learned the relation, we think that explains why we ended up with close accuracy.

6 Conclusions

Our project was aimed at the study of NYC crime records and the identification of network patterns within the city boroughs.
We created a number of features including more specific geographical variables of neighborhoods (within boroughs). With these, we built multiple graphs to analyse the various relations between crimes and specific corners of the city, taken into account all crimes then each crime separately. We also created a number of visualisation tools for crime locations (folium map) and targeted attacks to counter crime spread across the networks.
We created other features based on network measures of centrality: degree, betweeness and eigencentrality. We further investigated the notion of greedy algorithms under the Louvain Method for modularity optimisation which provided interesting spatial results.
Finally, we attempted to predict the presence of gangs around NYC neighborhoods using boosting approaches of Random Forest with Adaboost Classifier and XgBoost (with GridSearch). These were evaluated on the basis of KFold cross-validation accuracy leading to high prediction results of 98.14%.
Our study fits in an increasing scholarly interest for crime social analysis. An interesting build-up on our work, should more detailed data on suspects (so far confidential) be available, would be to enquire gangs growth (in numbers and in activity) as an infectious disease (which we made an attempt at already). A further area of study could be that of criminal gang membership as a virus within a community, spreading by interactions among gang members and the population.

References

[1] *Social Network Analysis and Gangs* Michael Sierra-Arévalo and Andrew V. Papachristos

[2] *Analysis of México's Narco-War Network (2007–2011)*Hernán Larralde and Jesús Espinal-Enríquez

[3] *T. Wang, C. Rudin, D. Wagner, and R. Sevieri. Learning to detect patterns of crime. In Machine Learning and Knowledge Discovery in Databases, pages 515–530. Springer, 2013.*

[4] *M. B. Short, M. R. D'Orsogna, V. B. Pasour, G. E. Tita, P. J. Brantingham, A. L. Bertozzi, and L. B. Chayes. A statistical model of criminal behavior. Mathematical Models and Methods in Applied Sciences, 18(supp01):1249–1267, 2008.*

[5] *J. H. Ratcliffe. A temporal constraint theory to explain opportunity-based spatial offending patterns. Journal of Research in Crime and Delinquency, 43(3):261–291, 2006.*

[6] *A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. ICWSM, 10:178–185, 2010.*

[7] *J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. Journal of Computational Science, 2(1):1–8, 2011.*

[8] *X. Wang, M. S. Gerber, and D. E. Brown. Automatic crime prediction using events extracted from twitter posts. In Social Computing, Behavioral Cultural Modeling and Prediction, pages 231–238. Springer, 2012.*