Final Report
Prediction of Crime Patterns in Chicago using Network Analysis

Submitted By:
Ravisutha Sakrepatna Srinivasamurthy
Saroj Kumar Dash

December 6, 2017

Contents

1 Introduction

In this project we analyzed the Chicago Crime Data [3] and used the dynamic data of Chicago crimes to build a network and predict the next trend in crimes. Following sections explains our objective, datasets, our approach and our findings.

2 Objective

Our objective is to to predict crime pattern in Chicago city by using data from previous years with the help of network analysis techniques. In order to validate our results, we are using 2015's data as the validation dataset.

3 Dataset

The dataset spans from a time period of 2001 to present. Since the data is produced by a government agency, data can be relied upon for its correctness. The 15 years old dataset of crimes describes about all the crime events that have happened in Chicago area. There are many analysis done on this datasets before but most of the analysis are in the traditional ways of qualitative or quantitative statistical analysis of the dataset. Our analysis is different as we try to view the dataset in form of a network and use network science concepts to extract features from it.

3.1 Dataset Description

Below is a list all the data sources that we used. The most challenging part of the project was to make connections between the various parts of the crime dataset and also among all other datasets obtained from other departments. Dataset Sources:

- Crime dataset[5]: Chicago crime data from 2001 to present.
- Chicago Library Data[6] Library locations and details with map coordinates.
- Library visitors by location[7]: This data gives information about the number of visitors for all libraries in Chicago city.
- Police Station Data[9]: Gives us information about the location of a police station.
- Police Districts Boundaries[10]: This data tells about the areas for which a police station is responsible for. Used to connect communities, crimes and police stations closest to it.
- 311 Service requests[8]: A set of datasets for all sanitation queries.
- Chicago public School progress Information[11] : Data was available from 2011 to 2017.

The main challenge in the project was to accumulate the data from each time line because there were lot of data which were not available for some years. Only the crime data could be traced back from 2001. But only a few were available from 2001. But all datasets were available from 2001. Hence dataset from 2011 to 2015 was used as training dataset. 2015 was used for validating the data model.

3.2 Data Cleaning

We cleaned the data and filtered the data according to months for each year (2011-2015). We add a new column called community based on the location information from the dataset if the community info was not directly available.

4 Approach

Chicago city can be divided into 77 communities. Each node is considered as a node. According to Chicago Police Dept., crimes can be categorized into 401 types. We considered each crime type as a node. Crime dataset had the crime location information. Hence we could connect crime with community. Similarly, communities were connected to other datasets like police stations, libraries, 311 queries in that community etc. which describes different aspects of the community. This helped us see the community in terms of sub networks where each subnetwork describes a certain aspect about of community. After our network was formed we found similarity between two communities. Later we will explain about the usage of similarity in predicting the crimes. We also preformed clustering using similarity measure to see the trends in the communities and how dynamic it is. We performed polynomial regression on the dataset. We were successful in capturing the crime patterns. We can use the same prediction model to project it for 2018 too (but it requires few more things to be done). In the way of achieving these results, we have made a framework which is flexible enough to add more dimensions to the Chicago Crime analysis. By adding other datasets related to Chicago city, we can better predict the crime.

4.1 Building network

We started with initially considering crime data for the year 2015. After we successfully built the network for the year 2015, we then used the same system to build network for years. In 2015, there were 263,477 reported crimes. There were total of 401 crime types according to Chicago Police website[4]. We considered the primary type field crime reported in the crime event. We considered each crime type as a node. This formed the crime-type subnetwork inside the complete network. To represent the location data we considered the community data. Each crime type is connected with a community with the weight of the edge as the number of such type of crime that has happened in that community. Chicago can be divided based on many attributes such as ward number, precinct, district, etc. but we found that community is a better way to separate them out as we can connect other datasets easily.

Along with crime dataset, we considered other aspects of the community such as education data(library,school data), economic status(311 request data) and police station data. Some of these datasets are based on wards and some are based on precincts etc. Hence it was difficult to unify them based on the community. To unify these datasets we used shapely and other GIS techniques. We used these techniques to find out to which community does the given library or the sanitation query belongs to. The edge weight in the case of 311 services were based on the number of queries that were done by that community and in case of library it was the number of visitors to that library in that month. In case of school data the the edge weight was the ACT score of that school which told us about the quality of the school in that area.

4.2 Connections within networks

We felt it to be good idea to connect different parts of the network other than just connecting to community. Hence, we connected crime type subnetwork with the police station network based on location of crimes. This also helped in getting a good similarity measure between communities. Weight is the number of crimes of the type that had occurred in that police district. In the future, we would like to connect school nodes with crimes based on proximity of crime location.

5 Prediction

To predict the number of crimes for a given month, we followed the following steps.

- Before building the network, we normalized the data. After building the network, we obtained similarity measures between 77 communities. We used random walk similarity and adamic adar similarity to get the similarity. Similarity between communities is given by a $77 * 77$ matrix.
- We used polynomial regression with degree 2 to predict the crime pattern. It requires feature vector as the input. So for a given month and for a given community, the feature vector is a concatenation of features from this given community and features from similar community. Features of each community is nothing but the weights of edges from community to all other nodes.
- This concatenation of feature actually lead to the improvements in predictions.

5.1 Results

Visualization After building the network, we used the software package gephi to visualize(Figure 1).

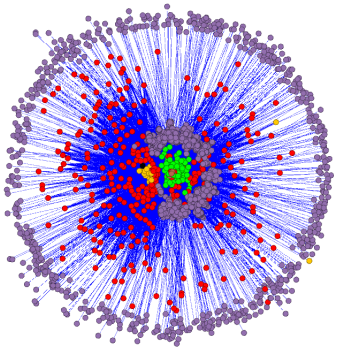

Figure 1: Visualization of the network for the year 2011. Nodes for this network are community(green), crime types(red), Schools(violet), Police stations(orange) and sanitations(brown).

Using results from similarity, we grouped communities in Chicago city. Below(Figures: 2 - 5) are maps of Chicago with communities grouped according to the similarity measure for the years form 2011 to 2014.

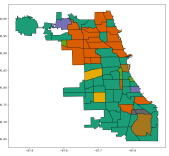

Figure 2: Clustering Community 2011

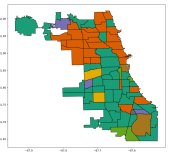

Figure 3: Clustering Community 2012

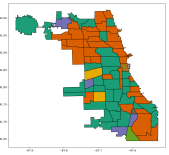

Figure 4: Clustering Community 2013

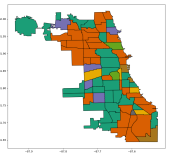

Figure 5: Clustering Community 2014

Figure 6: Similar communities over the years

Regression We used polynomial regression to predict the crime patterns for the year 2015. We had 4 years of complete dataset to use for our prediction. As explained above, each community will have its own feature vector. This vector comprises of weights of edges from the community to other nodes. This feature vector is concatenated with feature vector from the most similar community. This way we input the values for the regression model. We trained the regression model on just homicide and sexual assault crimes. The plot shows the predicted and actual output for the year 2015 [Figure 7]. This shows that our model successfully captures the crime pattern of Chicago city.

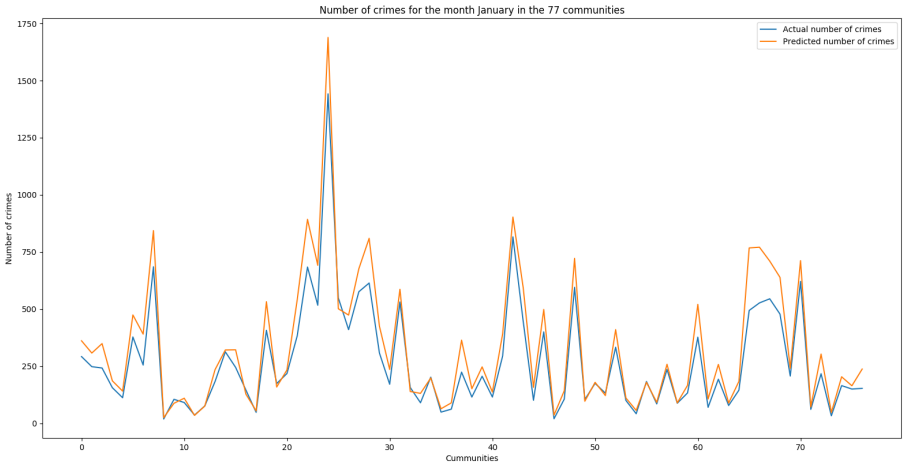

Figure 7: Actual and predicted number of crimes (homicide + sexual assault) for all 77 communities (for the month January)

6 Conclusion and Future Work

We have done a comprehensive initial analysis of the dataset by analyzing it with the network analysis. We would still want to explore the data by using some new similarity technique. And we want to consider more than one similarity measure for preforming regression. We also want to try and connect different subnetworks in our network. We believe that by making the network more connected and by adding more sets of subsets to the network, we can predict the crime pattern better. To predict the crime patten for 2018, we need to regress for all input features such as number of visitors to the library, average ACT scores for 2018 etc. This needs more effort and we are hoping to continue this project next semester.

Appendices

Appendix A :Make network python code:

```
#!/usr/bin/python3

#Author :    Ravisutha Sakrepatna Srinivasamurthy
#Project:    Analysis of Chicago Crime Data

import networkx as nx
import datetime
import csv
import numpy as np
from crime_network import Crime_Network
from police_network import Police_Network
from service_community import ServiceNetwork
from community_libraries import Library_Network
from school_network import SchoolNetwork
from crime_police import CrimePoliceNetwork
from path import Path
from normalize_network import Normalize

class Build_Network:
    def __init__(self):
        """Initialize the build network by storing the given dictionary.
        Example:
            net = Build_Network ()
            net.load_network ()
            G = net.get_network ()
        """
        self.new = 1
        return

    def add_net (self, comm_dict, color_attr=None, other_attr=None):
        """Initialize the build network by storing the given dictionary. """

        self.comm_dict = comm_dict
```

```
        self.create_graph (color_attr, other_attr)

    def _create_comm_nodes (self):
        """Create 77 community nodes. """

        for community in range (1, 78):
            self.G.add_node (community, color='green')
            self.G.nodes[community]['type'] = 'community'+ str (community)

    #Create graph using numpy array
    def create_graph (self, color_attr=None, other_attr=None, comm_network=True):
        """ Convert community dictionary to networkx graph. """

        #If the graph is created for the first time
        if (self.new == 1):
            self.G = nx.Graph ()
            if (comm_network == True):
                self._create_comm_nodes ()
            self.new = 0

        for source in self.comm_dict:
            for target in self.comm_dict[source]:
                if (target not in self.G.nodes):
                    self.G.add_node (target)
                if (color_attr != None):
                    self.G.nodes[target]['color'] = color_attr
                if (other_attr != None):
                    self.G.nodes[target][other_attr[0]] =  other_attr[1]

                try:
                    self.G.add_edge (source, target, weight=float(self.comm_dict[sour
                except ValueError:
                    self.G.add_edge (source, target, weight=0.0, color='blue')

        #print ("Your network is ready")
        return (self.G)

    #Write to file and view in gephi
    def write_file (self, path="new.graphml"):
        """ Write the network to a file. Helps in visualization. """

        nx.write_graphml(self.G, path[0])

    #Get the network
    def get_network (self):
        """ Returns the current network in form of Graph """
        return(self.G)

    #Get attributes
```

```
def get_attributes (self):
    """ Returns attributes of a community. """
    return (self.attr)

#Add atributes
def add_attributes (self, G):
    """ Helper attributes for better visualization. """

    for i in range (np.shape(self.A)[0]):
        if (i < 4):
            G.nodes[i]["color"] = 'green'
        elif (i < 11):
            G.nodes[i]["color"] = 'blue'
        elif (i < 413):
            G.nodes[i]["color"] = 'red'
        else:
            G.nodes[i]["color"] = 'yellow'

#Load network
def load_network (self, year=2015, month=1, save=False, connect=False):
    """ Loads the network. """

    path = Path ()
    norm = Normalize ()
    self.attr = {}

    print ("\t1. Creating network")

    #Get community and crime dictionary (Code: 10000)
    print ("\t2. Adding Crime Network")
    a = Crime_Network (path.get_path (year=year, month=month, type="crime"), 1000
    comm_crime = a.get_network ()
    self.attr["crime"] = comm_crime
    comm_crime = norm.maxMinNormalize (comm_crime)
    self.add_net (comm_crime, 'red', ('type', 'crime'))

    #Get community and police station dictionary (Code: 30000)
    print ("\t3. Adding Police Network")
    p = Police_Network (path.get_path (year=year, month=month, type="police"), -
    comm_police = p.get_network ()
    self.attr["police"] = comm_police
    comm_police = norm.maxMinNormalize(comm_police)
    self.add_net (comm_police, 'orange', ('type', 'police'))

    #Get community and 311 service dictionary (Code: 40000 - 110000)
    print ("\t4. Adding 311 service Networks")
    p = ["sanity", "vehicles", "pot_holes", "lights_one", "lights_all", "lights_a
    community = [-5, -6, -6, -5, -5, -5, -5, -5]
    code = [40000, 50000, 60000, 70000, 80000, 90000, 100000, 110000]
```

```
        for i, name in enumerate (p):
            print ("\t\t- Adding {} network".format (name))
            s = ServiceNetwork (path.get_path (year=year, month=month, type=name), co
            comm_sanity = s.get_network ()
            self.attr[name] = comm_sanity
            comm_sanity = norm.maxMinNormalize(comm_sanity)
            self.add_net (comm_sanity, 'brown', ('type', name))

        #Get community and library dictionary (Code: 120000)
        #print ("5. Adding Library Networks")
        #l = Library_Network (path.get_path (year=year, month=month, "library"), -1,
        #comm_library = l.get_network ()
        #net.add_net (comm_library, 'white', ('type', 'library'))

        #Get community and school dictionary (Code: 130000)
        print ("\t6. Adding School Network")
        s = SchoolNetwork (path.get_path (year=year, month=month, type="school"), 130
        comm_school = s.get_network ()
        self.attr["school"] = comm_school
        comm_school = norm.maxMinNormalize (comm_school)
        self.add_net (comm_school, 'violet', ('type', 'school'))

        #Connect police and crime network
        if (connect == True):
            print ("\t7. Connecting Police and Crime Network")
            pc = CrimePoliceNetwork (path.get_path (year=year, month=month, type="po
            crime_police = pc.get_network ()
            crime_police = norm.maxMinNormalize (crime_police)
            self.add_net (crime_police)

        if (save == True):
            net.write_file(path.get_path (year=year, month=month, type="output"))

if (__name__=='__main__'):
    """ Build network example usage"""

    years = [2011, 2012, 2013, 2014, 2015]
    for year in years:
        for month in range (1, 13):
            print ("For year: {} and for the month {}".format (year, month))
            net = Build_Network ()

            net.load_network (year=year, month=month)
            net.get_network ()
            net.get_attributes ()
```

References

[1] Ahmed, N. K., Rossi, R. A., Willke, T. L., & Zhou, R. 2016, arXiv:1610.00844

[2] F. Fouss, A. Pirotte, J. m. Renders and M. Saerens, "Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation," in IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 3, pp. 355-369, March 2007.

[3] https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2

[4] https://data.cityofchicago.org/Public-Safety/Chicago-Police-Department-Illinois-Uniform-Crime-c7ck-438e

[5] https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2

[6] https://data.cityofchicago.org/Education/Libraries-Locations-Hours-and-Contact-Information/x8fc-8rcq

[7] https://data.cityofchicago.org/Education/Libraries-2011-Visitors-by-Location/xxwy-zyzu

[8] https://data.cityofchicago.org/browse?q=311&sortBy=alpha&utf8=\%E2\%9C\%93

[9] https://data.cityofchicago.org/Public-Safety/Police-Stations/z8bn-74gv

[10] https://data.cityofchicago.org/Public-Safety/Boundaries-Police-Districts-current-/fthy-xz3r

[11] https://data.cityofchicago.org/Education/Chicago-Public-Schools-Progress-Report-Cards-2011-/9xs2-f89t