PREDICT THE CRIME: TWO DIFFERENT TIME SCALES

Math 450 - Project Report

December 3, 2019

This project is a compound of caution and imagination!

Tingying LU
tfl5119@psu.edu
Penn State
Department of Mathematics
Instructor: Timothy RELUGA

Contents

1 Highlight . . . 2
2 Introduction . . . 2
2.1 Background . . . 2
2.2 Goal . . . 2
2.3 Data . . . 3
3 Clustering . . . 3
3.1 Motivation . . . 3
3.2 Method . . . 4
4 Monthly Forecast Model . . . 4
4.1 Motivation . . . 4
4.2 Notation and Assumption . . . 5
4.3 Model . . . 5
4.4 Numerical Results . . . 6
5 Daily Forecast Model . . . 7
5.1 Motivation . . . 7
5.2 Notation and Assumption . . . 8
5.3 Model . . . 8
5.4 Numerical Results . . . 10
6 Conclusion and Discussion . . . 11

1 Highlight

1. Simplify the location data using clustering.
2. Regression model for long-time (monthly) forecast; Markov Chain model for short-time (daily) forecast.
3. Statistical tools, e.g. kernel density estimates, have been implemented to improve the results.
4. Tests, including the prediction errors, have been accomplished to verify our models together with lost of figures.
5. A short discussion, including potential improvements, is made at the end.

2 Introduction

We will first introduce the problem from the criminology point of view, and state our goal. The data structure will be mentioned at the end of the section.

2.1 Background

Through the development of our society, preventing crimes is a very important problem for law enforcement officers around the world. Having an accurate crime prediction is very necessary. With the advancement of computational technology, many new methods of predicting crimes have been developed. Now, Police departments have started to incorporate some mathematical crime prediction method to help in their investigations of criminals. Those mathematical method allow the police to focus on a relatively small area and hence cut short the time that criminals are free to victimize the innocent. According to the criminology, A criminal usually targets similar victims and often will target them in similar locations. Thus, investigate the previous crimes' data is very helpful during the crime prediction, especially in estimating time and the location. In this paper, we assume that different type of crimes is independent, and the crime distribution in the next time period only depend on the crime distribution of the current time period. According to those assumptions, we create a long time linear regression model and a short time Markov Chain model to predict the Chicago robbery crime.

2.2 Goal

Our goal is developing two different models, for both short (day) and long (month) time scales, to predict the possible crime in next time intervals based on the time and locations of the past crime information. We name the two model 'daily forecast model' and 'monthly forecast model' respectively.

There are two major motivations of considering two different time scale. To begin with, both the daily and monthly forecasts are required to provide appropriate warnings for different target users. Secondly, the accumulated error produced by implementing the daily forecast model iteratively could be significant.

2.3 Data

The data are from Gun Crimes Heat Map[1], which includes the data of Chicago crimes from 2001 to now. In the original data set, there are more than two hundred thousand cases with 10 different features including the case number, types of crime, time and location of each crime. For the types of crime, the Chicago police divide crime into robbery, arson, assault, battery, kidnapping, homicide, rape, theft, weapons violation, criminal damage and other reason. For simplicity, only the robbery cases happened during 2018 was considered in the project which provides us a total of 9127 samples, which means the correlations between different types of crimes are totally ignored in our study.

Due to our goal, only the time (in days) and location (latitude, longitude) features are taken into account, that is, the sample set becomes a 9127-by-3 matrix. For example, after sorted by time, the firs row of the matrix is written as

$$0 \quad 41.86168324 \quad -87.71041019,$$

here 0 stands for 'Day 0', and $(41.86168324, -87.71041019)$ represents latitude and longitude respectively. (positive latitude means north while negative longitude means west)

3 Clustering

In this section we are going to introduce a useful statistical tool, known as clustering, to simplify our location data as a precondition.

3.1 Motivation

Intuitively speaking, clustering is a process of labeling sample points $\{X_i\}$ embedded in a given metric space such that certain 'total energy' (a cost function defined by the metric) is minimized.

In our situation, detailed geographic coordinates are not admissible since it introduce too many degrees of freedom. Besides, predicting the exact location of the next crime is not achievable due to the randomness. Thus, the geographic coordinates will be replaced by the label obtained from the clustering. One can think of the labels as coarse-grained location data.

[1] https://data.cityofchicago.org/Public-Safety/Gun-Crimes-Heat-Map/iinq-m3rg

3.2 Method

The clustering method applied is the k-means clustering, (e.g. Chapter 13 of [2]) which assigns given observation to exactly one of k clusters, where k is chosen before the algorithm starts.

There are two benefits. To begin with, such k, total number of clusters, is selected a priori. This is important in later application. Moreover, as a byproduct, k centroids (cluster centers), average of the data points in each clusters, are generated to represent the corresponding clusters. One potential application for those centroids could be the site selections of the police stations.

For the implementation, we used the MATLAB function 'kmeans'[2]. Figure 1 shows the clustering result. In the following sections, our monthly/daily forecast models will be designed based on them, that is, the detailed geographic coordinates will be replaced by the label of the clusters. We will explain why we take $k = 4$ and $k = 2$ later.

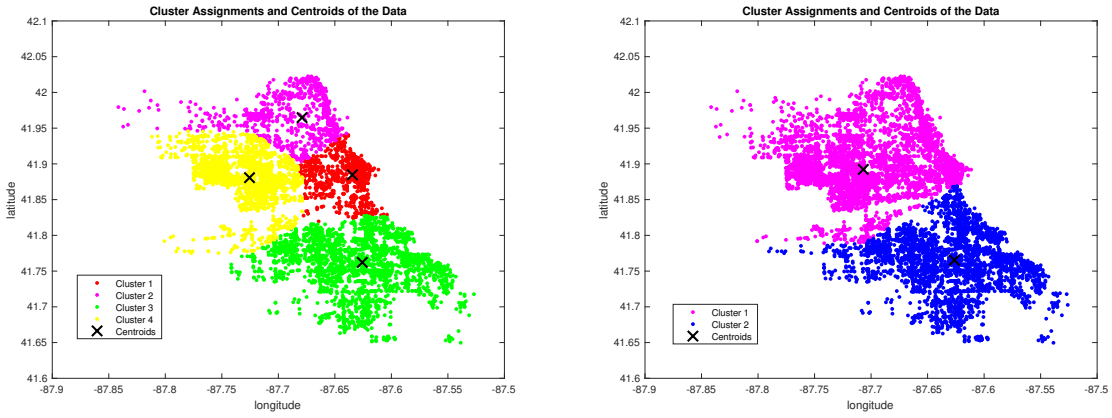

Figure 1: Cluster the location data based on k-means algorithm: $k = 4$ for monthly forecast model (left) and $k = 2$ for daily forecast model (right).

4 Monthly Forecast Model

In this section, we are going to discuss the monthly forecast model based on linear regression.

4.1 Motivation

Unlike the daily model, the total amount of crimes happening each month does not fluctuate fiercely as long as the environment (economics, populations, crime control strategies, etc.) stays stable. Thus, under the month time scale, the distribution of the crime over the clusters becomes a significant variable for the government to evaluate certain control strategies. By

[2]https://www.mathworks.com/help/stats/kmeans.html

setting our target users to be the government, longer time scale and quantitative results are preferred, and the regression-type of model is a good fit.

4.2 Notation and Assumption

Since our goal is to predict the crime distribution, the quantity of interests will be the density of crime among the clusters each month. Let N denotes the total number of clusters, then the density matrix $\rho := [\rho_1, \ldots, \rho_{12}] \in \mathbb{R}^{N\times 12}$, where the columns ρ_i, provide the density of each month. In practice, ρ_i will be approximated by the empirical density $\hat{\rho}_i$ computed from the available sample, that is, the density matrix will be approximated by $\hat{\rho}$ given by

$$\hat{\rho} = [\hat{\rho}_{ij}], \quad \hat{\rho}_{ij} = \frac{\#\{ \text{ cases in cluster i during month j } \}}{\#\{ \text{ cases in all clusters during month j } \}}, \quad i = 1,2,\ldots,N,\ j = 1,2,\ldots,12.$$

We pose following assumptions.

1. Our sample set (9127 cases) is large enough to provide good estimates towards the density matrix ρ.
2. We assume that the distribution in the next time period only depend on the distribution of the current time period and such relation is linear with time-independent coefficients.

4.3 Model

Based on the assumptions, we introduce the following linear regression model

$$\rho_{i+1} = P\rho_i + \epsilon_{i+1}, \quad i = 1,2,\ldots 11, \tag{1}$$

where $P \in \mathbb{R}^{N\times N}$ is the time-independent regression coefficients and ϵ_{i+1} represents the mean-zero error. To verify our model, we split the data into training data ($\hat{\rho}_1 - \hat{\rho}_9$) used for determining the regression coefficients P, and the test data ($\hat{\rho}_{10} - \hat{\rho}_{12}$) reserved for verification.

According to the linear regression theory, the regression coefficients P can be estimated via a least squares problem

$$P \approx \hat{P} = \arg\min_{P\in\mathbb{R}^{N\times N}} \|\hat{Y} - P\hat{X}\|, \quad \hat{Y} := [\hat{\rho}_2, \hat{\rho}_3, \ldots, \hat{\rho}_9], \quad \hat{X} := [\hat{\rho}_1, \hat{\rho}_2, \ldots, \hat{\rho}_8], \tag{2}$$

where $\hat{Y}$ and $\hat{X}$ can be treated as the output and the input respectively. So far, we have not decided the total number of clusters N. Obviously, the greater the value of N the more detailed location information will remain after the clustering. However, one cannot take N too large in our situation, since we want the least squares problem (2) is an over-determined system. Notice that $P \in \mathbb{R}^{N\times N}$, as a result, we need $N < 8$ (8 is the total number of input/output pairs) to satisfy the condition. In particular, we take $N = 4$.

4.4 Numerical Results

So far, we have formulated the linear regression model (1) and introduced the corresponding least squares problem (2) to estimate the regression coefficients P. Solving (2), we obtained

$$\hat{P} = \begin{pmatrix} -0.5804 & -0.2471 & 0.1375 & 0.7190 \\ 1.2575 & 0.2683 & -0.0582 & -0.2712 \\ 0.1013 & 0.7305 & 0.5014 & 0.2788 \\ 0.2215 & 0.2483 & 0.4193 & 0.2734 \end{pmatrix}. \tag{3}$$

Such $\hat{P}$ contains rich information. Here are my comments.

1. The column sum of $\hat{P}$ is always one, due to the fact that it maps a density vector to another density vector.

2. None of the component of $\hat{P}$ is close to zero. Their absolute values are greater than 0.1 with only one exception. This supports the significance of our model in the sense that there exist correlations between different clusters in adjacent time intervals. In other words, the crime level of a certain cluster in the coming month depends on the crime levels of all the clusters in current month.

3. The sign of the component $\hat{P}_{ij}$ corresponds to the positive or negative impacts of cluster j on cluster i in next month. For example, the diagonal elements can be interpreted as the 'survival rate' of the crimes inside each clusters, and one can tell that the cluster-1 is different from the other clusters since $\hat{P}_{11} = -0.5804$ is the only negative element on the diagonal. A negative 'survival rate' means the crime of the cluster is under control, since the crime has a relatively low chance to 'stay' insider the cluster. Return to the left figure of Figure 1, cluster-1 actually corresponds to the downtown area of Chicago, which yields the lowest crime rate among the city of Chicago. Thus, our model does consist with the reality.

4. Such $\hat{P}$ also suggests that it is possible to improve the safety situation of certain area by controlling its neighborhood. Since most of the off-diagonal elements are positive.

From (1), the error vector ϵ_i will be estimated by

$$\hat{\epsilon}_i = \hat{\rho}_i - \hat{P}\hat{\rho}_{i-1}.$$

We interpret such $\hat{\epsilon}_i$ as the local error in the sense that it corresponds the error after a single implementation of the model. Meanwhile, global (accumulated) errors can be generated via iterative implementations. In particular, we define the estimated global error as

$$\tilde{\epsilon}_i = \hat{\rho}_i - \hat{P}^{i-1}\hat{\rho}_1.$$

Compute the ratio between the norm of $\hat{\epsilon}_i$ ($\tilde{\epsilon}_i$) and the norm $\hat{\rho}_i$. We obtain the relative local (global) errors. The left figure of Figure 2 shows the two relative errors from February to September. One can tell that the local error is under control.

The last piece of the numerical results is the comparison between the predication and true density vectors based on the test data. Recall that we have reserved the density vectors of the last three months $\{\hat{\rho}_{10}, \hat{\rho}_{11}, \hat{\rho}_{12}\}$ as test data. On the other hand, the density forecasts are given by

$$\text{forecast of } \hat{\rho}_j = \hat{P}^{j-9}\hat{\rho}_9, \quad j = 10, 11, 12.$$

The right figure of Figure 2 shows the corresponding results. One can tell that the predication errors are quite significant. We will address such issue in the discussion.

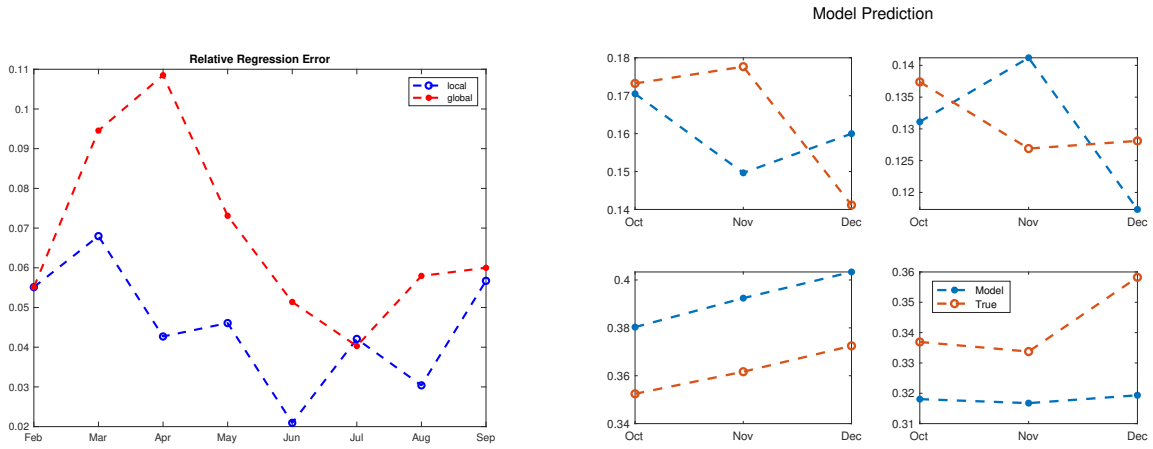

Figure 2: The relative error (left) in the regression and the prediction of the last three months compared with the true density (right).

5 Daily Forecast Model

In this section, we are going to discuss the daily forecast model based on Markov Chain.

5.1 Motivation

Different from the monthly forecast model, a daily forecast model is more helpful for the citizens and tourists. People in the communities can avoid traveling near the potential dangerous area based on the daily warning.

Since our target users are the individuals, a qualitative forecast is already good enough, for example, using 'L' (Low), 'M' (Medium), and 'H' (High) to classify the crime risk. Such qualitative classification introduces a natural state space to construct a Markov Chain model.

5.2 Notation and Assumption

To define the state space, recall that N denotes the total number of clusters, and each cluster will be assigned a letter to represent the crime risk level ('L' for low, 'M' for medium, and 'H' for high). Thus, we define the product space

$$\mathscr{S} := \{\text{L, M, H}\}^N \tag{4}$$

as our state space. In practice, one can define a pair thresholds (lb, ub) for each cluster such that if the total number of cases lies inside the interval $[lb, ub]$, the crime risk will be assigned as 'M'. (lower/higher than lb/ub will be assigned as 'L'/'H')

To legitimate our Markov Chain model, we make the following assumptions.

1. The thresholds (lb_i, ub_i) $i = 1, 2, \ldots, N$ exist for each cluster.
2. Let $X_k \in \mathscr{S}$ denotes the crime risk vector at day k. Then the sequence of random variables $\{X_k\}$ builds a Markov Chain.

5.3 Model

Our daily forecast model is a Markov Chain model with state space $\mathscr{S}$ (4) and transition matrix P. Several issues need to be settled down before we discuss the numerical results.

Similar to the monthly model, we have issues in choosing appropriate value for N. Notice that $\#\mathscr{S} = 3^N$, which means the transition matrix P should be a 3^N-by-3^N matrix, and its components will be estimated by the empirical probabilities. One can already tell that the adequate sample size (sample size enough to produce reasonable estimates) grows exponentially (at least) as N increases. Since our sample set is fixed, we take $N = 2$ in our model, which gives us a total of 9 states. The corresponding clustering result has been shown in Figure 1.

Arrange the state space in the following way

$$s_1 = (\text{L, L}),\ s_2 = (\text{L, M}),\ s_3 = (\text{M, L}),\ \ldots,\ s_9 = (\text{H, H}),$$

and the transition matrix P satisfies

$$P = [P_{ij}], \quad 1 \le i, j \le 3, \quad P_{ij} = \mathbb{P}\left(X_{k+1} = s_i \mid X_k = s_j\right) \quad \forall k,$$

where P_{ij} will be estimated by the empirical conditional probability, that is,

$$P_{ij} \approx \hat{P}_{ij} = \frac{\#\{(X_{k+1} = s_i, X_k = s_j)\}}{\#\{X_k = s_j\}}. \tag{5}$$

The last issue will be selecting the thresholds (lb_1, ub_1) and (lb_2, ub_2) for the two clusters.

Recall that these thresholds define a map $S:\mathbb{N}^2 \to \mathscr{S}$ given by

$$S(a,b) = (S_1(a), S_2(b)), \quad s.t. \quad S_1(a) = \begin{cases} \text{L} & 0 \le a < lb_1 \\ \text{M} & lb_1 \le a \le ub_1 \\ \text{H} & ub_1 < a \end{cases}, \quad S_2(b) = \begin{cases} \text{L} & 0 \le b < lb_2 \\ \text{M} & lb_2 \le b \le ub_2 \\ \text{H} & ub_2 < b \end{cases}. \tag{6}$$

Such map will be used in generating the observation of $\{X_i\}$ from the samples. We introduce the following ratio criteria

$$\#\{\text{L}\} : \#\{\text{M}\} : \#\{\text{H}\} = 3 : 4 : 3. \tag{7}$$

The motivation comes from the fact that we are designing a model for daily crime risk warning, which means 'H', representing high risk, should be a relatively rear event. Thus, the thresholds can be determined by the inverse cumulative distribution function (ICDF). For example, let $F_1(x)$ be the cumulative distribution function (CDF) of the random variable describing the total number of crimes happening in cluster-1 each day. Then, the thresholds (lb_1, ub_1) satisfy

$$lb_1 = F_1^{-1}(0.3), \quad ub_1 = F_1^{-1}(0.7).$$

To estimate the iCDFs based on the sample, we applied a smoothing technique known as the Kernel Density Estimates (KDE). (e.g. Chapter 6 of [1]) In MATLAB, 'ksdensity'[3] is the corresponding function, and Figure 3 shows the result. With the estimated ICDF and the ratio criteria (7), we introduce the following threshold for our map S.

Cluster	lb_i	ub_i
1	8.5	12.5
2	11.5	17.5

Table 1: The value of thresholds.

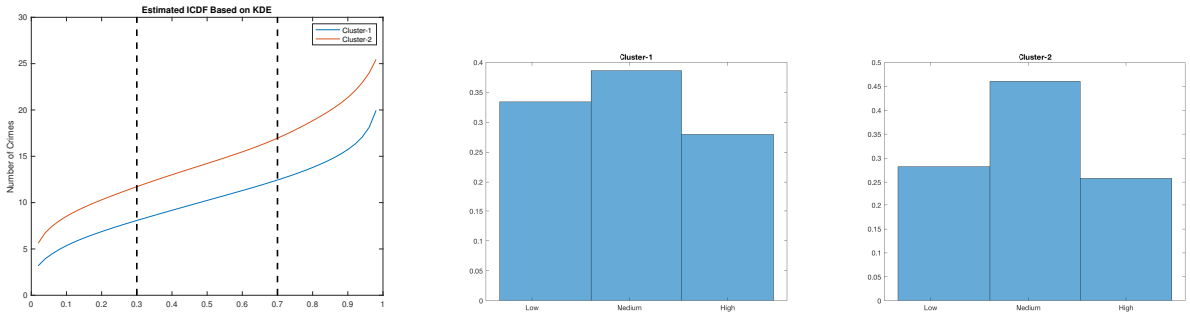

Figure 3: The estimated ICDFs based on KDE. (left) The histogram of the risk state. (right)

Based on Table 1, apply the map S in (6) to the data and we obtain the observations of our Markov Chain $\{X_i\}$. Figure 3 (right) shows the resulting marginal histogram of two clusters. One can tell that the ratio criteria (7) has been taken care of.

[3] https://www.mathworks.com/help/stats/ksdensity.html

5.4 Numerical Results

Compute the estimated transition matrix $\hat{P}$ based on (5), and we obtain the following 9-by-9 matrix (round to 4 digits of the decimal point)

$$\hat{P} = \begin{pmatrix} 0.3830 & 0.2000 & 0.0833 & 0.1944 & 0.0423 & 0.0000 & 0.1579 & 0.0851 & 0.0278 \\ 0.1489 & 0.0600 & 0.0833 & 0.1667 & 0.3099 & 0.0294 & 0.1579 & 0.0851 & 0.0556 \\ 0.0851 & 0.1000 & 0.0000 & 0.0278 & 0.0986 & 0.0588 & 0.1053 & 0.0426 & 0.0278 \\ 0.1277 & 0.0600 & 0.0833 & 0.1111 & 0.0986 & 0.1176 & 0.1579 & 0.0851 & 0.0556 \\ 0.1489 & 0.1800 & 0.3333 & 0.0556 & 0.1690 & 0.3235 & 0.2105 & 0.2979 & 0.1111 \\ 0.0213 & 0.0800 & 0.2083 & 0.0833 & 0.0986 & 0.0882 & 0.0526 & 0.0851 & 0.1667 \\ 0.0638 & 0.1400 & 0.0000 & 0.0556 & 0.0423 & 0.0294 & 0.1053 & 0.0000 & 0.0278 \\ 0.0213 & 0.1000 & 0.2083 & 0.1667 & 0.0845 & 0.2647 & 0.0526 & 0.1702 & 0.1667 \\ 0.0000 & 0.0800 & 0.0000 & 0.1389 & 0.0563 & 0.0882 & 0.0000 & 0.1489 & 0.3611 \end{pmatrix}. \tag{8}$$

Such $\hat{P}$ characterize the Markov Chain $\{X_i\}$ in the sense that it determines the evolution of the distribution of the X_i on the state space $\mathscr{S}$. Here are my comments.

1. Compared with the regression coefficients obtained in the month forecast model (3), although both of them have 1 as their columns sums, the transition matrix $\hat{P}$ is non-negative in the sense that all the components are greater or equal than zero.

2. $\hat{P}$ in (8) does contains zero elements, but this only means the corresponding empirical conditional probabilities are zero, which might be caused by the lack of samples.

3. Recall the equilibrium density of a give Markov Chain, if exists, is determined by the eigenvector of the transition matrix with respect to the eigenvalue $\lambda = 1$. Apply the MATLAB function 'eig' to $\hat{P}$ in (8), and we find $\lambda = 1$ is the single eigenvalue with eigenvector (round to 4 digits of the decimal point)

$$\hat{p}_{eq} = (0.1327, 0.1375, 0.0661, 0.0963, 0.1953, 0.0931, 0.0523, 0.1285, 0.0982). \tag{9}$$

   Thus, it is relatively safe to say that our Markov Chain yields a unique equilibrium density $p_{eq} \approx \hat{p}_{eq}$ in (9). Figure 4 visualizes $\hat{p}_{eq}$ on the state space $\mathscr{S}$. One can tell that the peak locates at 'M-M' state (both clusters are of medium crime risk), which is consistent with the reality.

4. In Figure 4, we also ploted the empirical density vector $\tilde{p}_{eq}$ given by

$$(\tilde{p}_{eq})_i = \frac{\#\{X_k = s_i\}}{\#\{X_k\}}, \quad i = 1, 2, \ldots, 9.$$

   It is easy to show that such $\tilde{p}_{eq}$ satisfies $\hat{P}\tilde{p}_{eq} = \tilde{p}_{eq}$ using (5). Thus, $\tilde{p}_{eq} = \hat{p}_{eq}$, which explains why the two curves are almost the same.

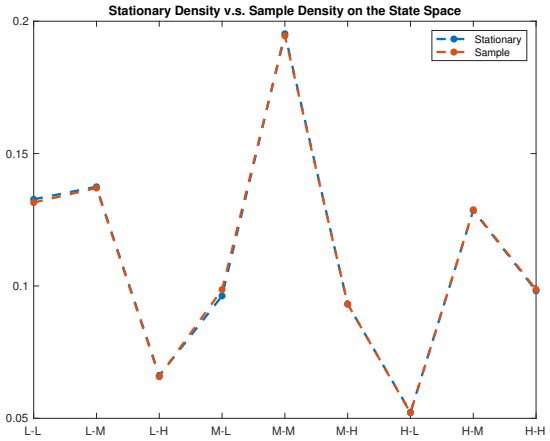

Figure 4: The estimated equilibrium density on the state space $\mathscr{S}$. Blue dashed line shows the vector $\hat{p}_{eq}$ while red dashed line shows the vector $\tilde{p}_{eq}$.

6 Conclusion and Discussion

To summarize, after clustering the location data, we have introduced two crime forecast models under short/long-time scales based on different motivations and target users. Both of them are able to explain certain phenomenon in the realities and achieve some success. But there are definitely far from perfect.

For the monthly forecast model, the predication strength is rather weak, potential reasons are as follows.

1. The relation between the successive density vectors is nonlinear.
2. The correlation among different types of crime cannot be ignored.
3. There are memory effects on the density vectors, that is, ρ_{k+1} depends on not only just ρ_k but also other density vectors in the previous time.

For the daily forecast model, the ghost is the curse of dimensionality. As we have mentioned, the admissible sample size increases at least exponentially according to the total number of clusters. In the application point of view, one need a number of clusters to provide geographically detailed forecast to the users. While the amount of the available sample is usually fixed, the possible remedies are

1. model reduction to shrink the state space, e.g. ignore the interaction between non-adjacent clusters;
2. bootstrapping to enlarge the sample size.

I learnt a lot during the numerical experiments and composing the report. I find your course and lecture notes very helpful in terms of formulating and analyzing the model. If I have opportunities in the future, I will try to improve my current 'toy' models.

References

1. Wasserman, Larry. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

2. Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol.1. No.10. New York: Springer series in statistics, 2001.

APPENDIX

MATLAB Code

1. Code for clustering and preconditioning of the data.

```
%% data input

clearvars;
load project.mat
% data contains three columns: (month, latitude, longitude)
% here latitude is always positive which means N
% here longitude is always negative which means W

%% clustering
% we are going to cluster the data based on the coordinates
% method: kmeans
d = data(:,[3,2]); % position data
N = 4; % total number of clusters

%figure
%scatter(d(1:1:end,1), d(1:1:end,2))

opts = statset('Display','final');
[idx,C] = kmeans(d,N,'Distance','cityblock',...
    'Replicates',5,'Options',opts);
% idx contains the cluster number of each data point


figure;
plot(d(idx==1,1),d(idx==1,2),'r.','MarkerSize',11)
hold on
plot(d(idx==2,1),d(idx==2,2),'m.','MarkerSize',11)
plot(d(idx==3,1),d(idx==3,2),'g.','MarkerSize',11)
plot(d(idx==4,1),d(idx==4,2),'y.','MarkerSize',11)

plot(C(:,1),C(:,2),'kx',...
     'MarkerSize',14,'LineWidth',2)
legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Centroids',...
       'Location','NW')
title 'Cluster Assignments and Centroids of the Data'
ylim([41.6,42.1])
xlim([-87.9, -87.5])
xlabel('longitude')
ylabel('latitude')
hold off

% add cluster into the data

c_data = [idx, data];
```

```
% c_data contains four columns: cluster, month, latitude, longitude


%% decompose the data into model and test data
% we are going to take the first 9 months data as observations
% and the last 3 months data as test data

m_data = c_data(:,2);
[B,I] = sort(m_data); % I is the index
c_data = c_data(I,:);
flag = find(B==9,1,'last'); % find the last data (sorted) index s.t. month=9

% split
m_data = c_data(1:flag,:); % training data
t_data = c_data(flag+1:end,:); % test data
```

Listing 1: Clustering and preconditioning of the data

2. Code for monthly forecast model.

```
%% Main: Regression to estimate the transition matrix

%load the data
clearvars;
load clustered_data.mat
% m_data are used for training (containing data from m1-m9)
% t_data are used for test (containing data from m10-m12)

data = m_data(:,[1,2]); % we only need the first two columns
N = max(data(:,1)); % total number of clusters

%% compute the density

rho = zeros(N,9); % density matrix

for j = 1:9
    index = find(data(:,2)==j);
    temp = data(index,:); % take out data from the same month
    c_count = zeros(N,1);
    for k = 1:N
        flag = find(temp(:,1)==k);
        c_count(k,1) = length(flag);
    end
    rho(:,j) = c_count./sum(c_count); % normalization
end

X = rho(:,1:8); % input
Y = rho(:,2:9); % output
P = Y/X; %least square solution


```

```
%% compute the regression error

% norm of the density vector

n_rho = zeros(1,9);

for j = 1:9
   n_rho(1, j) = norm(rho(:,j));
end

% local error

loc_err = zeros(N, 8);
loc_res = zeros(1,8);

for j=1:8
   loc_err(:,j) = rho(:,j+1)-P*rho(:,j);
   loc_res(1,j) = norm(loc_err(:,j));
end

% global error

glob_err = zeros(N, 8);
glob_res = zeros(1, 8);
rho_0 = rho(:,1);
for j = 1:8
    rho_0 = P*rho_0;
    glob_err(:,j) = rho(:,j+1) - rho_0;
    glob_res(1,j) = norm(glob_err(:,j));

end

%plot the result

figure
plot(2:1:9,loc_res./n_rho(2:end),'b--o','linewidth',2)
hold on
plot(2:1:9,glob_res./n_rho(2:end),'r--*','linewidth',2)
hold off
xticks([1 2 3 4 5 6 7 8 9])
xticklabels({'Jan','Feb','Mar','Apr','May', 'Jun', 'Jul', 'Aug', 'Sep'})
legend('local','global')
title('Relative Regression Error')

%% compute the predication error

% we are going to use the transition matrix to predict density in last
% three months

p_rho = zeros(N,3); % prediction
```

```
rho_0 = rho(:,end); % density at m9

for j = 1:3
    rho_0 = P*rho_0;
    p_rho(:,j) = rho_0;
end

% compute the true density

data = t_data(:,[1,2]); % we only need the first two columns

t_rho = zeros(N,3);

for j = 10:12 % last three months
    index = find(data(:,2)==j);
    temp = data(index,:); % take out data from the same month
    c_count = zeros(N,1);
    for k = 1:N
        flag = find(temp(:,1)==k);
        c_count(k,1) = length(flag);
    end
    t_rho(:,j-9) = c_count./sum(c_count);
end

% plot the prediction and compare with true density
figure
for k = 1:4
subplot(2,2,k)
plot(10:1:12, p_rho(k,:),'--*','linewidth',2)
hold on
plot(10:1:12, t_rho(k,:),'--o','linewidth',2)
hold off
xticks([10 11 12])
xticklabels({'Oct','Nov','Dec'})
end

legend('Model','True')
suptitle('Model Prediction')
```

Listing 2: Monthly forecast model

3. Code for daily forecast model.

```
%% Main: Markov Chain Model

% input data
clearvars;
close all;
% data structure: (latitude, longtitude, day)
load data_day.mat

```

```
%% clustering

% we are going to cluster the data based on the coordinates
% method: kmeans
% total number of clusters: 2

d = data_day(:,[2,1]); % only requires position data
N = 2; % total number of clusters

opts = statset('Display','final');
[idx,C] = kmeans(d,N,'Distance','cityblock',...
    'Replicates',5,'Options',opts);
% idx contains the cluster number of each data point(positioin)


figure
plot(d(idx==1,1),d(idx==1,2),'m.','MarkerSize',11)
hold on
plot(d(idx==2,1),d(idx==2,2),'b.','MarkerSize',11)
plot(C(:,1),C(:,2),'kx','MarkerSize',14,'LineWidth',2)
hold off
legend('Cluster 1','Cluster 2','Centroids','Location','SE')
title 'Cluster Assignments and Centroids of the Data'
ylim([41.6,42.1])
xlim([-87.9, -87.5])
xlabel('longitude')
ylabel('latitude')

%% kde on each cluster to determine the threshold
% find the thresholds to classify the low-mid-hight state
% count the number of crimes each day on each cluster

d = c_data(:,[1,4]);% (cluster, day)
min_day = round(min(d(:,2))); % min day
max_day = round(max(d(:,2))); % max day

% collect the number of crimes in each cluster each day
c_count = zeros(max_day-min_day+1,2);

flag = 1;
for j = min_day:max_day
    index = find(d(:,2)==j);
    temp = c_data(index,:);
    for k = 1:2
        temp1 = find(temp(:,1)==k);
        c_count(flag,k) = length(temp1);
    end
    flag = flag+1;
end

```

```
% plot the results
figure
histogram(c_count(:,1),25,'Normalization','Probability')
hold on
ksdensity(c_count(:,1),'Support',[0,35],'Function','pdf',...
'NumPoints',50)
hold off
xlabel('Number of Crimes')
title('Histogram v.s. Kernel Density Estimates: Cluster-1')

figure
histogram(c_count(:,2),25,'Normalization','Probability')
hold on
ksdensity(c_count(:,2),'Support',[0,35],'Function','pdf',...
'NumPoints',50)
hold off
xlabel('Number of Crimes')
title('Histogram v.s. Kernel Density Estimates: Cluster-2')

t_1 = 0.3.*ones(1,31); t_2 = 0.7.*ones(1,31);

% use the icdf to select the threshold values manually
figure
ksdensity(c_count(:,1),'Support',[0,35],'Function','icdf',...
'NumPoints',50)
hold on
ksdensity(c_count(:,2),'Support',[0,35],'Function','icdf',...
'NumPoints',50)
plot(t_1, 0:1:30,'k--','linewidth',2)
plot(t_2, 0:1:30,'k--','linewidth',2)
hold off
ylabel('Number of Crimes')
title('Estimated ICDF Based on KDE')
legend('Cluster-1', 'Cluster-2')
datacursormode on

%% classify state L-M-H for each date
% the two thresholds of c-1 are 8.5 and 12.5
% the two thresholds of c-2 are 11.5 and 17.5

% assign state
% L(Low)->0, M(Medium)->1, H(High)->2

N = size(c_count,1); %total number of days
state = zeros(N,2);

for j=1:N
   if c_count(j,1) < 8.5
       state(j,1) = 0; %low
   elseif c_count(j,1) < 12.5
```

```
        state(j,1) = 1; %medium
    else
        state(j,1) = 2; %high
    end

    if c_count(j,2) < 11.5
        state(j,2) = 0; %low
    elseif c_count(j,2) < 17.5
        state(j,2) = 1; %medium
    else
        state(j,2) = 2; %high
    end


end

% check the histogram of the state
% make sure that low, medium, and high take roughly 30%, 40%, and 30% resp.

figure

histogram(state(:,1),3,'Normalization','Probability')
xticks([0.25, 1, 1.75])
xticklabels({'Low','Medium','High'})
title('Cluster-1')

figure
histogram(state(:,2),3,'Normalization','Probability')
xticks([0.25, 1, 1.75])
xticklabels({'Low','Medium','High'})
title('Cluster-2')

%% estimate the transition matrix and determine the stationary density

% couple the X_n and X_n+1
c_state = [state(2:end,:) state(1:end-1,:)]; %(X_{n+1}, X_n) n = 1:end-1

% count
P = zeros(9,9);
flag1 = 1;
flag2 = 1;
for j = 0:2
    for k = 0:2
        for l = 0:2
            for m = 0:2
                temp = c_state - [j k l m];
                for r = 1:4
                    % find zero in each component
                    index = find(~temp(:,r));
                    temp = temp(index,:);
```

```
                end
                P(flag1,flag2) = length(index);
                flag2 = mod(flag2,9)+1;
            end
        end
        flag1 = flag1+1;
    end
end

% normalized to get the transition matrix
P = P./sum(P);

% compute the stationary distribution

[L,D] = eig(P);% colums of L are the eigenvectors, diag of D are eigenvectors
rho_eq = L(:,1)./sum(L(:,1)); % normalize to obtain a density vector


%% varification using the histogram on the state space

rho_state = zeros(9,1);
flag =1;
for j = 0:2
    for k = 0:2
        temp = state - [j,k];
        for l=1:2
            index = find(~temp(:,1));
            temp = temp(index,:);
        end
        rho_state(flag) = length(index);
        flag = flag+1;
    end
end
rho_state = rho_state./sum(rho_state);

% visualize the rho_eq and rho_state
figure
plot(1:9,rho_eq, '*--', 'linewidth',2)
hold on
plot(1:9,rho_state,'*--','linewidth',2)
xticks([1 2 3 4 5 6 7 8 9])
xticklabels({'Low-Low','Low-Medium','Low-High',...
    'Medium-Low', 'Medium-Medium', 'Medium-High','High-Low','High-Medium','High-High'
    })
legend('Stationary','Sample')
title('Stationary Density v.s. Sample Density on the State Space')
```

Listing 3: Daily forecast model