ENGR6991: Project Report

Emergency 911 Calls:

Data Processing and Analysis

Lin Hongyi

40016844

Abstract

Mobile phones can be used in countries with different emergency numbers. 9-1-1 is an emergency telephone number for the North American Numbering Plan (NANP), one of eight N11 codes. Like other emergency numbers around the world, this number is intended for use in emergency circumstances only, and using it for any other purpose (such as making false or prank calls) is a crime in certain jurisdictions. Therefore, it is important to consider the number of emergency call. The project uses the data of emergency (911) calls for Montgomery County, PA as an example to explore the possibility of building a station for handling emergency events. The data includes three types: Fire, Traffic and EMS. In this project, I only consider the Traffic Emergency, which is the most popular event in our daily life. To analyze the data, I apply different algorithm, like K-means Algorithm, Kalman Filters Algorithm and Particle Filter Algorithm to process the data. In the end, I also compare each results to draw the best place for building a traffic station, which can improve the speed of handling emergency events of traffic and improve the efficient. The project can also be a reference of other events.

Key Words: *Emergency Call 911, K-Means algorithm, Kalman Filter, Particle Filter, Python, Data Analyze, Data Process.*

1. Introduction

Before the 1960s, the United States didn't have one universal phone number for Americans to call if they needed help from the police or fire department. In the case of large cities, there Ire often multiple police and fire departments covering different areas. Needless to say, this system wasn't optimized to get emergency help where it needed to go very quickly.

To solve this problem, the National Fire Chief's Association suggested a national emergency phone number in 1957. A report to President Johnson's Commission on Law Enforcement and Administration of Justice suggested that a single telephone number should be designated for callers to use in emergencies nationwide, or at least in major cities. To make this universal emergency number a reality, the Federal Communications Commission (FCC) partnered with the American Telephone and Telegraph Company (also known as AT&T) in late 1967 to figure out what the number should be. After mulling it over, AT&T proposed in 1968 that the numbers 9-1-1 should make up the new universal emergency phone number.

The phone number 9-1-1 is short, easy to remember, and can be dialed relatively quickly given the few digits. In addition, the fact that it was only three digits meant the number could easily be distinguished from other, normal phone numbers.

Congress supported AT&T's proposal for 9-1-1 as the national emergency number and passed legislation to that affect. Just over ten years after Congress established 9-1-1 as the country's universal emergency phone number, approximately 26% of United States citizens could dial 9-1-1 and be connected with their local emergency services. In 1989, that number had risen only to 50%. However, just a decade after that, it rose to 93% of the country. Today, approximately 99% of people in the United States have access to the 9-1-1 emergency phone number system.

Nowadays, 9-1-1 is for emergencies where health, safety or property is in immediate jeopardy, or there is a crime in progress. Therefore, each community is required to have a police station, fire station and hospital where can offer emergency help. However, building these stations might be costly, according to a recent report, it needs nearly 10,000,000 for a hospital, 600,000 for a fire station and 200,000 for a policy station. Therefore, it is important to find a best area to establish the station.

In this paper, I use the data of emergency (911) call: Fire, Traffic, EMS for Montgomery County, PA to analyze the best location to build the station.

Subsequently, I make a spatial model to analyze the data. I suppose that each call is regarded as a sensor and each content as sensory information. These virtual sensors, which I call social sensors, are of a huge variety and have various characteristics. Because, some people might be moving when they are calling for help. About 70 percent of 9-1-1 calls came from cell phones in 2014 and finding out where the calls came from required triangulation. A USA Today study showed that where information was compiled on the subject, many of the calls from cell phones did not include information allowing the caller to be located. Chances of getting as close as 100 feet Ire higher in areas with more towers. But if a call was made from a large building, even that would not be enough to precisely locate the caller. New federal rules, which service providers helped with, require location information for 40 percent of calls by 2017 and 80 percent by 2021.

In this project, I process and analyze the data first to find the most popular problem to solve. Then I apply Kalman filters and particle filters, which are widely used for location estimation. Also, according to the meanings of Kalman algorithm, it is easier to find the most density place of calling 911 and it is helpful to find out the best location of building the station. In addition, I also use K-means algorithm to compare the efficiency of different algorithm. Moreover, the model can be used to analyze different data of other same events.

This paper is organized as follows: In the next section, I explain the data first, followed by the spatial model by using Kalman filter, Particle filter and K-means algorithm in section 3. In section 4, I describe the experiments and evaluation. Finally, I will discuss about the advantage and disadvantage of the project and conclude the paper.

2. Dataset Process

In this paper, I use the data of Montgomery County, PA to process. The data is taken directly from Montco's Active Incident page. The data is collected everyone 5 minutes, but it's only loaded up every few hours here. Consequently, I only use the data of 2016 to process.

The content of the dataset is from 2015-12-10 to 2017-08-18:

- lat: String variable, Latitude
- lng: String variable, Longitude
- desc: String variable, Description of the Emergency Call
- zip: String variable, Zip code
- title: String variable, Title
- timeStamp: String variable, YYYY-MM-DD HH:MM:SS
- twp: String variable, Township
- addr: String variable, Address
- e: String variable, Dummy variable (always 1)

	lat	**lng**	**desc**	**zip**	**title**	**timeStamp**	**twp**	**addr**	**e**
timeStamp									
2016-01-01 00:10:08	40.121354	-75.363829	ROSEMONT AVE & W WASHINGTON ST; NORRISTOWN; S...	19401	EMS: ASSAULT VICTIM	2016-01-01 00:10:08	NORRISTOWN	ROSEMONT AVE & W WASHINGTON ST	1
2016-01-01 00:14:45	40.140505	-75.310874	GERMANTOWN PIKE & HANNAH AVE; EAST NORRITON; ...	19401	EMS: FALL VICTIM	2016-01-01 00:14:45	EAST NORRITON	GERMANTOWN PIKE & HANNAH AVE	1
2016-01-01 00:20:43	40.246837	-75.681381	VINE ST & CENTER ST; WEST POTTSGROVE; Station...	19464	EMS: ABDOMINAL PAINS	2016-01-01 00:20:43	WEST POTTSGROVE	VINE ST & CENTER ST	1
2016-01-01 00:25:30	40.097222	-75.376195	MARK LN & DEAD END; UPPER MERION; Station 317...	NaN	EMS: ALTERED MENTAL STATUS	2016-01-01 00:25:30	UPPER MERION	MARK LN & DEAD END	1
2016-01-01 00:30:28	40.148432	-75.219812	BANNOCKBURN AVE & S SPRING GARDEN ST; AMBLER;...	19002	EMS: DIABETIC EMERGENCY	2016-01-01 00:30:28	AMBLER	BANNOCKBURN AVE & S SPRING GARDEN ST	1

Table 2.1

At first, I load the data by python using pandas and numpy function, and I create a table 2.1 as followed to make it more visual. Next, I enumerate all different title of the data, and get this result Figure 2.1 as followed.

```
Traffic: VEHICLE ACCIDENT -           45959
Traffic: DISABLED VEHICLE -           14351
Fire: FIRE ALARM                      11027
EMS: RESPIRATORY EMERGENCY            10290
EMS: FALL VICTIM                       9948
EMS: CARDIAC EMERGENCY                 9914
EMS: VEHICLE ACCIDENT                  7858
Traffic: ROAD OBSTRUCTION -            6009
EMS: SUBJECT IN PAIN                   5534
EMS: HEAD INJURY                       5157
EMS: UNKNOWN MEDICAL EMERGENCY         3607
EMS: SYNCOPAL EPISODE                  3466
EMS: SEIZURES                          3302
Fire: VEHICLE ACCIDENT                 3293
```

Figure 2.1

Also, I integrate the number of three title: Traffic, Fire and EMS. Finally, I decided to use the data of Traffic Figure 2.2 and Figure 2.3. Because building a police station is more realistic in our daily life, and it is also more meaningful to solve traffic problem which can directly improve the efficient to solve traffic jam or other emergency event.

Next step is to find the area of the largest number of traffic problems. The result is also as followed. To make it more clear, I draw the plot Figure 2.4.

```
EMS        98725
Traffic    69481
Fire       29458
Name: Category, dtype: int64
```

Figure 2.2

```
LOWER MERION      7525
UPPER MERION      4935
ABINGTON          4017
CHELTENHAM        3572
PLYMOUTH          2775
UPPER DUBLIN      2546
UPPER MORELAND    2494
MONTGOMERY        2433
HORSHAM           2429
NORRISTOWN        2386
Name: twp, dtype: int64
```

Figure 2.3

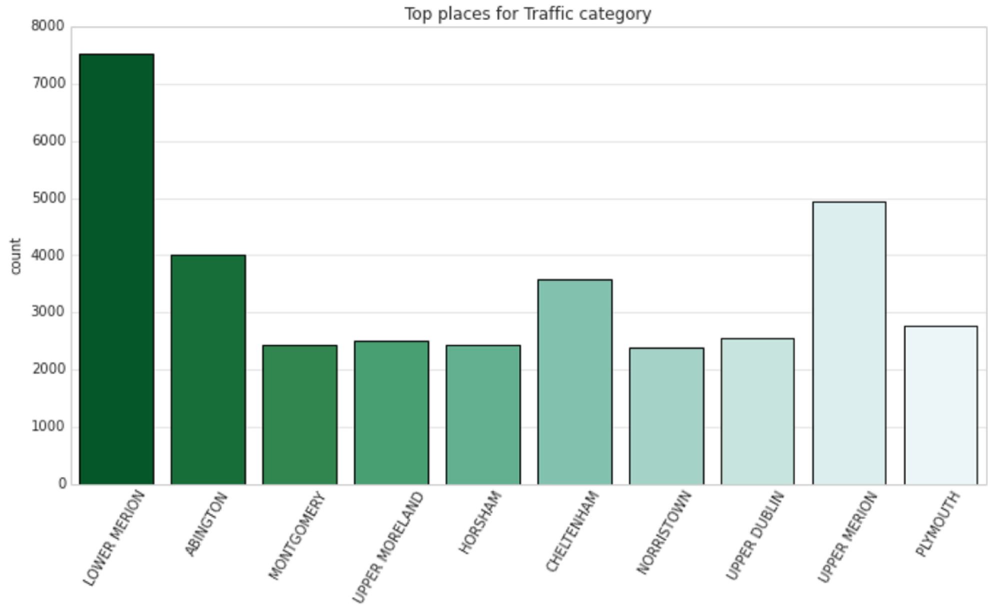

Figure 2.4

Consequently, I screen the data of Lower Merion to do next process, and also I eliminate the other date out of 2016, therefore the data is only during the period of 2016. I only use longitude and latitude, so I list the data in order by the date. At last, I use the data to draw the map of this area Figure 2.5, and the red point of area means the place of calling for help.

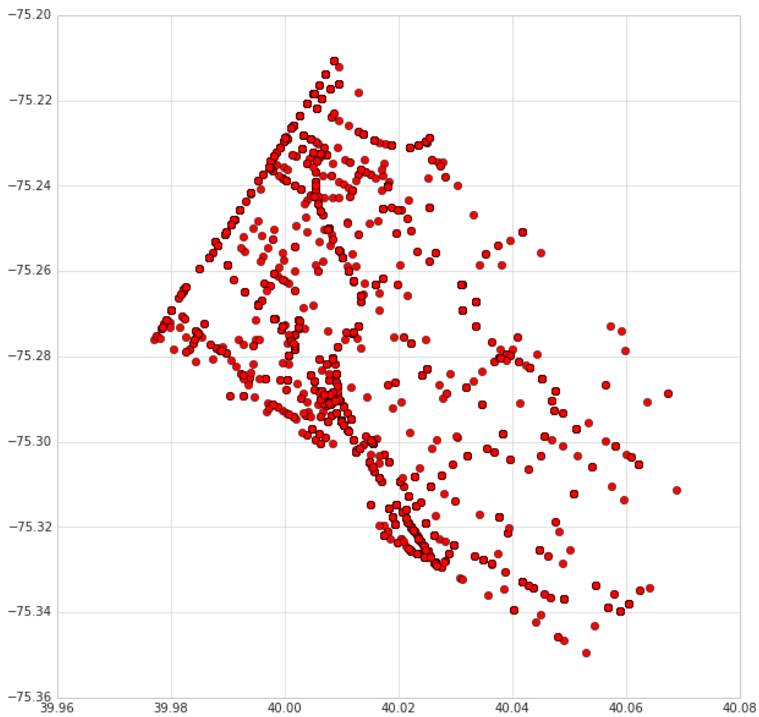

Figure 2.5

3. Spatial Model

Each call is associated with a location, how to estimate the area of the most efficient place for building a station is to find the central of the area. To solve the problem, several methods of Bayesian filters are proposed, such as Kalman filters and particle filters. For this project, I use Kalman filters and particle filters, both of which are widely used in location estimation.

3.1 Kalman Filter

Kalman filtering, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by using Bayesian inference and estimating a joint probability distribution over the variables for each time frame. Also the Kalman filter assumes that the posterior density at every time step is Gaussian and that it is therefore parameterized by a mean and covariance.

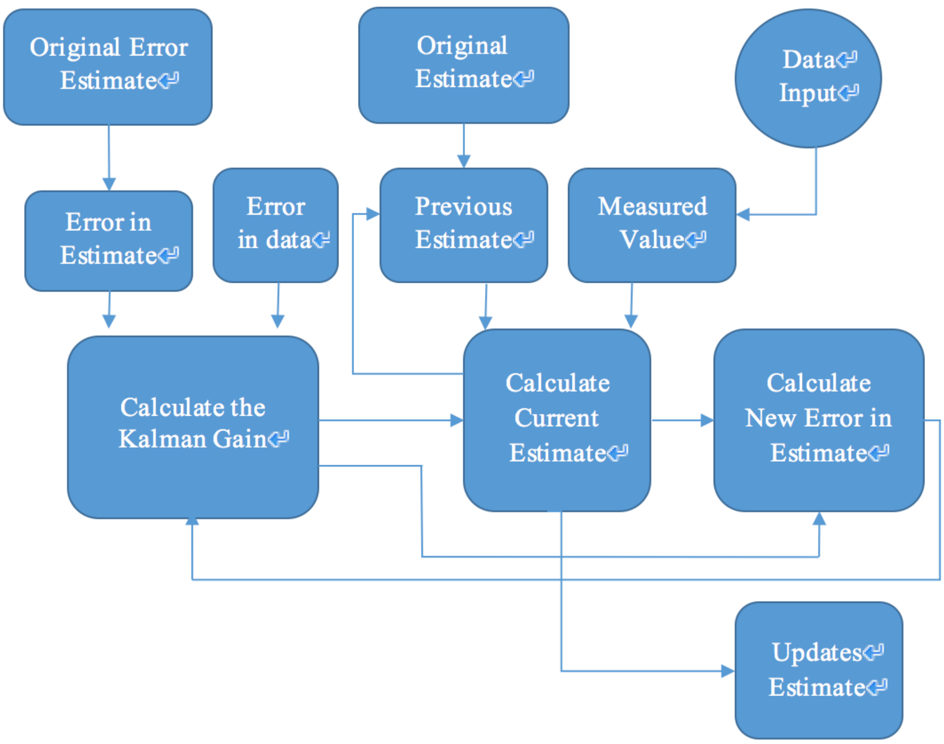

Figure 3.1 The Flow Chart of Kalman Filter

The algorithm works in a two-step process. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement (necessarily corrupted with some amount of error, including random noise) is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. The algorithm is recursive. It can run in real time, using only the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is required.

In this paper, I use 2-dimension Kalman filter. We can write it as $x_t = F_t x_{t-1} + v_{t-1}$ and $z_t = H_t x_t + n_t$. F_k and H_k are known matrices defining the linear functions. The covariance of v_{k-1} and n_k are Q_{t-1} and R_k.

The Kalman filter algorithm can consequently be viewed as the following recursive relation:

$$p(x_{t-1}|z_{t-1}) = \mathcal{N}(x_{t-1}; m_{t-1|t-1}, P_{t-1|t-1})$$

$$p(x_t|z_{t-1}) = \mathcal{N}(x_t; m_{t|t-1}, P_{t|t-1})$$

$$p(x_t|z_t) = \mathcal{N}(x_t; m_{t|t}, P_{t|t})$$

Where $m_{t|t-1} = F_t m_{t-1|t-1}$, $P_{t|t-1} = Q_t + F_t P_{t-1|t-1} F_t^T$, $m_{t|t} = m_{t-1|t-1} + K_t(z_t - H_t m_{t|t-1})$, and $P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}$, and where $\mathcal{N}(x; m, P)$ is a Gaussian density with argument x, mean m, covariance P, and for which the following are true: $K_t = P_{t|t-1} H_t^T S_t^{-1}$, and $S_t = H_t P_{t|t-1} H_t^T + R_t$. It works better in a linear Gaussian environment.

It is important to construct a good model and parameters when utilizing Kalman filters. In this project, it is not necessary to consider the time transition property, therefore we only use location information (Longitude $d_{x_t}$and Latitude d_{y_t}), we set $x_t = (d_{x_t}, d_{y_t})^t$; $z_t = (d_{x_t}, d_{y_t})$, $F = I_2$, $H = I_2$ and $u_t = 0$, $Q_t = 0$, $R_t = [\sigma^2]$, and $n_t = \mathcal{N}(0; R_t)$.

3.2 Particle Filter

Particle filters method is based on Monte Carlo methodologies to solve filtering problems arising in signal processing and Bayesian statistical inference. The filtering problem consists of estimating the internal states in dynamical systems when partial observations are made, and random perturbations are present in the sensors as well as in the dynamical system. The objective is to compute the conditional probability (a.k.a. posterior distributions) of the states of some Markov process, given some noisy and partial observations.

For location estimation, it maintains a probability distribution for the location estimation at time t, designated as the belief $Bel(x_t) = \{x_t^i, w_t^i\}, i = 1 \dots n$. Each x_t^i is a discrete hypothesis about the location of the object. The w_t^i are non-negative weights, called importance factors, which sum to one. The Sequential Importance Sampling (SIS) algorithm is a Monte Carlo method that forms the basis for particle filters. The SIS algorithm consists of recursive propagation of the weights and support points as each measurement is received sequentially. The step of the algorithm I employ is as followed.

1. Initialization: Calculate the weight distribution $D_w(x, y)$ from the calling users geographic distribution in Lower Merion.
2. Generation: Generate a particle set $S_0 = (s_{0,0}, s_{0,1}, s_{0,2}, \dots, s_{0,N-1})$ and allocate them on map evenly: particle $s_{0,k} = (x_{0,k}, y_{0,k}, weight_{0,k})$, x corresponds to the longitude and y corresponds to the latitude. Then Weight them based on weight distribution $D_w(x, y)$.
3. Re-sampling: Re-sample N particles from a particle set S_t using weights of each particles and allocate them on the map. Generate a new particle set S_{t+1} and weight them based on weight distribution $D_w(x, y)$.
4. Prediction: Predict the next state of a particle set S_t from the Newton's motion equation.
5. Weighing: Re-calculate the weight of S_t by measurement.

6. Measurement: Calculate the current object location $o(x_t, y_t)$ by the average of $s(x_t, y_t) \in S_t$.
7. Iteration: Iterate Step 3,4,5 and 6 until convergence.

3.3 K-Means Algorithm

k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. The most common algorithm uses an iterative refinement technique.

k-means clustering, or Lloyd's algorithm, is an iterative, data-partitioning algorithm that assigns n observations to exactly one of k clusters defined by centroids, where k is chosen before the algorithm starts. The algorithm proceeds as follows:

1. Choose k initial cluster centers (centroid).
2. Compute point-to-cluster-centroid distances of all observations to each centroid.
3. There are two ways to proceed:

   Batch update — Assign each observation to the cluster with the closest centroid.

   Online update — Individually assign observations to a different centroid if the reassignment decreases the sum of the within-cluster, sum-of-squares point-to-cluster-centroid distances.
4. Compute the average of the observations in each cluster to obtain *k* new centroid locations.
5. Repeat steps 2 through 4 until cluster assignments do not change, or the maximum number of iterations is reached.

4. Results

In this section, we describe the experimental results and evaluation of emergency call location estimation. The location information of emergency call is obtained and used for location estimation of the place.

4.1 Results of Kalman filter

The figure shows the location of all emergency call in Lower Merion using the red point. We can find that the original data is a little irregular, therefore we using Kalman Algorithm to find the most density area of this place. After Kalman filter, the location is shown by green point in the map Figure 4.1.1, which the longitude and latitude are closing to a certain value shown on the Figure 4.1.2.

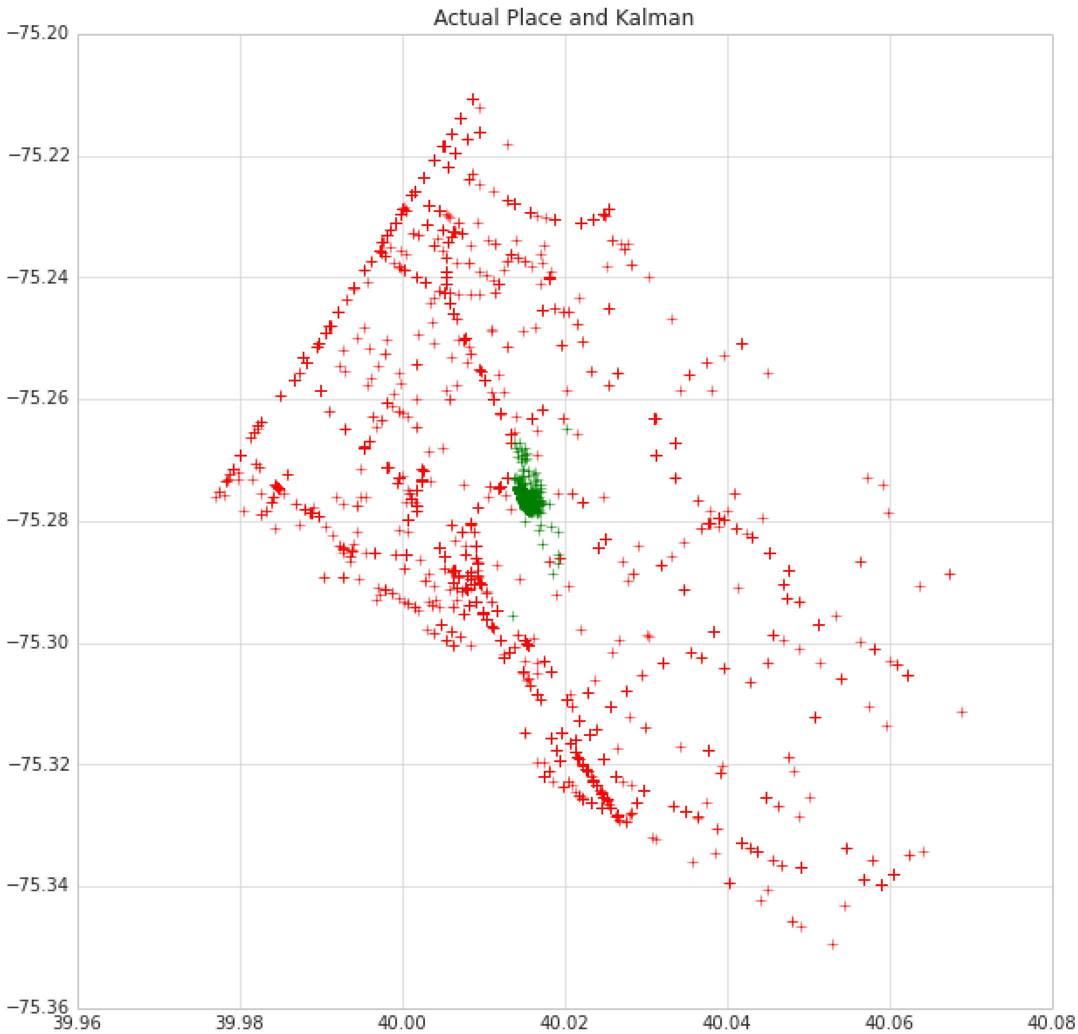

Figure 4.1.1

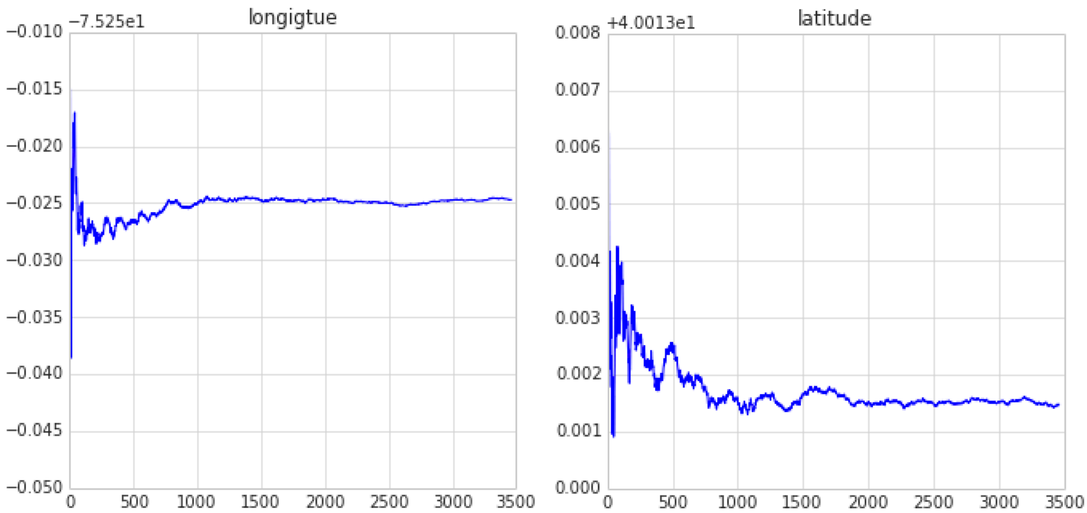

Figure 4.1.2

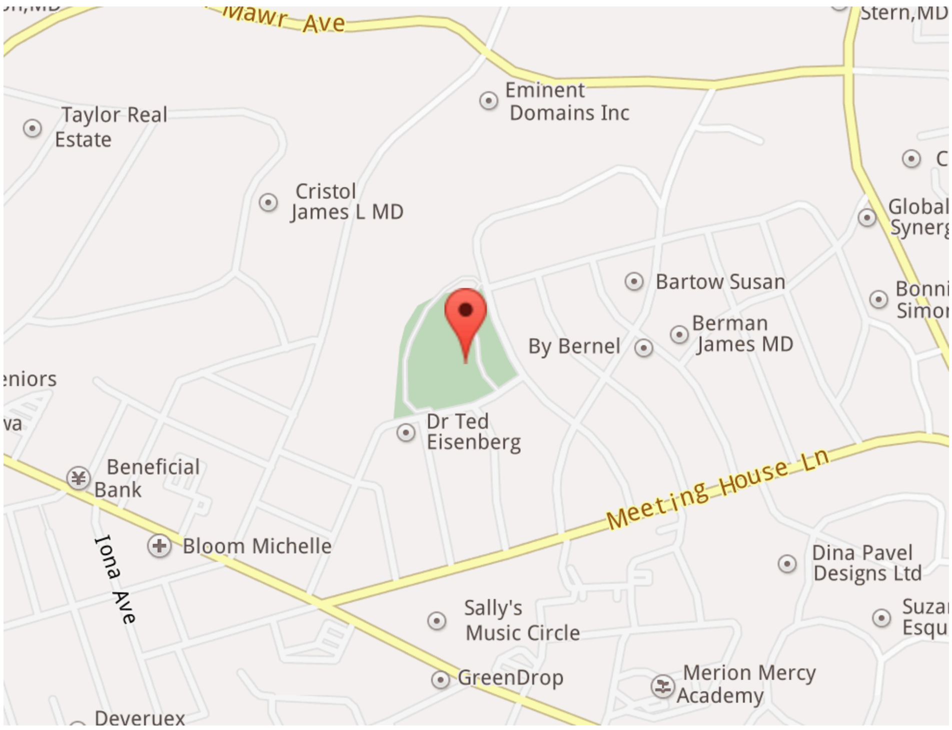

Figure 4.1.3

Therefore, the most density area is (40.01315, -75.2525) Figure 4.1.3. Also I search this place by using Google map, it is a park shown on the map. It is clearly that this could be a one of the best place to build the station.

4.2 Results of Particle filter

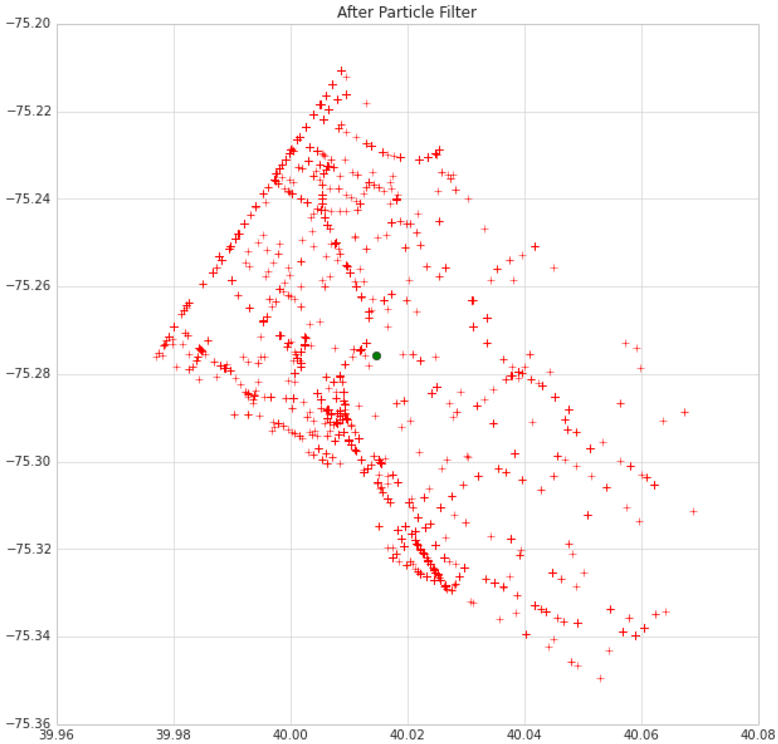

Figure 4.2.1

After using Particle Filter, the most density one is a point of the map Figure 4.2.1 shown. Also, the point is (40.01469274, -75.27578906) Figure 4.2.2, it is more accurate then Kalman Filter. It is another place; however, it is close to the area which is obtained by using Kalman Filter.

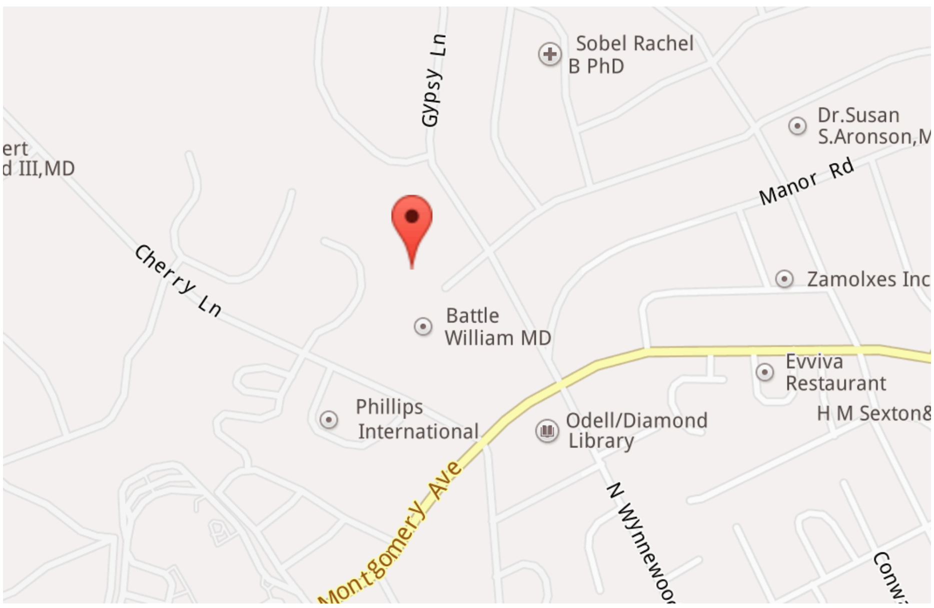

Figure 4.2.2

4.3 Results of K-means Algorithm

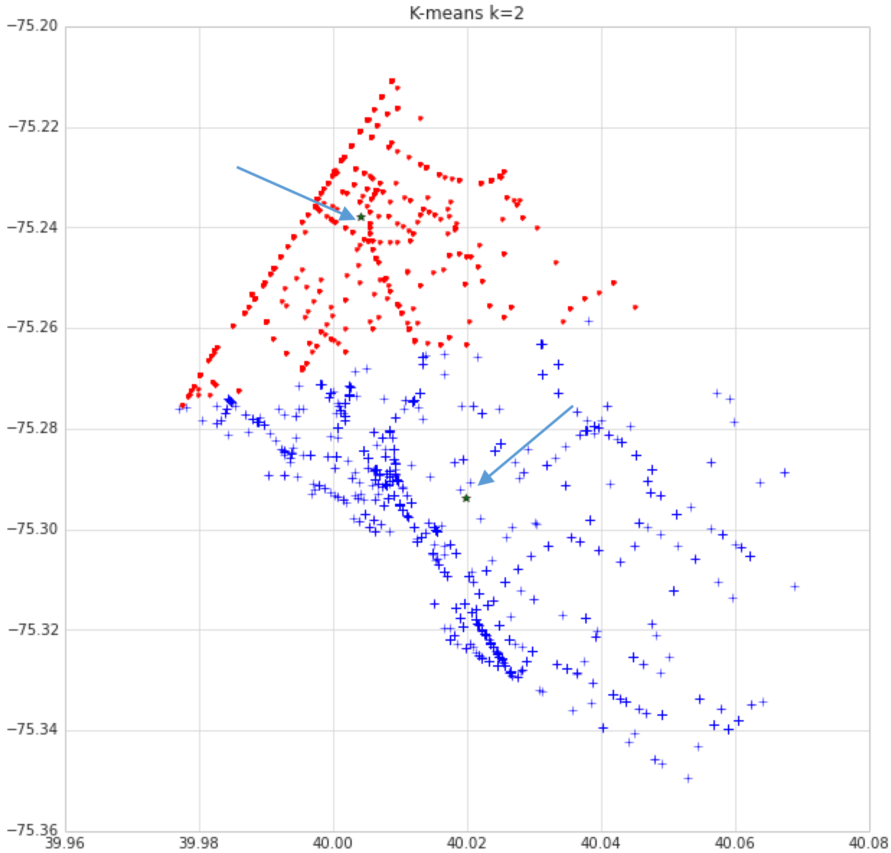

Figure 4.3.1

The figure presents the location by using K-means Algorithm, in this project, considering the practicability of construction, I only choice k=2 and k=3 to compare.

Figure 4.3.2 Point 1 (40.0043076, -75.23802717)

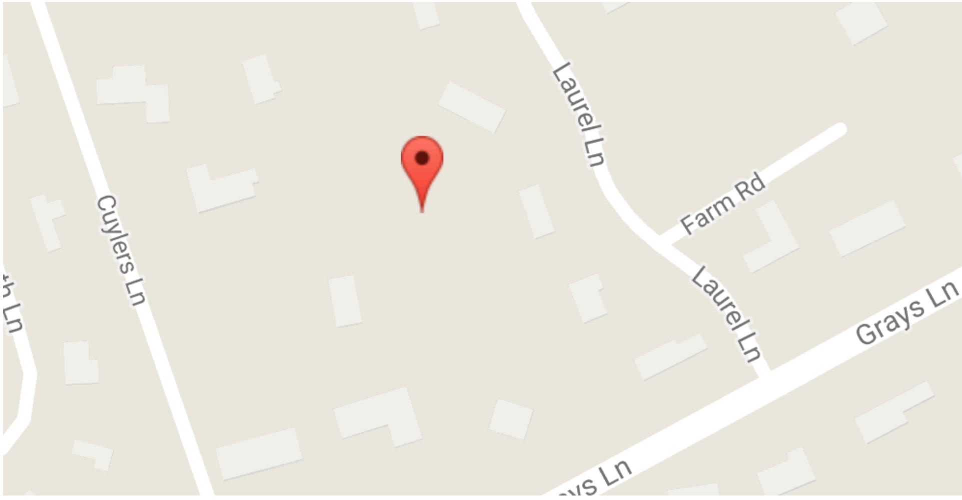

Figure 4.3.3 Point 2 (40.01982991, -75.29397644)

The Figure 4.3.1 is k=2 and the other one Figure 4.3.4 is k=3. When k=2, the points are (40.0043076, -75.23802717) and (40.01982991, -75.29397644). For k=3, the points are (40.03089409, -75.32008292), (40.01379396, -75.28149707) and (40.00513451, -75.23377827).

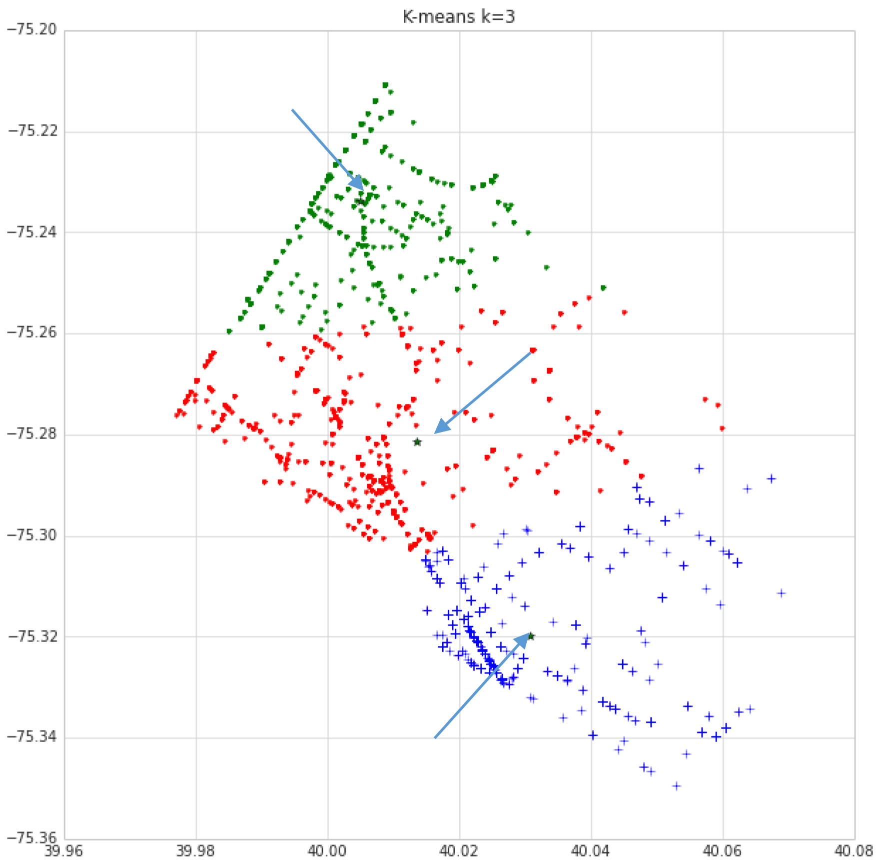

Figure 4.3.4

To sum up, it is true that building 3 stations can improve the efficiency of solving the traffic problem, however, considering the cost of station and the feasibility of building the station in 3 different place, it is unreality. Therefore, the two maps show the different place which can offer help for reference to governments to choose the best place.

5. Conclusion

As described in this paper, I use different algorithm to compare in order to find a better place to build the station. The Kalman filter can gain an area which represents the most density place, however compared with the particle filter, which is just a point, the results of Kalman filter is less accurate than the results of particle. As for K-means algorithm, it is not like the former algorithms, it divided the area into 2 and 3 parts to obtain different central places of the area. In my opinion, to some extent, the K-means Algorithm works better in solving this problem, since it can have more choices for government to choose, and the other two algorithms only offer one area to select. However, in further study, it is necessary to include other factors, such as the distance between the station and the place of farthest, the speed of police car and the time of arriving time when driving, etc.

Reference

1. Maskell S, Gordon N. A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking[C]// Target Tracking: Algorithms and Applications. IET, 2002:2/1-2/15 vol.2.
2. Ray A N, Chakravarthy C V. System and method for providing location information to a public safety answering point during an emergency 911 call from a WiFi handset: US, US 8290470 B2[P]. 2012.
3. Welch G, Bishop G. An Introduction to the Kalman Filter[M]. University of North Carolina at Chapel Hill, 2001.
4. Merwe R V D, Doucet A, Freitas N D, et al. The unscented particle filter[C]// International Conference on Neural Information Processing Systems. MIT Press, 2000:563-569.
5. Fox D, Hightower J, Liao L, et al. Bayesian Filtering for Location Estimation[J]. IEEE Pervasive Computing, 2003, 2(3):24-33.
6. Hartigan J A, Wong M A. Algorithm AS 136: A K-Means Clustering Algorithm[J]. Journal of the Royal Statistical Society, 1979, 28(1):100-108.
7. Alriksson P, Rantzer A. Experimental Evaluation of a Distributed Kalman Filter Algorithm[C]// Decision and Control, 2007, IEEE Conference on. IEEE, 2007:5499-5504.
8. Shaheen S A, Schwartz A, Wipyewski K. Policy Considerations for Carsharing and Station Cars: Monitoring Growth, Trends, and Overall Impacts[J]. Transportation Research Record Journal of the Transportation Research Board, 2004, 1887(1):128-136.
9. Oliphant T E. Python for Scientific Computing[J]. Computing in Science & Engineering, 2007, 9(3):10-20.
10. Kristensen M R B, Lund S A F, Blum T, et al. Separating NumPy API from Implementation[C]// Pyhpc 2014 -, Workshop on Python for High PERFORMANCE and Scientific Computing. 2014.