EN.601.475/675: Introduction to Machine Learning

Determine the More Efficient Implementation of Police Stations in Baltimore through Clustering

Bowen Deng, James Guo, Lisong He, Jiachen Wang

December 11, 2024

Contents

I Introduction

I.1 Background

According to a new article in 2024, Baltimore has been ranked as the "deadliest large city in the nation." [1] Especially in terms of large cities (*with more than 500,000* population), Baltimore has exceeded the danger of Memphis and Detroit, which used to be the worst in the past. [1]

The crime problem is a significant issue in Baltimore city, which threatens the surrounding industries and institutions. [2] According to the report, there have been exceptionally high amount of robbery, especially in terms of violence crime within the city. [1] Therefore, the variety of crime is something that worth getting into within the project.

From our data set, we noted that the crime count had been high up in the past years (∼50,000 cases each year). [3] Hence, we want to analyze about the efficiency of the police departments and look for a more optimal set up. We aim to **use Machine Learning to explore approaches that can potentially address the issues with crime in the city**.

I.1.1 Sources

We use our preliminary data set from the **Open Baltimore website** on the recorded crime data. [3] This is the data set from the police department of Baltimore in terms of the criminal cases in the city. On the webpage, we have access to a variety of variables, as displayed below as an excerpt:

RowID	CCNumber	CrimeDateTime	CrimeCode	Description	Weapon	Post	Gender	Age	Race	Location	Latitude	Longitude
138357	23L09461	12/31/2023 2:00:00 PM	6C	LARCENY		811				4600 EDMONDSON AVE	39.29341	-76.69569
⋮ *Further data are omitted for presentation purposes.* ⋮												

Table I.1. Excerpt from the crime data set on open Baltimore website [3].

In terms of the columns, we have removed a few rows that not interested and kept the most important rows. As we train the model for clustering, we have kept the following columns with corresponding purposes as explained:

Column	Reason for Keeping
CrimeDateTime	We use the date as a criterion of the selection of data for training and validating.
CrimeCode	This code is a representation of the seriousness of the crime.
Description	We assign weight based on the seriousness of the description of the crime.
Latitude & Longitude	This represents the location of the crimes as a subset of the $\mathbb{R}^2$ space.

Table I.2. Selected columns for analysis and the reasoning of choice.

The data includes crime data from 1023[1] up to the most recent weekday. [3] For the sake of the project, we utilize the data from **October 1st, 2024** to **November 6, 2024** to train the model, and use excerpts of data from **November 7, 2024** to **December 6, 2024** as the validation set. As a side note, there are about 130 cases each day during this period. [3]

[1] This is clearly an example of an mistake, and we will discuss more about this in III.1 Errors in Data Set.

Since the data is open source and continuously updated, such project has the potential to be trained further with more data fed into it.

I.1.2 Main Problem

With the given setup, we are looking for an optimal solution to address the crimes in Baltimore city. The main goals is to reduce the crime rate in the city with minimal amount of cost. While costs are quantifiable, but crime rate is not something that we can easily model based on our data set and implementation of the police stations, so we want to find alternations to model that.

According to a study about crime rate and police response, the police presence deters the happening of crime, that is, the better coverage of police through faster response time could help decrease the crime rate. [4] Hence, we are able to transform the crime rate from an abstract variable to the response time of police forces, which is a better to manage variable.

Therefore, we can define our problem, *equivalently*, as:

- **Efficiency**: We want to "span" the city up with less as possible police department to minimize the cost of operation.
- **Response time**: We want to minimize to total amount of response time for police to react to a crime.

While, we have established our problem, we would then want to think about ways to address the problem.

I.2 Research Procedure

The problem naturally falls into the category of **clusterings problem**, which is a type of **unsupervised learning**. Here, we consider the latitude and longitude as a subset of the $\mathbb{R}^2$ plane, and we find the clusters of the crime occurrences and distribute the police department with a given size to respective clusters to minimize the distance from department to the locations with denser and more serious crimes.

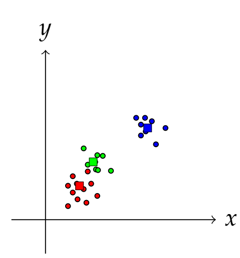

Figure I.3. A sample clustering and distribution based on crime(circles) and Police department(square)..

With no doubt, a simple cluster could give a police department setup, but such setup is giving a trivial set of centroids, which does not count in the environment specific to Baltimore city. Therefore, we will introduce the following heuristics in our project.

I.2.1 Alternations to Distances

The k-clustering algorithm is capable of clustering the points via different metric spaces (X, d). [5] The typical metric space is $\mathbb{R}^n$, $\| \bullet \|_2$, or the Euclidean distance. Recall the definition of a metric space, it must satisfy a few properties.

Definition I.2.1. Metric Space.

Let (X, d) be a metric space, then X is a set, and $d : X \to \mathbb{R}_{\geq 0}$ is the metric that must satisfy the following properties for any $x_1, x_2, x_3 \in X$:

1. **Positivity**: $d(x_1, x_2) \geq 0$,
2. **Definiteness**: $d(x_1, x_1) = 0$ if and only if $x_1 = 0$,
3. **Symmetry**: $d(x_1, x_2) = d(x_2, x_1)$, and
4. **Triangular inequality**: $d(x_1, x_3) \leq d(x_1, x_2) + d(x_2, x_3)$.

⌟

Even if our set is a subset of a Euclidean space ($\mathbb{R}^2$), there are various of different metric space, such as *discrete metric*, or the ℓ^p *norms*. Specifically, there have been use of Manhattan distance in many city settings to better fit urban layouts, or other metrics that we might could use. Moreover, we will also develop a *special metric* optimized for Baltimore city (accounting for sea in Inner Harbor area) based on ℓ^2 norm.

I.2.2 Weights in Clustering

In light of the fact that our data includes includes more fields, we would also want to assign weights on our clustering. Unlike standard clustering, assigning weights can change the setup significantly.

In the context of the project, we have thought about the **severity of crime**. In general, a **murder** is more sever than a **robbery**, and more sever than **larceny**. Specifically, as the one of the source indicates, there have been more robbery throughout the city and less murdering. [1] Thus, it is important to distribute the police forces corresponding to the severity of cases.

II Methodology

Here, we briefly discuss the further deployment of the project through the heuristics in details.

II.1 K-means Method

K-means clustering is a simple and effective method to associate points of a dataset with a class, *i.e.*, the clusters. With the setup, we want to the clusters to be unique for each data and mutually disjoint for the clusters.[5]

Algorithm II.1.1. K-Means Algorithms.

For the algorithm, we start by randomly selecting a set of points as the its centroid. Then, we iteratively do the follows:

1. Assign each data point to a cluster based on the **distance metric** from the point to the centroids.
2. Then, we calculate the new centroids of each cluster by taking the average of all the points that belong to the current cluster.

Aside, we break the above iteration only if one of the two conditions are fulfilled:

- When the number of iterations exceed the maximum number of iterations.
- When the movement of each cluster centroid is within a level of tolerance.

Hence, we will end up with a set of centroids, and each data point will also be classified into one of the centroids. [5] ⌟

Since points in our dataset are coordinates in **longitude** and **latitude**, it is interpreted as a subset of $\mathbb{R}^2$, with number of clusters, k, fixed, we have k centroids, namely:

$$\{\boldsymbol{\mu}_i\}_{i=1}^{k} \subset \mathbb{R}^2.$$

Suppose our data set X is composed of n crime cases. For the classification, we cluster the data by assigning each point $\mathbf{x}_i \in X$ to its closest center:

$$D_i := \arg\min_{1\leq i\leq k} d(\mathbf{x}_n, \boldsymbol{\mu}_i) \text{ for } 1 \leq i \leq n, \tag{$*$}$$

where d is the distance metric (Euclidean distance).

Following that, we can calculate the next iteration of centroids by computing the average value of all points assigned to them, *i.e.*, for each $1 \leq i \leq k$:

$$\boldsymbol{\mu}_i = \frac{1}{\sum_{D_j=i} 1} \sum_{D_j=i} \mathbf{x}_j. \tag{$**$}$$

Recall that we are iterating the above steps until convergence or enough amount of iterations. [5]

Proposition II.1.2. Effect of K-means Algorithm.

With the above setup, the **K-means algorithm** minimize the loss:

$$\sum_{i=1}^{n}\sum_{j=1}^{k} \mathbb{1}_{D_i}(j)\left[d(\mathbf{x}_i, \boldsymbol{\mu}_j)\right]^2.$$

The loss will be reduced in each step of iteration, and it will eventually converge to a optimal solution based on the selection of initial centroids. [5] ⌟

II.2 Minkowski Distance and ℓ^p Space

The first heuristic that we introduce is to modify the distance function. The Minkowski distance functions form a whole family of metrics, denoted ℓ^p metrics, depending on the parameter p.[6]

Definition II.2.1. Minkowski Distance.

Let $\mathbb{R}^n$ be a n-dimensional vector space, and suppose $(x_1, \ldots, x_n), (y_1, \ldots, y_n) \in \mathbb{R}^n$. For integer $p \geq 1$, we define:

$$L_p\big((x_1, \ldots, x_n), (y_1, \ldots, y_n)\big) := \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}. \text{ [6]}$$

It can be proven that the **Minkowski distance metric** is a metric, *i.e.*, it satisfies all the conditions in Definition I.2.1. [6]

Example II.2.2. Common Minkowski Distances.

Some common **Minkowski distances** are as followed:

- L_1, known as the **Manhattan distance** (also the City-Block distance), is namely:
$$L_1\big((x_1, \ldots, x_n), (y_1, \ldots, y_n)\big) = \sum_{i=1}^{n} |x_i - y_i|.$$
- L_2, or the typical **Euclidean distance**, is acquainted as:
$$L_2\big((x_1, \ldots, x_n), (y_1, \ldots, y_n)\big) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}.$$
- Another famous is the L_∞, which is the maximum distance, as it accounts for the maximum of all dimensions:
$$L_\infty\big((x_1, \ldots, x_n), (y_1, \ldots, y_n)\big) = \max_{i=1}^{n} |x_i - y_i|.$$

Specifically, the unit ball in $\mathbb{R}^2$ of these metrics can be represented as follows:

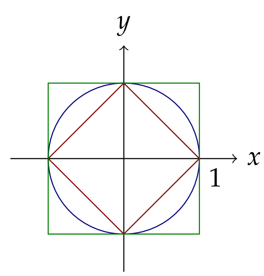

Figure II.1. Unit Sphere in L_1(red), L_2(blue), and L_∞(green).

Furthermore, we consider the ℓ^p spaces by lifting the restriction that p must be integer.

Definition II.2.3. ℓ^p Space Metric.

Consider $\mathbb{R}^n$ as a n-dimensional vector space (technically, it could be infinite dimensional with some constraints), and suppose $(x_1, \ldots, x_n), (y_1, \ldots, y_n) \in \mathbb{R}^n$. For $p \geq 1$, we define:

$$L_p\big((x_1, \ldots, x_n), (y_1, \ldots, y_n)\big) := \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}. \text{ [7]}$$

Even though this generalizes the case, it can still be proven that the ℓ^p **distance metric** is a metric for $p \geq 1$, *i.e.*, it satisfies all the conditions in Definition I.2.1. [7]

As we develop these foundations, we eventually modified the K-means algorithms at Algorithm II.1.1. For equation $(*)$, we modify it as:

$$D_i := \arg\min_{1\leq i\leq k} L^p(\mathbf{x}_n, \boldsymbol{\mu}_i) \text{ for } 1 \leq i \leq n,$$

where p does not necessarily have to be 2.

Here, we inspect the road map of Baltimore city, as follows:

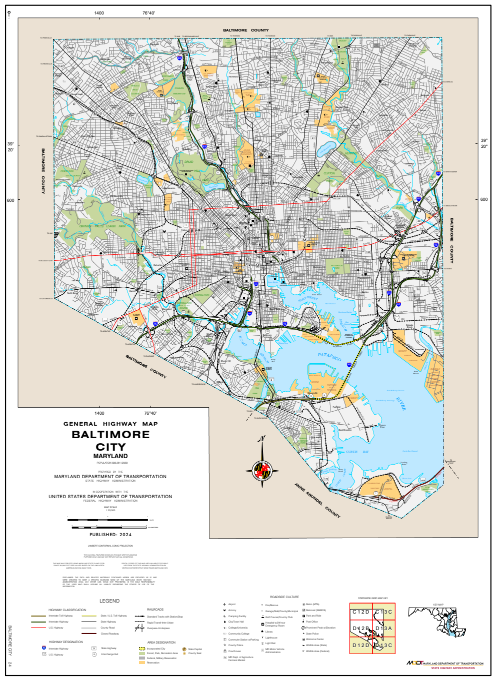

Figure II.2. General road map of Baltimore city [8].

It is clear that we cannot simply consider the road map of Baltimore as strict grids (corresponding to Manhattan distance) or as straight line (corresponding to Euclidean distance). Therefore, we will be evaluating the ℓ^p metric with $p = 1, 1.25, 1.5, 1.75$, and 2.

II.2.1 Optimized Metric from ℓ^2 Metric

Recall the Figure II.2, it is not hard to notice that the land of the city is not **totally connected**, having any ℓ^p distance metric cannot account for the cross of the watershed.[2] We want to account for this.

For the sake of discussion, we uses ℓ^2 metric as it is geometrically easy to manipulate. For simplicity, we also consider the setup in a more trivial case.

[2]Well, it is reasonable to assume that police cars cannot hop over the water.

Example II.2.4. Path of Non-Total Connected Set.

Consider the set $([0,4] \times [0,4]) \setminus \{(t,2) : 1 < t < 3\} \subset \mathbb{R}^2$, where it could be represented as follows:

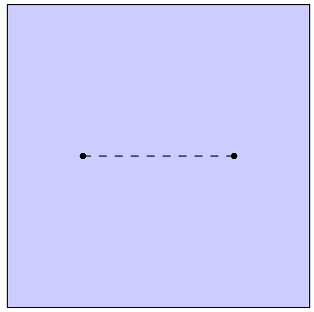

Figure II.3. The set $([0,4] \times [0,4]) \setminus \{(t,2) : 1 < t < 3\} \subset \mathbb{R}^2$.

Clearly, this set is not **totally connected**. If we consider some path on it:

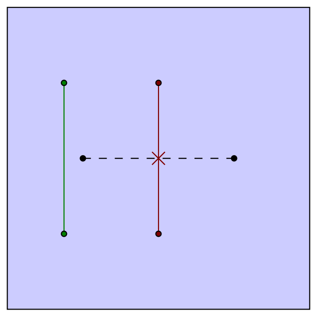

Figure II.4. Paths on total the disconnected set.

The green path is definitely valid as the connection line is a subset of the space, but red path is **not valid**, as it gets to the cross ($\times$) which is not accessible. Hence, we need to find an alternative path:

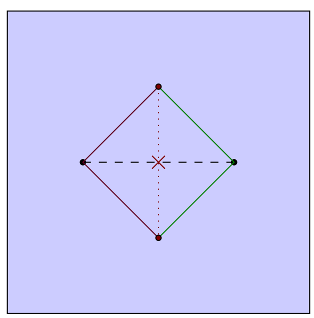

Figure II.5. Alternative path (purple and green) for unaccessible path on total the disconnected set.

Hence, when finding the distance between two points, we would also want to check if the path is inside the set. It is necessary for a path to be valid so that the distance is valid. ⌟

Therefore, we may induce the following metric based on the ℓ^2 norm.

Definition II.2.5. Optimized ℓ^2 Metric.

Let $X \subset \mathbb{R}^n$ be a subset of a finite dimensional Euclidean space, and let $x_1, x_2 \in X$ be arbitrary. Consider

$C : [0,1] \to X$ as any path between x_1 and x_2 such that $C(0) = x_1$ and $C(1) = x_2$, we let the distance be:

$$d(x_1, x_2) := \inf_{C \text{ is a path in } X \text{ between } x_1 \text{ and } x_2} \text{len}(C).$$

⌟

Here, we do want to prove that the above definition is, in fact, a metric based on Definition I.2.1.

Proof.

- **Positivity**: Since $\text{len}(C) \geq 0$, by the property of infimum, $d(x_1, x_2) \geq 0$.
- **Definiteness**: Suppose $d(x_1, x_2) = 0$, then there exists a path C between x_1 and x_2 whose length is 0, so x_1 and x_2 has to be the same point. Moreover $d(x_1, x_1) = 0$ since we can have the path $C(t) \equiv x_1$.
- **Symmetry**: Clearly if $C(t)$ is a path from x_1 to x_2, then $C(1-t)$ is a path from x_2 to x_1 and they have the same length.
- **Triangular inequality**: Suppose $x_1, x_2, x_3 \in X$ are arbitrary, let C_1 be a path from x_1 to x_2, C_2 is a path from x_2 to x_3, then we define C as:

$$C(t) := \begin{cases} C_1(2t), & \text{when } 0 \leq t \leq 1/2, \\ C_2\big(2(t - 1/2)\big), & \text{when } 1/2 < t \leq 1. \end{cases}$$

Clearly C is continuous by definition, $C(0) = x_1$ while $C(1) = x_3$, and $C([0,1]) \subset X$, so C is a legit path. Note that $\text{len}(C) = \text{len}(C_1) + \text{len}(C_2)$, so by definition of infimum, we have:

$$d(x_1, x_3) \leq d(x_1, x_2) + d(x_2, x_3).$$

Hence, we have shown that our definition of optimized ℓ^2 metric is, in fact, a metric. □

As we discussed as an assumption, we are using Euclidean distance as metric when finding the length of the path. Of course, it is possible to use other metric, but that makes the calculation of length of a curve more complicated. Moreover, this assumption is giving us some additional benefits.

Remark II.2.6. Piecewise Linear Curves are Optimal.

Since line segments are of the shortest length of all curves between two points, if the slit set is a closed line segment, the optimal path would be the line segments that cuts the endpoints of the slit segment, as of in Figure II.5. ⌟

Thereby, implementation-wise, we will be creating a metric over a subset in $\mathbb{R}^2$ that excludes the watershed in Baltimore city, as follows:

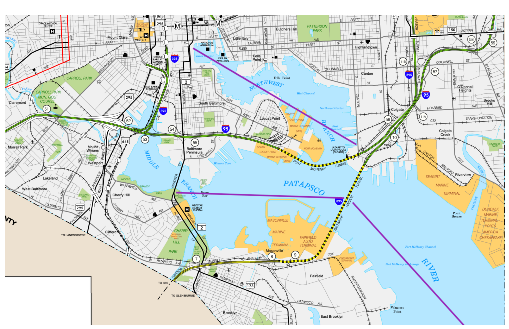

Figure II.6. Inaccessible paths (purple) on Baltimore map.

When a path attempts to intersect one of the purple line segments in the figure, it has to either path through the left or the right endpoints of the purple line segments.

II.3 Assigning Weights

According to our dataset, we first calculate the probability of occurrence of each type of crime in the variable `VIOLENT_CR`. Then we multiply all of the probability by 1000 and allocate numbers into each crime based on the **severity of crime**: homicide, rape, robbery, aggressive assault, common assault, burglary, larceny and auto theft.

Here, we assign a weight w_i on each point $\mathbf{x}_i \in X$ for $1 \le i \le n$, where w_i is the weight of the crime based on the **severity** above.

In implementation, we modify the K-means algorithms at Algorithm II.1.1. For equation ($**$), we modify it as computing the weighted average value of all points assigned to them, *i.e.*, for each $1 \le i \le k$:

$$\boldsymbol{\mu}_i = \frac{1}{\sum_{D_j=i} w_j} \sum_{D_j=i} w_j \mathbf{x}_j.$$

Therefore, for our **K-means** algorithm, it is more inclined to put the weight to the more sever and serious crimes, such as homicide. The model would *theoretically* be locating the police departments to locations that more serious crimes tend to happen.

II.4 Evaluation Methods

As we develop all the methods, one natural problem is on how we can evaluate the effectiveness of our model. Recall in I.1.1, we noted that we use the data from **October 1st, 2024** to **November 6, 2024** to train the model, and we use excerpts of data from **November 7, 2024** to **December 6, 2024** to validate our model.

Moreover, as a side note, we are utilizing Google map to find the time needed to go from a police department to a crime scene at 12AM and record the minimum possible time range.

II.4.1 Evaluating Different ℓ^p Metrics

For each ℓ^p model, we first train our model using the input data, weighted, and we then have the clustering model ready. Here, we use the validation procedure as follows.

Algorithm II.4.1. Evaluating Furtherest Possible Points in Each Cluster.

We do the following steps for each cluster:

1. Within each cluster, we *recluster* all points in it into 3 new clusters with the same distance metric but unweighted.

2. Within each new inner cluster, we calculate the distance between the all points in the new cluster with the original centroid.
3. Calculate the time needed to approach these three points from the original cluster using Google map.
4. Record the longest two time of the three time from the last step within each new clusters and average them.

As we conclude, we average the response time of clusters. ⌟

The reason we use this algorithm is that it will effectively find the furtherest distance at different directions of the centroid, so it could account for the different road setup of the city. The reason to remove the shortest time within each cluster is because we are doing clustering, the data could form a cluster that is close to centroid and that furtherest distance is not conclusive.

Remark II.4.2. Technical Difficulties.

Note that we developed this algorithm since it is hard to obtain a Google API for automatically calculating time. Moreover, given the constraint of manually finding time, we only recluster into 3 new clusters. If there had been more advanced technologies, it would be more effective to recluster into more smaller clusters or even computing the average travel time for all crime scenes within the cluster. ⌟

II.4.2 Comparing with Current Baltimore Police Setup

Recall that we are using data from **November 7, 2024** to **December 6, 2024** as validation set. The way that we compare our model with the Baltimore current setup (9 police departments) is to compare the response time between ours and theirs. [3]

From the over 3000 data points of the crime in the past month, we have randomly selected 35[3] crimes within the city. In particular, we will look up the average time needed to travel from our police department planning to the crime compared to the Baltimore current police departments. In particular, we would have two setups.

- First, we would use $k = 7$ to build 7 police departments, and we want to show that our model (with less location) can still perform no worse than the current setup.
- Also, we would use $k = 9$ to build 9 police departments, and we want to show that our model (with same number of location) will outperform the current setup.

The discussions of the results using such methods will be at III.2.

[3] Again, this number is limited by the manual search for travel time, this could be increased if we could obtain better data from Google map through APIs. See Remark II.4.2.

III Evaluations and Results

Now, we want to put together the observations throughout this project.

III.1 Errors in Data Set

Prior to our results, we want to note some errors/missing components that are in the data set [3]. Some sample of data with error are as below.

RowID	CCNumber	CrimeDateTime	CrimeCode	Description	Weapon	Post	Gender	Age	Race	Location	Latitude	Longitude
128782	23L09461	1523/09/08 17:30:02	7A	AUTO THEFT		421	F	0	WHITE	5100 HARFORD RD NORTHEAST	-76.565003	39.34795
128784	23C04992	1023/03/16 13:01:02	4B	AGG. ASSAULT	PER_WEA	822	F	30	BLACK	4600 PEN LUCY RD SOUTHWEST	39.285659	-76.693882
427984	18H02741	2018/8/7 16:30	6E	LARCENY	NA	822	M	-1	BLACK	3600 BROOKLYN AVE	39.23494022153	-76.59923441235
⋮ Other error data are omitted for presentation purposes. ⋮												

Table III.1. Selected errors (marked with red) from the crime data set on open Baltimore website [3].

III.1.1 Errors in Fields

A group of data was recorded using another date/time format. Among them is an `AUTO THEFT` that happened in 1523, while The first car was built after that, which indicates that there is a mistake in this recording date. In this case, this is probably due to an error when entering the data, and we simply omit these lines from the data (even if they could be a recent data).

Another type of error is recording wrong coordinates. Most of the cases in this dataset happened near $(39, -76)$ given by $(\text{latitude}, \text{longitude})$, as it is where Baltimore. However, there is a group of cases that happened around $(-76, 39)$, which is in Antarctica. Considering that the probability of a criminal escaping towards the far north is almost impossible, we can claim that this group of data is recorded by mistake, and we simply fix it by switching the coordinates.

There is an error in the recording age, too. Ages at -1, 0, and 1 are observed in the original dataset. We don't know the likelihood of a child at 0 or 1 year old using a knife or gun to hurt others, but of course, it is not possible for a person who was not born to rob others. We consider these cases recorded by mistake. However, since the age is not one of our research interest, we would still keep the data as it it had the correct age.

III.1.2 Missing Data

There are recordings without age, race, address, or other information covered in the original dataset. Considering some of the officers are not familiar with the recording system, or they are tired during the recording period, these could be careless mistakes. However, there could also be cases when such information is unknown, and it might not necessarily be random. However, since we are ignoring those fields, these loss are fine.

However, considering the crime location, time, and type as things that could not be unknown, the loss of those data would be MCAR, and we will just ignore the data without these information.

III.2 Evaluating Distance Metrics and Weights

After discussing the errors, we would use modified data to evaluate the effectiveness of our model.

III.2.1 Conclusion on ℓ^p Distances

First of all, we can notice that the centroids, in fact, are different in these different models.

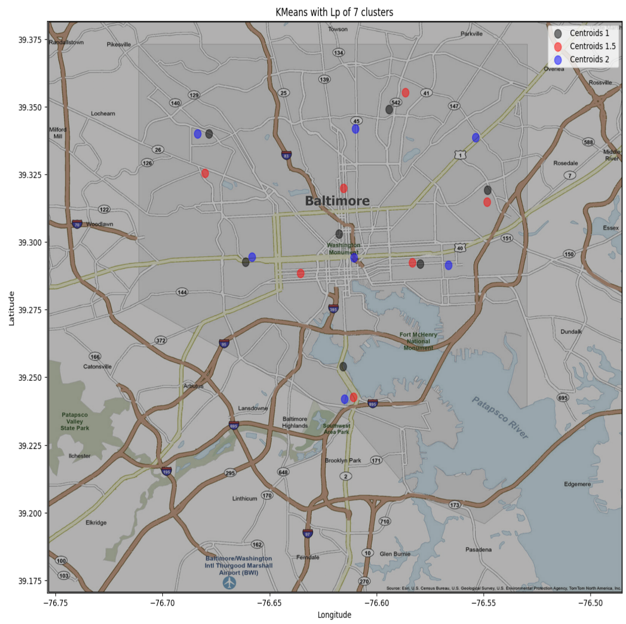

Figure III.2. Different locations of the police departments for different p on Baltimore map.

Also, we give a visualization of the furtherest point in the reclustering.

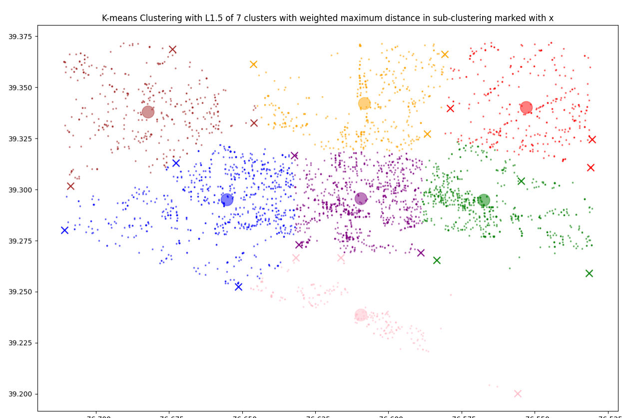

Figure III.3. Reclustering inside the current clustering for $p = 1.5$ with 7 clusters.

Then, we will be using Algorithm II.4.1 to compare the effectiveness between the different ℓ^p metrics for $p = 1, 1.25, 1.5, 1.75$, and 2 with $k = 7$. The results are as follows:

p	Minimum Travel Time (min)	
	Average	**Standard Derivation**
1	9.64	2.41
1.25	9.43	1.95
1.5	9.14	1.29
1.75	10.14	2.51
2	10.29	2.33

Table III.4. Minimum travel time from centroid to furtherest crime w.r.t model of different ℓ^p metrics.

From the above table, it is reasonable to use the **minimum travel time** since the police should be following the fastest path as possible.

Remark III.2.1. Potentially Faster does not Impact Trend.

In fact, the police cars should be able to get faster than the minimum travel time since they are allowed speeding and running a red light. However, since we are comparing between each metric with the (approximate) same rate of slowing down, we can ignore such effect. ⌟

Now, with the above statistics, we can conclude that **$p = 1.5$ is the best metric for all the ℓ^p metric that is tested**. Consequently, we will be using $p = 1.5$ to train the model in which we compare to the current Baltimore setup of police departments.

Remark III.2.2. Potential Deeper Development.

Potentially, we could develop an algorithm that aims to find a smaller average minimum travel time. For example, as of right now, the average is at minimum for the interval $[1.25, 1.5]$, so it could be possible for us to get the center of such interval and iteratively compare for the smaller average.
Note that we cannot really use gradient descent since the p function does not guarantee continuity. However, since the $n/2^k$ lattice is dense in the space (this can be proven in [7]), it is possible to see if we might obtain a p with shorter travel time. ⌟

III.2.2 Digression to Optimal ℓ^2 Metric

Recall we have defined a optimal ℓ^2 metric in Definition II.2.5, where we enforce the path to be within the not totally-connected set.

There, we have defined 3 inaccessible path in Figure II.6, and have trained the model with such metric.

Remark III.2.3. Computational Complexity for Optimal ℓ^2 Metric.

Different from the ordinary ℓ^p metric, the computation of distance is more computationally costly. To check the intersection of paths with inaccessible regions, we developed a $\mathcal{O}(N^2)$ check by simulating the paths, and moreover with the different distances between the points, the `numpy` operations of matrices no longer works. Therefore, the computational time explodes for the optimal ℓ^2 metric compared to the other

ℓ^p metrics. ⌟

However, it is possible to compare this metric with the Euclidean metric.

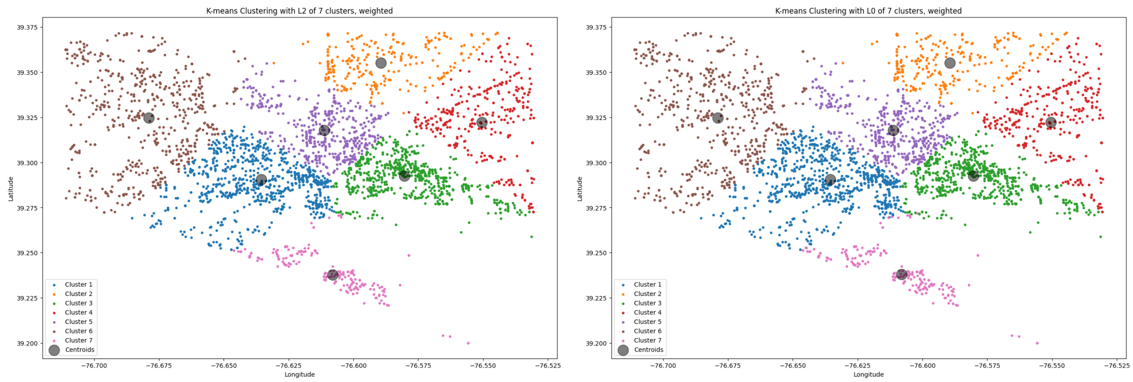

Figure III.5. Model trained with ℓ^2 metric (left) and optimal ℓ^2 metric (right).

For the $k = 7$, we noted that the centroids are almost the same, whereas there are some minor differences between the label of the points. We consider this reasonable for the following reasons:

- The clusterings were already separated by the sea, so it is reasonable that having the block of watershed is not changing the structure of the clusterings.
- The minor changes of the points are the points between the two inaccessible regions that are close to a bridge to the south region, so the optimized ℓ^2 metric is classifying them to the south department, which is further, but not blocked by water (since there is a bridge).

Therefore, we conclude the optimized ℓ^2 metric as a minor effective method, despite the computation intensity. It is noteworthy that this could also be modified for other distance metrics, but the challenge is to visualize the shortest path between points graphically.

III.2.3 Remarks on Weights

We started the idea of training based on weights. It is not possible to quantitatively analyze the model with weights. Given that there are less homicides compared to other crimes, the model with weights will work no better than the model without weights given randomized data points.

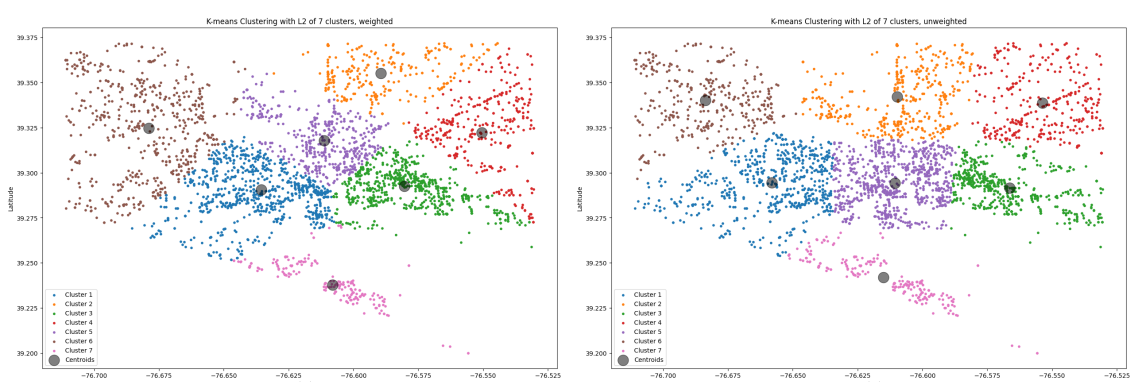

Figure III.6. Unweighted (left) and weighted models on Baltimore map.

It is noteworthy to point out that we added the weights so that the police could react faster to the more serious crimes. Also, given that murdering crimes are quite spread out in the city, giving them weights is not necessarily having a large impact on the model.

III.2.4 Comparison with Current Deployment

Then, we get to our major goal, to compare out model with the current deployment of police departments in Baltimore city. Here, the setup follows the setups in II.4.2.

Just as a reminder, we have:

- **Setup 1** as having 7 police department, which is less operation cost than current setup.
- **Setup 2** as having 9 police departments, which is approximately the same operation cost with the current setup.

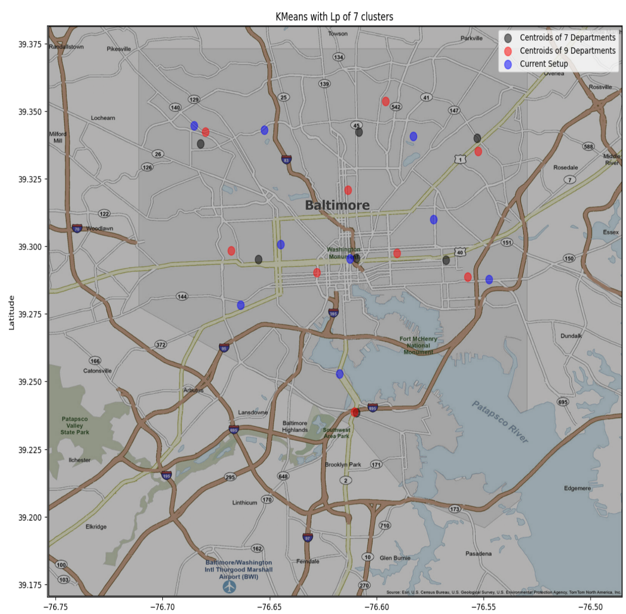

Figure III.7. Different locations of the police departments for the setups on Baltimore map.

The statistics of the models lie as follows.

Setup	Minimum Travel Time (min)	
	Average	Standard Derivation
Baltimore Current Setup	5.60	2.02
Setup 1	5.51	1.91
Setup 2	4.26	1.62

Table III.8. Minimum travel time from centroid to furtherest crime w.r.t model of different ℓ^p metrics.

Therefore, we can see that with our trained model, we can sustain (approximately) the same average travel time with the current setup with less number of police department, and we can reduce the average travel time with the current setup given the same number of police department.

Consequently, we conclude that our clustering model with $p = 1.5$ with added weights happens to be a better model compared to the current Baltimore police setup.

IV Conclusions

Now, we will be evaluating our model and suggest possible further develops given less constraints.

IV.1 Finding the Optimal Deployment

Our weighted k-means algorithm identifies $\ell^{1.5}$ norm as the most effective one in terms of police deployment across the city of Baltimore. This metric balances the trade-offs between the Manhattan and Euclidean distances, as well as the Patapsco River, fitting Baltimores urban geography and structure. [6] An optimal ℓ^2 norm is implemented in the algorithm to address the issue of police crossing the bay to be deployed, whose result is compared with ordinary ℓ^2 norm which omits the sea. Additionally, incorporating weights based on crime severity and frequency further optimized police station placements to address critical crimes like homicides. Our implementation has shown that with fewer resources (7 police departments), we can achieve response time comparable to the current setup, while the same level of resources (9 police departments) reduced response time by 81 seconds.

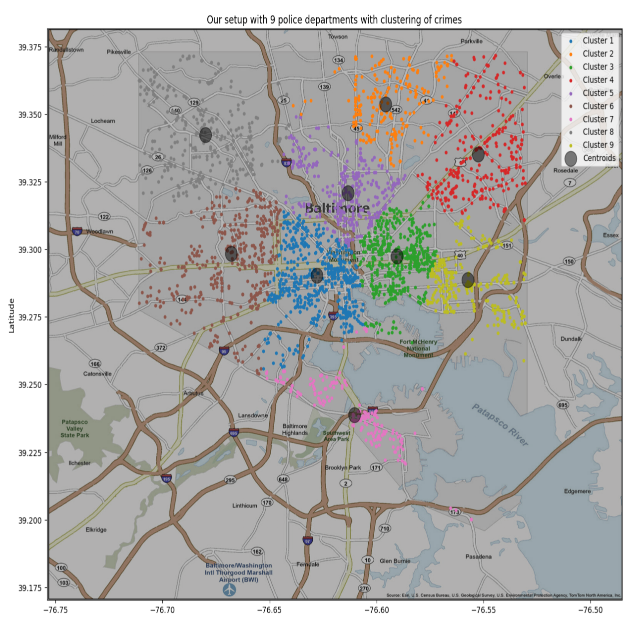

Figure IV.1. Map of the optimal model that we have developed.

IV.2 Suggestions and Further Directions

To address geographic challenges, such as water barriers, integrating path-specific metrics would further refine deployment strategies. These enhancements can help a more responsive and efficient police presence across the city.

Future work could focus on leveraging real-time crime trends to dynamically optimize police resource allocation and patrol routes rather than physically relocating stations. For instance, clustering models can inform where additional officers should be deployed temporarily or which areas require more frequent patrolling. Integrating tools like Google Maps APIs can enhance the granularity and scalability of evaluations. Also, predictive analytics could identify potential crime hotspots, allowing preemptive measures. More importantly, testing these methods in other cities with varying crime patterns and geographic layouts would further validate the model's generalizability, adaptivity, and effectiveness.

References

[1] V. Hill, "Baltimore named deadliest large city in the u.s., surpasses memphis and detroit," Fox News, August 2024. [Online]. Available: https://foxbaltimore.com/news/local/baltimore-named-deadliest-large-city-in-the-us-surpasses-memphis-and-detroit

[2] Campus security. Johns Hopkins University Public Safety. [Online]. Available: https://publicsafety.jhu.edu/community-safety/campus-security/

[3] (2023, July) Part 1 crime data. Open Baltimore. [Online]. Available: https://data.baltimorecity.gov/datasets/baltimore::part-1-crime-data/about

[4] S. Weisburd, "Police presence, rapid response rates, and crime prevention," *The Review of Economics and Statistics*, vol. 103, no. 2, pp. 280–293, 05 2021. [Online]. Available: https://doi.org/10.1162/rest_a_00889

[5] I. Shpitser, "Semi-supervised and unsupervised learning." [Online]. Available: https://courselore.org/files/25934429879842656545/lecture13.pdf

[6] A. K. Elmagarmid, *Similarity Search The Metric Space Approach*. Springer Science, 2006.

[7] R. S. Elias M. Stein, *Real Analysis: Measure Theory, Integration, and Hilbert Spaces*. Princeton University Press, November 2009.

[8] "General highway map: Baltimore city," Maryland Department of Transportation. [Online]. Available: https://www.roads.maryland.gov/Town_Gridmaps/100000_Baltimore%20City.pdf