

Crime Hotspots in Denver: Spectral Clustering and Discrete Barycenters

Natalia Villegas Franco

July 10, 2018

Advisor: Steffen Borgwardt
Department of Mathematical and Statistical Sciences



University of Colorado
Denver

Contents

1	Introduction	2
2	Discrete Wasserstein Barycenters	2
2.1	Linear programming model	3
2.1.1	Parameters	3
2.1.2	Variables	5
2.2	Experiments	6
2.3	Optimization using group of months	7
3	Fixed Transport	8
3.1	Optimization method	9
3.2	Experiments	9
4	Strongly Polynomial 2-Approximation	11
4.1	Optimization method	11
4.2	Experiments	12
5	Spectral Clustering	14
5.1	Theoretical framework	14
5.2	Applying spectral clustering	15
6	Conclusions	19

Abstract

Response times for police calls are one of the most important factors in regards to dealing with crime. These times could be reduced by placing the police in strategic locations. In this project, we apply spectral clustering to partition the locations of crime incidents in the Denver area into smaller parts. Then, we implement and compare three discrete barycenter models to determine an optimal police distribution in each part of the partition. A google maps html file displays the crime hotspots and suggested police placement for the whole region.

1 Introduction

Crime response planning via linear programming gives the optimal police location with regard to a set of crime locations from the Open Data Catalog [5]. The analysis in the project is focused on murder and robbery data but could be extended to other crimes.

The data is classified by months. The motivation to do this is that the number of policemen in Denver for a year is fixed even though some months have more crime than others. We need the police to be able to react to the crime in all the months. Therefore, the question is where to locate the police to respond faster to the crime. The goal is to minimize the expected distance between the crime location and the police location.

The Denver Open Data Catalog includes criminal offenses in the City and County of Denver for the previous five calendar years plus the current year to date. The data is based on the National Incident Based Reporting System (NIBRS). The data is filtered in the preprocessing stage of a Python program, avoiding crime locations outside of Denver area.

In the first part of the project, we implement and compare the Discrete Wasserstein Barycenters models using murder data. Then, we included robbery to the study. This increases the amount of input data for 2016 from 52 to 1204 cases. To be able to include different types of crime and work with a bigger data set, we apply spectral clustering to partition the locations of crime incidents in the Denver area into smaller parts. Then, we apply the discrete barycenter models to determine the optimal police location in each part.

The main result of the project is a Python code that extracts relevant data from Denver Open Data Catalogue, computes spectral clustering, implements discrete barycenters for crime data in AMPL, and provides a visualization of results using Google maps. Repository at <https://github.com/nataliavillegasfranco/barycenters>.

2 Discrete Wasserstein Barycenters

The weighted Wasserstein barycenters are optimal solutions to optimal mass transport problems for several marginals, with applications in economy [4, 13], signal processing [19] and statistics [6]. In this project, we applied Wasserstein Barycenters to analyze crime data.

The Discrete Wasserstein Barycenters model [7] will be called exact model when is compared with other models in following sections and is defined in the next subsection.

2.1 Linear programming model

We desire to solve the following linear programming model:

$$\min \sum_{i=1}^N \lambda_i \sum_{j=1}^{|S_0|} \sum_{k=1}^{|P_i|} \|x_j - x_{ik}\|^2 y_{ijk}$$

Subject to

$$\sum_{k=1}^{|P_i|} y_{ijk} = z_j \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, |S_0|,$$

$$\sum_{j=1}^{|S_0|} y_{ijk} = d_{ik}, \quad \forall i = 1, \dots, N, \quad \forall k = 1, \dots, |P_i|,$$

$$y_{ijk} \geq 0, \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, |S_0|, \quad \forall k = 1, \dots, |P_i|,$$

$$z_j \geq 0, \quad \forall j = 1, \dots, |S_0|.$$

Where the objective function minimizes the Euclidean distance between the crime and possible police locations. The two first constraints balance the supply and demand of police in the possible police locations. The two last constraints ensure all weights are positive.

In the next subsections, we will explain the parameters and variables of the model and how we apply it to solve this problem using data from three months.

2.1.1 Parameters

The crime location k in the month i is represented by x_{ik} . The location can be represented by latitude-longitude coordinates as in Figure 1. The three different marker colors, black, green and brown, represent the different months January, February and March ($N = 3$) and each month i has a particular number of crimes which are represented by $|P_i|$ ($|P_1| = 3$, $|P_2| = 2$ and $|P_3| = 3$).

The parameter λ_i represents the weight for each month i according to its relevance. For instance, if it is known that in March Denver is more dangerous than in the other months, it is possible to define a larger weight for that particular month. The restrictions for λ are

$$\lambda \geq 0, \quad \sum_{i=1}^N \lambda_i = 1.$$

In all the experiments below, λ_i is defined to have the same weight for each month:

$$\lambda_i = \frac{|P_i|}{\sum_{i=1}^N |P_i|}.$$

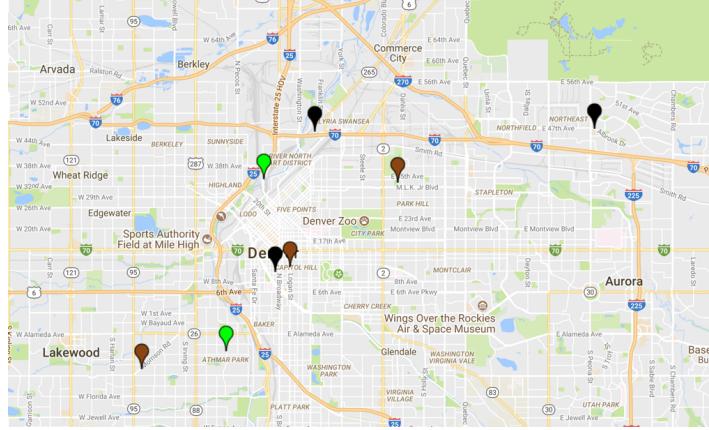


Figure 1: Example of crime location data for Denver from January to March of 2016.

So, in this example:

$$\begin{aligned}\lambda_1 &= \frac{|P_1|}{\sum_{i=1}^N |P_i|} = \frac{3}{8} = 0.375 \\ \lambda_2 &= \frac{|P_2|}{\sum_{i=1}^N |P_i|} = \frac{2}{8} = 0.25 \\ \lambda_3 &= \frac{|P_3|}{\sum_{i=1}^N |P_i|} = \frac{3}{8} = 0.375.\end{aligned}$$

Next, the parameter d_{ik} represents the crime weight, and it is subject to:

$$\sum_{k=1}^{|P_i|} d_{ik} = 1, \quad \forall i = 1, \dots, N.$$

In all the experiments below, the same weight is used for all the crimes in the same month:

$$d_{ik} = \frac{1}{|P_i|}, \quad \forall k = 1, \dots, |P_i|.$$

So, in our three months example:

$$\begin{aligned}d_{1k} &= \frac{1}{|P_1|} = \frac{1}{3}, \quad \forall k = 1, \dots, 3 \\ d_{2k} &= \frac{1}{|P_2|} = \frac{1}{2}, \quad \forall k = 1, \dots, 2 \\ d_{3k} &= \frac{1}{|P_3|} = \frac{1}{3}, \quad \forall k = 1, \dots, 3.\end{aligned}$$

The last parameter is the possible locations of the police, represented by x_j , which is computed using the crime location data x_{ik} . It is defined as the mean of all the individual combinations of the crime locations of each month. Therefore, the number of possible police locations $|S_0|$ is:

$$|S_0| = \prod_{i=1}^N |P_i|.$$

In the previous example:

$$|S_0| = \prod_{i=1}^3 |P_i| = |P_1| \cdot |P_2| \cdot |P_3| = 3 \cdot 2 \cdot 3 = 18.$$

In Figure 2, there are 18 blue markers representing the possible police locations.

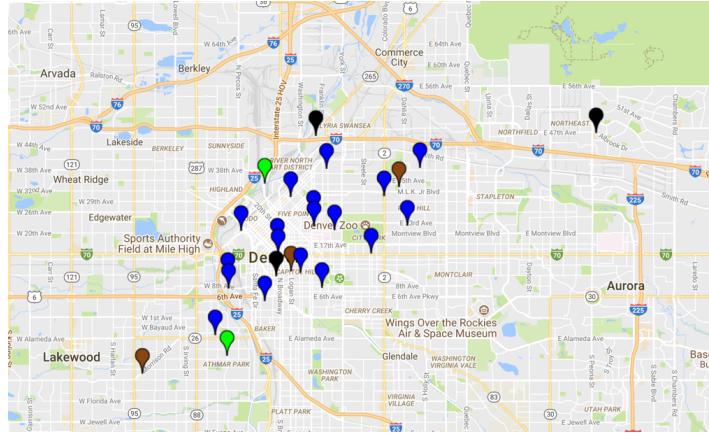


Figure 2: In blue, the possible locations x_j of the police using crime data of Denver from January to March of 2016.

2.1.2 Variables

The linear programming model is controlled by the transport variable (y_{ijk}) which balances the weights between crimes and police.

The variables z_j represent the police weights, which control the supply part of the model. Each weight represents the relevance of each police location. z_j is subject to:

$$\sum_{j=1}^{|S_0|} z_j = 1.$$

At the end, we take all the police location x_j satisfying:

$$z_j > 0.$$

So, in our example, there are 4 police locations with positive weight as seen in Figure 3, and the other 14 are removed because they have weight 0.

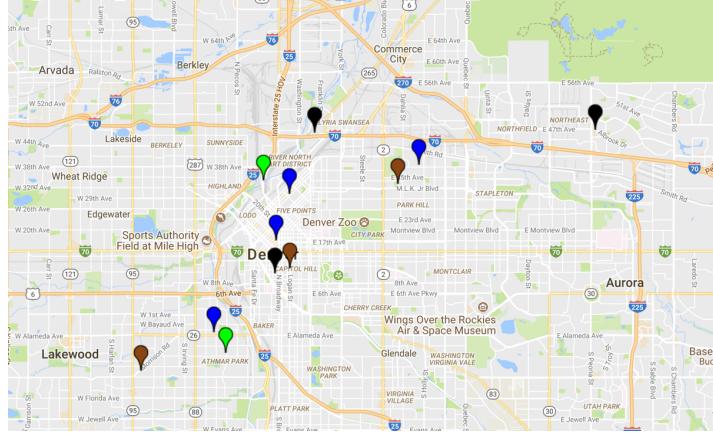


Figure 3: Final location of the police (in blue all x_j with $z_j > 0$) using crime data of Denver from January to March of 2016.

2.2 Experiments

The execution time using the exact model for three months is 0.12 seconds (Figure 3). The value of the objective function is 0.00182. We are going to use the value of the objective function to compare different models. As we can see in Figure 3, each barycenter (suggested police location) is the center of mass of a set of crimes

The execution time using the exact model for 9 months is 59 seconds (Figure 4). The value of the objective function is 0.00377.

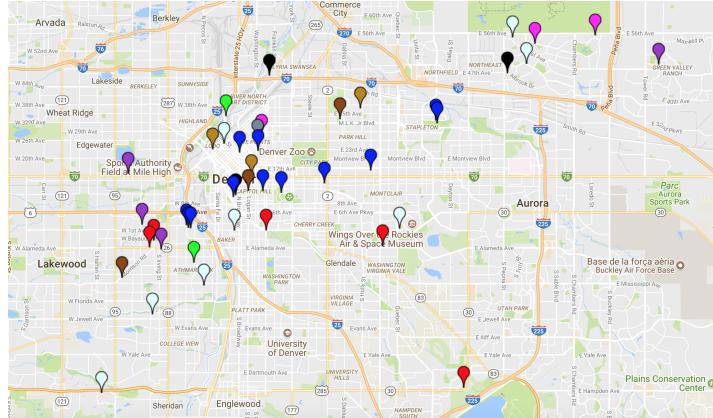


Figure 4: 9 month experiment.

The execution time using the exact model for 12 months is 66 minutes (Figure 5). The computer ran out of memory before AMPL can give an optimal result.

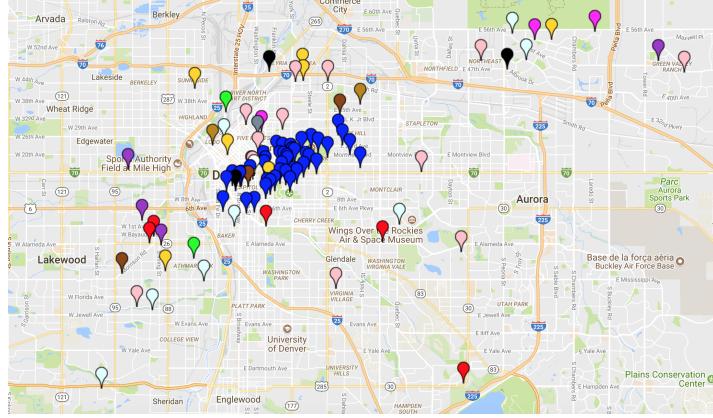


Figure 5: 12 month experiment.

2.3 Optimization using group of months

Using groups of months, it is possible to run the exact model for a year. The execution time using the exact model for 12 months in groups of two months is 22 minutes. The value of the objective function is 0.00177.

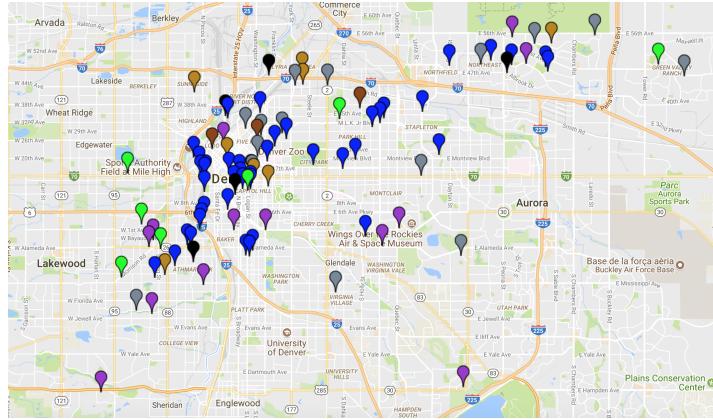


Figure 6: 12 month experiment using 2-month groups (6 groups).

The execution time using the exact model for 12 months in groups of three months is 70 seconds. The value of the objective function is 0.00107. In this case, there are a lot of police markers. This happens because we are increasing the number of murders per group.

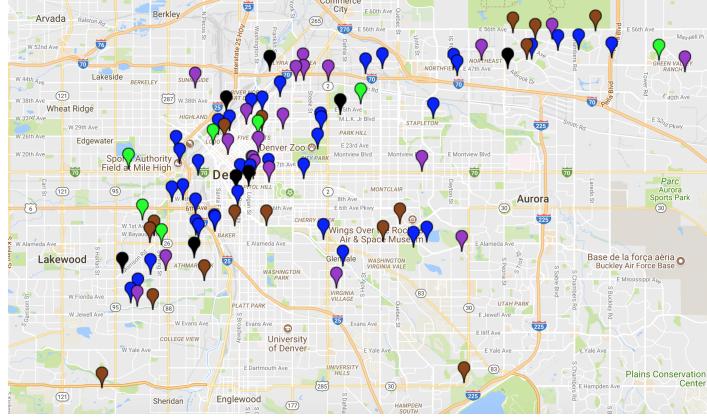


Figure 7: 12 month experiment using 3-month groups (4 groups).

To address this, we define an $\epsilon > 0$. It will represent the tolerance level for choosing police location. We only choose the location x_j if:

$$z_j > \epsilon.$$

So, going back to the 12 month experiment using 3-month groups, we can observe in Figure 8 the result using $\epsilon = 0.03$, showing that it is possible to see a significant reduction of the police markers.

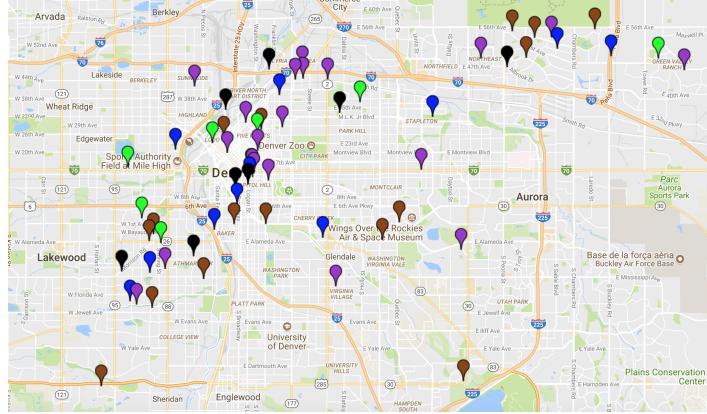


Figure 8: 12 month experiment using 3-month groups (4 groups) and $\epsilon = 0.03$

3 Fixed Transport

Steffen Borgwardt and Stephan Patterson have been working on an alternative linear program for the Discrete Barycenter Problem where the objective is to assign mass to z_j and get a fixed transport to the measures [3]. We applied this model to our data and saw a decrease in computational time. We were able to use this approach because for each month i , we rely only on one x_{ik} to generate each x_j . In other words, police officers only respond to murders in their own jurisdictions. We discuss this in greater detail in Sections 3.1 and 3.2.

3.1 Optimization method

Using a fixed transport we obtain the following LP as an alternative to solving the Discrete Barycenter Problem:

$$\min \sum_{h=1}^{|S^*|} c_h w_h$$

Subject to

$$\sum_{h \in S_{ik}^*} w_h = d_{ik}, \quad \forall i = 1, \dots, N, \quad \forall k = 1, \dots, |P_i|,$$

$$w_h \geq 0, \quad \forall h = 1, \dots, |S^*|.$$

Where S^* and S_{ik}^* are defined:

$$S^* = \{(x_{1k_1}, \dots, x_{nk_n}) : x_{ik_i} \in \text{supp}(P_i)\} := \{s_1^*, \dots, s_{|S^*|}^*\},$$

$$S_{ik}^* = \{h : s_h^* = (\dots, x_{ik}, \dots)\}.$$

The indexes k_i in a specific tuple s_h^* can take values from $1, \dots, |P_i|$ and they are the indexes defining the murder in the month i , x_{ik_i} . Now, we can define the indexes k_i depending on which tuple h we are considering. So, we are going to define the index k_{hi} , the murder index in the tuple s_h^* and month i . Using this notation, we can define each tuple $s_h^* = (x_{1k_{h1}}, \dots, x_{nk_{hn}})$, $h = 1, \dots, |S^*|$, and corresponds to a set of crime data x_{ik_i} , $i = 1, \dots, n$. We need also to define x^h as the possible police location generated by the tuple h . Furthermore, we introduce a variable w_h for each $s_h^* = (x_{1k_{h1}}, \dots, x_{nk_{hn}})$, $h = 1, \dots, |S^*|$, representing mass associated with it. Finally, the corresponding cost c_h of transporting a unit of mass to the measures is

$$c_h = \sum_{i=1}^n \lambda_i \|x^h - x_{ik_{hi}}\|^2.$$

3.2 Experiments

The execution time using fixed transport model for 3 months (Figure 9) is 0.23 seconds. The value of the objective function is 0.00182.

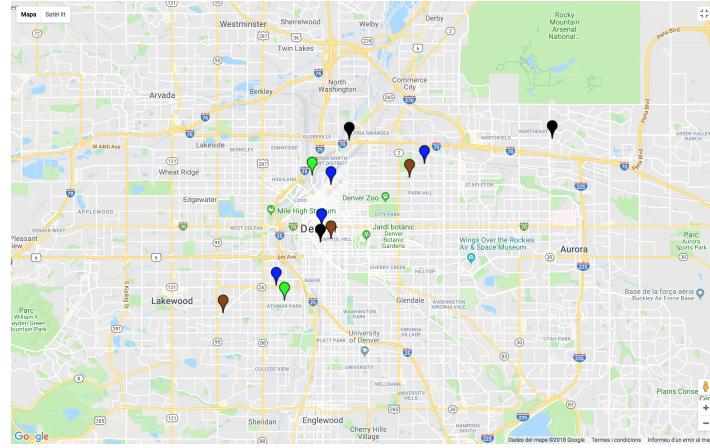


Figure 9: 3 month experiment.

The execution time using fixed transport model for 9 months (Figure 10) is 17 seconds. The value of the objective function is 0.00383.

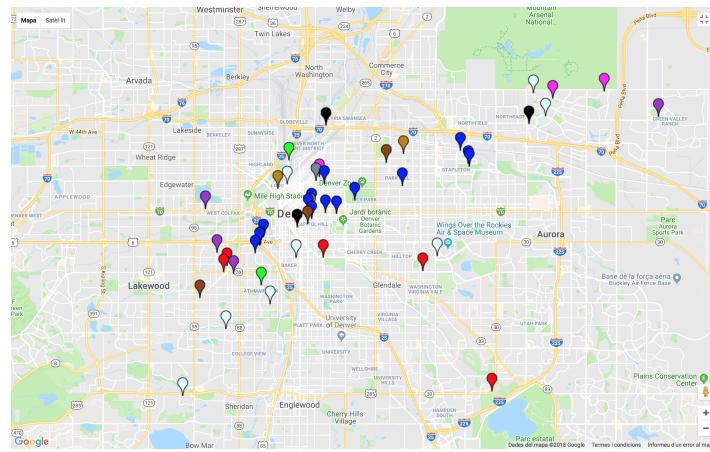


Figure 10: 9 month experiment.

Finally, the execution time using fixed transport model for 12 months (Figure 11) is 2460 seconds. The value of the objective function is 0.0032936.

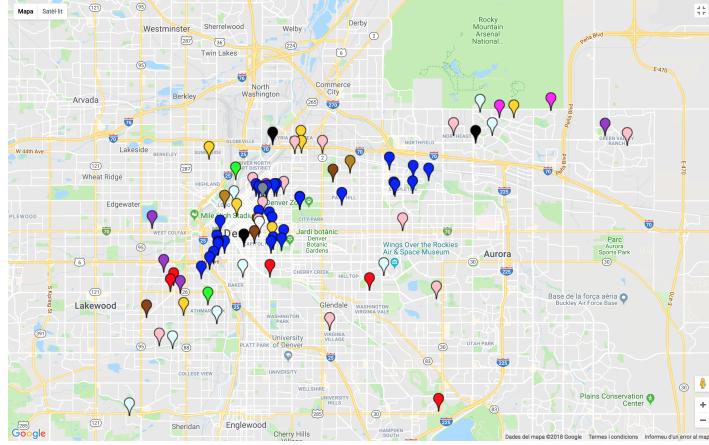


Figure 11: 12 month experiment.

4 Strongly Polynomial 2-Approximation

The Strongly Polynomial 2-Approximation algorithm [2] trades a small error for a significant reduction in computational effort. The Strongly Polynomial 2-Approximation will be called 2-approximation in following sections and it is defined in the next subsection.

4.1 Optimization method

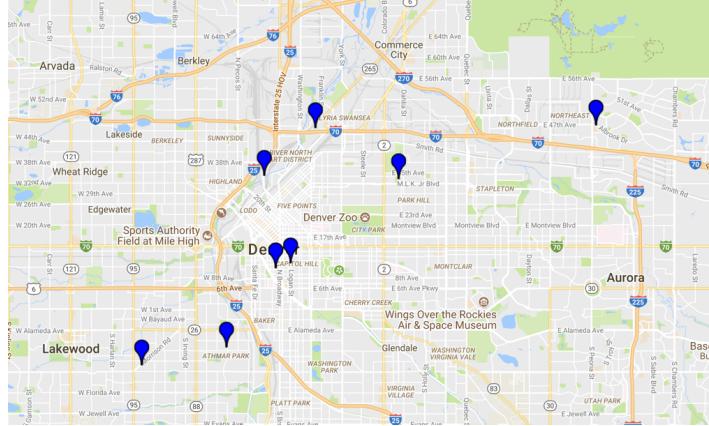


Figure 12: Possible locations of the police x_j using the 2-approximation method.

The main difference between the exact model and the 2-approximation method is computing the x_j parameter (possible police location). In the exact model, the possible police location is the mean of all combinations of crime locations of each month. In the 2-approximation, the possible police location is the exact location of the crimes, it is computed in the original supports.

Figure 12 shows the possible police locations, we can see a significant reduction in the number of x'_j s. This implies a significant reduction of the execution time compared

to the exact model. Figure 13 shows the final police location after applying the mean of all the crime locations x_{ik} involved in the transportation of each x_j mass.

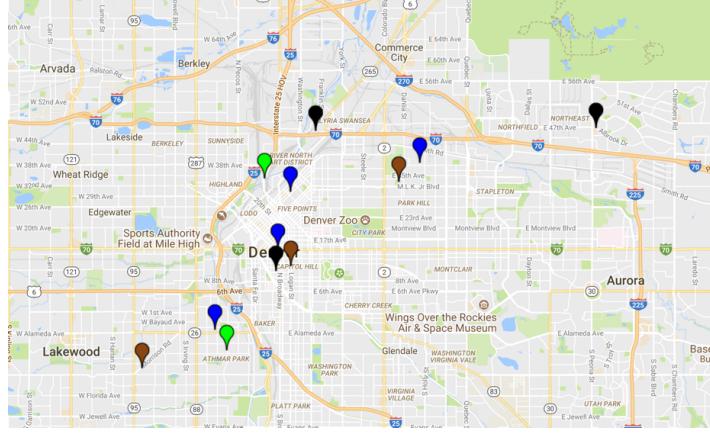


Figure 13: Result of the police location (in blue) using Strongly Polynomial 2-Approximation.

4.2 Experiments

The execution time using 2-aproximation for 3 months (Figure 14) is 0.11 seconds. The value of the objective function is 0.0018227, compared with the objective function in the exact model, the percentage error is 0.2%.

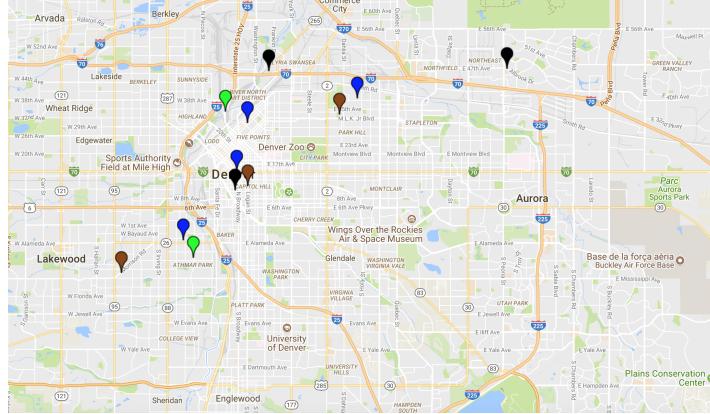


Figure 14: 3 month experiment

Similarly, the execution time using 2-aproximation for 9 months (Figure 15) is 0.17 seconds. The value of the objective function is 0.00379, compared with the objective function in the exact model, the percentage error is 0.6%. But the computational time decrease from 59 seconds to 0.17 seconds.

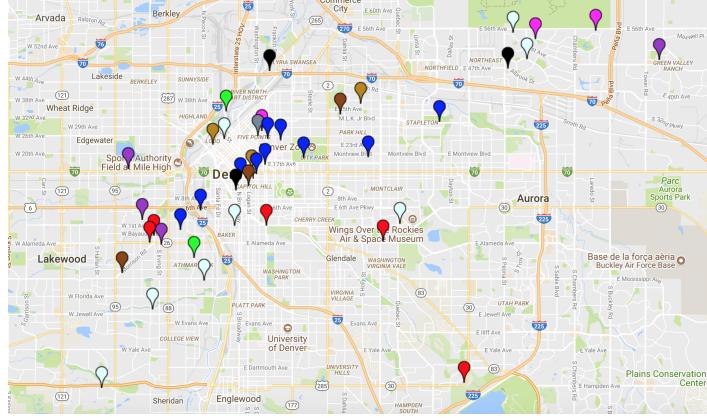


Figure 15: 9 month experiment

The execution time using 2-aproximation for 12 months (Figure 16) is 0.29 seconds. The value of the objective function is 0.00334. We cannot compared it with the exact model because the computer ran out of memory before obtaining an optimal result. There are fewer police markers than the exact model with groups for the same number of murders.

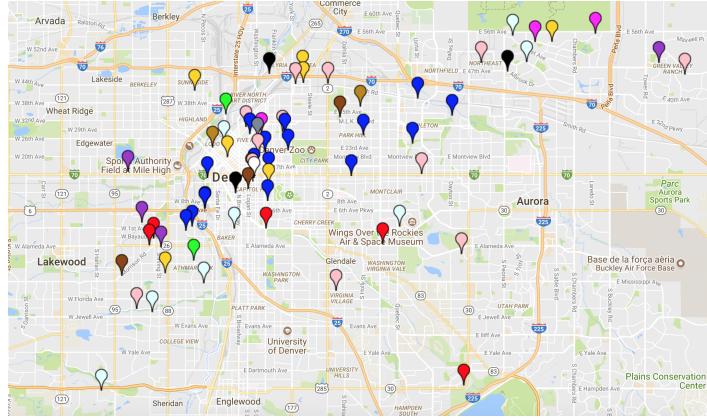


Figure 16: 12 month experiment

Finally, the execution time using 2-aproximation for 67 months (Figure 17) is 9 seconds. The value of the objective function is 0.00352. Those are all the months available (from February 2012 to August 2017) with a total number of murders of 263. The model sends all the police in the middle. This makes sense because it should send police to all the months.

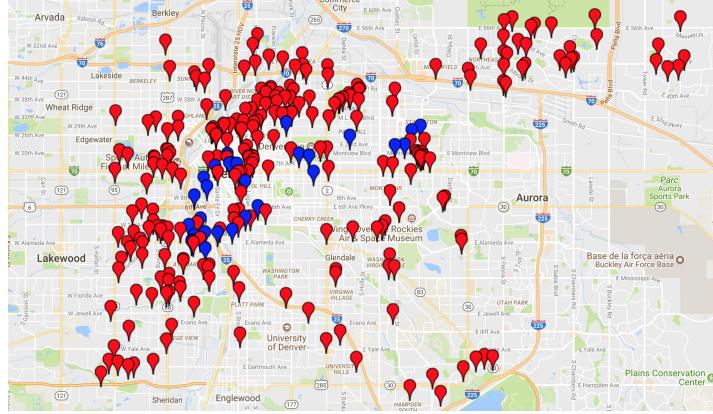


Figure 17: 67 month experiment

5 Spectral Clustering

Now, we would like to include in the study crimes like robbery. This increases the amount of input data for the year of 2016 from 52 murders to 1204 murders and robberies. To be able to work with a bigger data set, we apply spectral clustering to partition the locations of crime incidents in the Denver area into smaller parts.

5.1 Theoretical framework

Spectral clustering is a method that uses weighted graphs to partition the network into groups of different sizes and densities. We provide a brief overview, following [8, 11]. Let $G = (V, E)$ be an undirected graph with vertex set $V = \{v_1, \dots, v_n\}$ and edge at $E = \{(v_i, v_j) : v_i, v_j \in V\}$. For each v_i we define the degree of $v_i \in V$ as

$$d_i = \sum_{j=1}^n w_{ij},$$

where w_{ij} is a weight between two vertices v_i and v_j ; note that $w_{ij} \geq 0$ and because the graph is undirected $w_{ij} = w_{ji}$.

Different similarity graphs

For a set of data points x_1, \dots, x_n , with $x_i \in \mathbb{R}^2$ let the similarity graph be represented by $G = (V, E)$, where each vertex v_i in V represents a data point x_i . Given $s_{ij} \geq 0$ i.e., some notion of similarity between all pairs of data points x_i and x_j , two vertices are connected if the similarity value is positive or larger than a certain threshold; the edge is then weighted by s_{ij} . There are several constructions to transform a given set x_1, \dots, x_n of data points with pairwise similarities s_{ij} or pairwise distances d_{ij} into a graph $G = (V, E)$ (see [11] for details).

The unnormalized graph Laplacian

Let $W = (w_{ij})_{i,j=1,\dots,n}$ the weighted adjacency matrix of the graph and the degree matrix \mathcal{D} the diagonal matrix with the degrees d_1, \dots, d_n on the diagonal. The unnormalized graph Laplacian matrix is defined as

$$L = \mathcal{D} - W.$$

Unnormalized spectral clustering algorithm

To be able to formalize spectral clustering, we need to define the similarity matrix. Given a set of data points x_1, \dots, x_n and some notion of similarity $s_{ij} = s(x_i, x_j)$, there is a similarity matrix $S = (s_{ij})_{i,j=1,\dots,n}$ which is assumed to be symmetric and non-negative.

Algorithm 1 Unnormalized spectral clustering algorithm

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$ and number m of clusters to construct.

- Construct a similarity graph. Let W be its weighted adjacency matrix.
- Compute the first m eigenvectors u_1, \dots, u_m of L .
- Let $U \in \mathbb{R}^{n \times m}$ be the matrix containing the vectors u_1, \dots, u_m as columns.
- Let $y_i \in \mathbb{R}^m$ be the vector corresponding to the i -th row of U , for $i = 1, \dots, n$
- Cluster the points $(y_i)_{i=1,\dots,n}$ in \mathbb{R}^m into clusters C_1, \dots, C_m .

Output: Clusters A_1, \dots, A_m with $A_i = \{j | y_j \in C_i\}$.

The output of the algorithm returns the partitioned network. Points in different clusters are dissimilar to each other; that is, between-cluster similarities are minimized and within-cluster similarities are maximized.

5.2 Applying spectral clustering

During the last years, spectral clustering has been used across multiple fields, including ultrasound image segmentation [1], financial modeling and forecasting [10], protein detection [15], neuronal assembly identification [12], bioinformatics [9], recognition of semiconductor defect patterns [17], delineation of tumors [18], and keyframe detection (large-scale mapping and scene recognition) [16]. Now, we want to apply it to partition crime data in Denver.

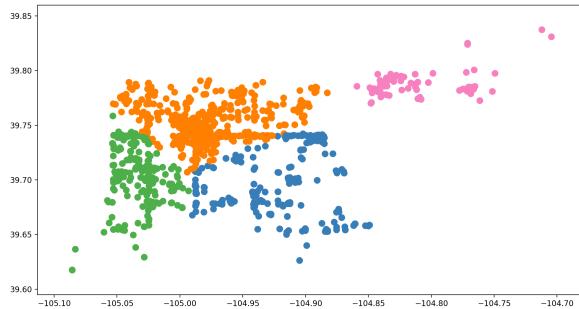


Figure 18: Partitioned network for murder and robbery data using spectral clustering with 4 clusters.

To apply spectral clustering, we consider the k -nearest neighbor approach [11]. Spectral clustering generates a network that captures the relationships between crimes based on their location, using a similarity matrix $S = (s_{ij})_{i,j=1,\dots,n}$, where $s_{ij} = s(x_i, x_j)$. For the k -nearest neighborhood approach, we use the similarity function

$$s_{x_i, x_j} = (\max\{d_{u,v}^*; \forall u, v = 1, \dots, n\} - \min\{d_{u,v}^*; \forall u, v = 1, \dots, n\}) - d_{i,j}^*.$$

where $d_{i,j}^*$ is the Euclidean distance $d_{i,j}^* = \|x_i - x_j\|$.

The input parameter for this type of graph is the value of k , where the crime i is connected to the crime j if x_j is among the k -nearest neighbors of x_i .

We apply spectral clustering [14] to robbery and murder data from the year 2016 and get partition in Figure 18.

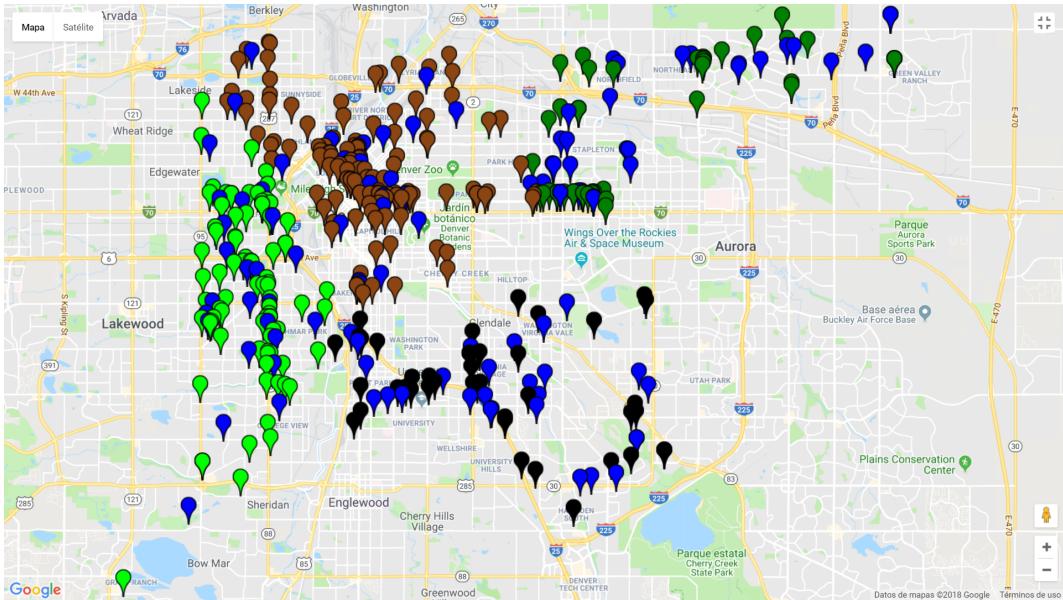


Figure 19: Fixed transport with 4 clusters. Murder and robbery data for 2016 (3 months). Blue markers represent the suggested police location

Figure 19 shows the fixed transport model with 4 clusters with murder and robbery data for January to March 2016. The total execution time is 85 seconds. When we applied the fixed transport model for murder and robbery data for 4 months, just the text file size with the parameters for AMPL was 13 gigabytes. The total execution time was 6684 seconds, almost 2 hours. The time creating the x_j increases exponentially. Therefore, we could not continue increasing the number of months with the fixed transport model.

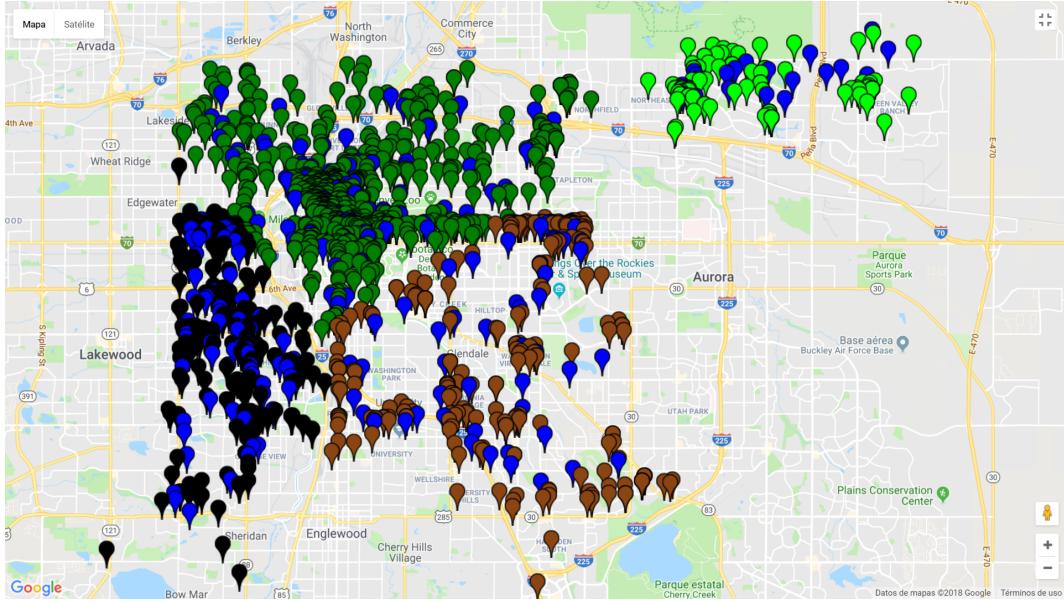


Figure 20: Strongly Polynomial 2-Approximation with 4 clusters. Murder and robbery data for 2016 (12 months).

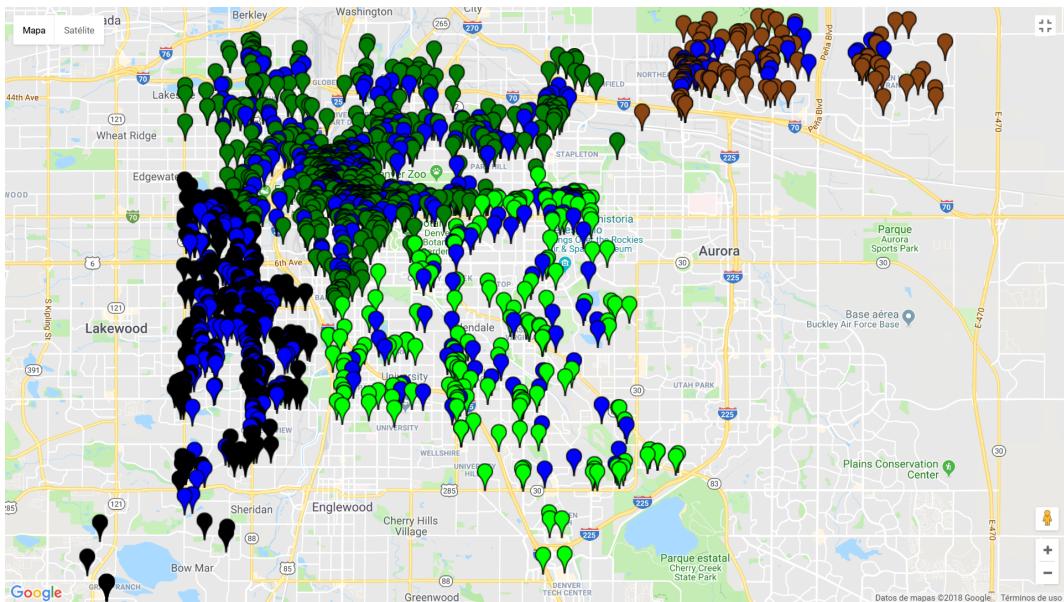


Figure 21: Murder and robbery locations in Denver in 2015 (4 clusters) and suggested police presence locations (blue markers).

Figure 20 and 21 shows the 2-approximation model with 4 clusters with murder and robbery data for January to December 2016 and 2015 respectively. In order to validate the quality of this results, we used the suggested police location obtain with the model for 2015 and predicted the best police location for the year 2016 (Figure 22). In other words, we compared the distance between the crimes in 2016 and the suggested police location obtain from 2015. The difference in the objective functions was less than 7%,

showing that with this model we can predict the police locations that will respond faster to the crimes.

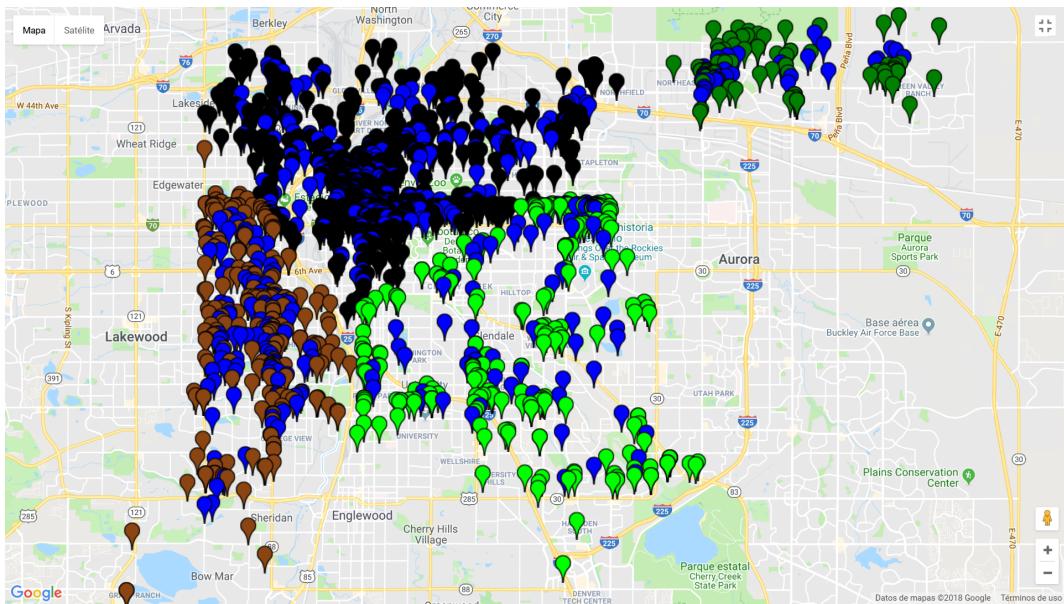


Figure 22: Crimes in 2016 and the suggested police location obtain from 2015.

6 Conclusions

In this project, we applied three different methods to optimize the positioning of police with regards to a set of crime locations in the city of Denver. Figure 23 shows the murder locations in Denver and suggested police presence locations. The radii of the shapes in the figure are relative to their masses. Their masses represent the percent of police resource that should be sent to each location, for example, a police weight of 0.02 means that the 2% of the police resource should be sent to that location. Recall that adding all the police weights for the year is equal to one.

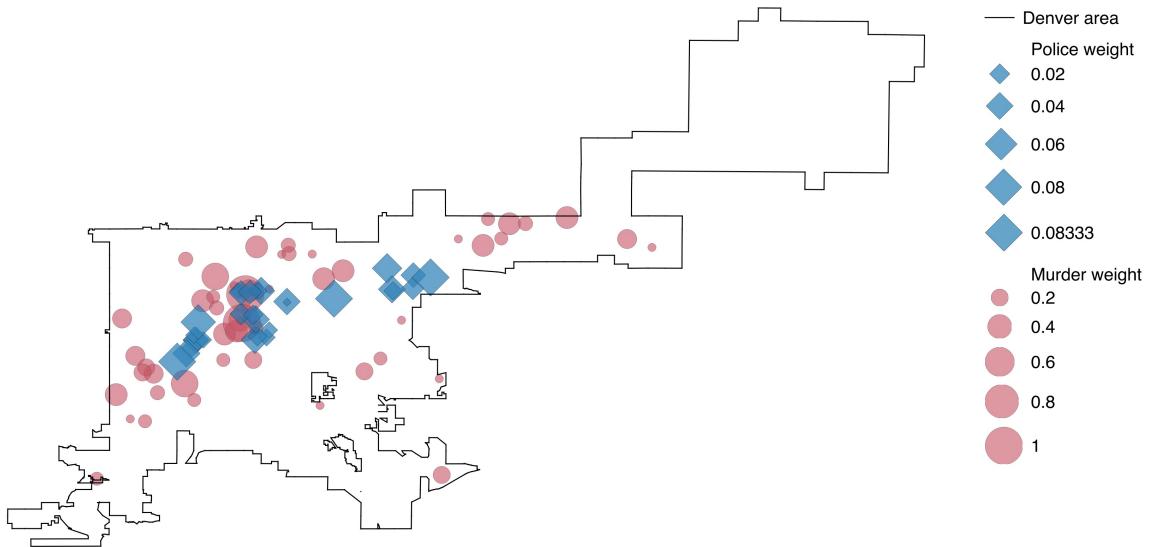


Figure 23: Murder locations in Denver in 2016 and suggested police presence locations

In order to take into account all the possible crimes, we need to deal with a big data set. Therefore, we applied spectral clustering to partition the city and study each part of the partition independently. Table 1 compares the three models using murder data in Denver for the year 2016.

In the table, we can observe that in the exact and fixed transport models, the execution time of the model scales exponentially with the sample size, whereas in the 2-approximation, the execution time scales linearly. Additionally, during the project, the greatest percentage error of the 2-approximation compared to the exact model was 0.5%. Therefore, when analyzing a large amount of data, it is recommended to use the 2-approximation model because it best optimizes the tradeoff between accuracy and

execution time.

Table 1: Comparison between models.

Method	Number of months	Objective Function	Execution time (s)
Exact model	3 months	0.0018198	0.12
Exact model	9 months	0.0037722	59
Exact model	12 months	None	3960
2-months groups	12 months / 6 groups	0.0017717	1320
3-months groups	12 months / 4 groups	0.0010652	70
Fixed transport	3 months	0.0018227	0.23
Fixed transport	9 months	0.0038293	17
Fixed transport	12 months	0.0032936	2460
2-approximation	3 months	0.0018227	0.11
2-approximation	9 months	0.0037932	0.17
2-approximation	12 months	0.0033421	0.29
2-approximation	67 months	0.0035176	9

References

- [1] Neculai Archip, Robert Rohling, Peter Cooperberg, and Hamid Tahmasebpour. Ultrasound image segmentation using spectral clustering. *Ultrasound in Medicine and Biology*, 31(11):1485–1497, 2005.
- [2] Steffen Borgwardt. Strongly polynomial 2-approximation of Discrete Wasserstein barycenters. *eprint arXiv:1704.05491*, 2017.
- [3] Steffen Borgwardt and Stephan Patterson. Improved linear programs for discrete barycenters. *eprint arXiv:1803.11313*, 2018.
- [4] G. Carlier and I. Ekeland. Matching for teams. *Economic Theory*, (2)(42):397–418, 2010.
- [5] City and Denver Police Department / Data Analysis Unit County of Denver. Denver open data catalog. <https://www.denvergov.org/opendata/dataset/city-and-county-of-denver-crime>.
- [6] C. Matran E. del Barrio, J.A. Cuesta-Albertos and A. Mayo-Iscar. Robust clustering tools based on optimal transportation. *Statistics and Computing*, 2018. <https://doi.org/10.1007/s11222-018-9800-z>.
- [7] Steffen Borgwardt Ethan Anderes and Jacob Miller. Discrete wasserstein barycenters: Optimal transport for discrete data. *Mathematical Methods of Operations Research*, 84 (2), pages 389–409, 2016.
- [8] Natalia Villegas Franco. On the perception of corruption: A spectral clustering analysis. Undergrad Thesis, 2013.
- [9] Desmond J. Higham, Gabriel Kalna, and Milla Kibble. Spectral clustering and its use in bioinformatics. *Journal of Computational and Applied Mathematics*, 204(1):25–37, 2007.
- [10] Shian-Chang Huang. Integrating spectral clustering with wavelet based kernel partial least square regressions for financial modeling and forecasting. *Applied Mathematics and Computation*, 217(15):6755–6764, 2011.
- [11] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.
- [12] Karim Oweiss, Rong Jin, and Yasir Suhail. Identifying neuronal assemblies with local and global connectivity with scale space spectral clustering. *Neurocomputing*, 70(10-12):1728–1734, 2007.
- [13] R. McCann P-A. Chiaporri and L. Nesheim. Hedonic price equilibria, stable matching and optimal transport; equivalence, topology and uniqueness. *Economic Theory*, (2)(42):317–354, 2010.

- [14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [15] Guimin Qin and Lin Gao. Spectral clustering for detecting protein complexes in protein-protein interaction (PPI) networks. *Mathematical and Computer Modelling*, 52(11-12):2066–2074, 2010.
- [16] Ricardo Vázquez-Martín and Antonio Bandera. Spatio-temporal feature-based keyframe detection from video shots using spectral clustering. *Pattern Recognition Letters*, 34(7):770–779, 2013.
- [17] Chih-Hsuan Wang. Recognition of semiconductor defect patterns using spatial filtering and spectral clustering. *Expert Systems with Applications*, 34(3):1914–1923, 2008.
- [18] Fei Yang and Perry W. Grigsby. Delineation of FDG-PET tumors from heterogeneous background using spectral clustering. *European Journal of Radiology*, 81(11):3535–3541, 2012.
- [19] Jianbo Ye, Panruo Wu, James Z. Wang, and Jia Li. Fast discrete distribution clustering using wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65(9):2317–2332, 5 2017.