Popular Routes Discovery

Tal Ben Yakar
Tel Aviv
Tal-by@csail.mit.edu

Abstract

Finding the optimal driving route has attracted considerable attention in recent years, the problem sounds simple however different companies these days, taxi alternatives companies like Uber and Via trying to find what is the best route to drive find it as a very challenging problem. Ridesharing and maps companies like HERE, navigation companies like waze and public transportation companies like moovit and others. AI robots in addition, need to have the ability to route in the optimal manner. In this work we formulate the problem of finding optimal routes as an optimization problem and come up with a neat, low memory and fast solution to the problem using machine learning algorithms.

Keywords: **AI**, machine learning, optimal **routes,** robots**,** autonomous vehicles, optimal route, clustering, K-Means, K-Medoids, K-Medians, partition.

1 Introduction

Many people tried to solve such an interesting problem. How does Google Maps determine optimal routes? `Richard et al [9] show that an optimal set of facility locations can be drawn from a finite set of candidate points, all of which are easy to determine. They considered the optimal location of p facilities in the plane, under the assumption that all travel occurs according to the Manhattan (or rectilinear or /i) metric in the presence of impenetrable barriers to travel. Facility users are distributed over a finite set of demand points, with the weight of each point proportional to its demand intensity. Each demand point is assigned to the closest facility. The objective is to locate facilities so as to minimize average Manhattan travel distance to a random demand. Mohamad et al [6] researched Approximation Algorithms for Metric Facility Location Problems. Meyerson and Brian Tagiku [10] found a way to minimize Average Shortest Path Distances via Shortcut Edge Addition

Similar problem apply robots AI and pathfinding algorithms. Some might use search algorithms to do so. In addition, autonomous (i.e., robotic, self-driving) vehicles are rapidly becoming a reality. Zhang et al [11] provide a rigorous approach to the problem of congestion-aware, system-wide coordination of autonomously driving vehicles, and to the characterization of the sustainability of such robotic systems.

2 Model description and problem formulation

A route is a pair of points from which rider or a rider can be picked-up and dropped-off. Formally, route R_j is (A_j, B_j) where A_j and B_j are longitude-latitude pairs.

Let us assume that a rider who wants to travel from one location to another has a probability of traveling along a route. The probability dependant on the total walking distance from the rider's current location to the route's pick-up location and from the route's drop-off location to the desired drop-off location.

Specifically, if a rider intends to do ride ri which goes from a point pi (lng, lat) to a point di then the probability of this rider traveling along a route R_j is:

$$f(H_j, r_i) = e^{-(M(P_j,p_i)+M(D_j,d_i))}$$

Where we can assume that M is the Manhattan metric between (lng, lat) pairs (x_p, y_p) and (x_q, y_q).

$$M((x_p, y_p), (x_q, y_q)) = \alpha|x_p - x_q| + \beta|x_p - x_q|$$

If there are multiple routes available to a rider, the rider will take the one that has minimum walking involved i.e. .

$$j^* = argmax_j f(R_j, r_i)$$

Out of a set of routes $\{R_j\}$ we want to find top N popular ones. In this case we chose five however N can be any constant value desired.

3 The Data

The data for contained 9 features. the data containing relevant historical information about the ride of the riders. such as desired pickup location, desired drop-off location, timestamps, trip distance, duration, etc.

4 Background

Clustering is an unsupervised learning approach of partitioning the data set into clusters in the absence of class labels. The members of a cluster are more similar to each other than to the members of other clusters. One of the most fundamental and popular clustering techniques are K-Means [1] and Fuzzy K-Means [2] clustering algorithms. K-Means clustering technique uses the mean/centroid to represent the cluster. It divides the data set comprising of n data items into k clusters in such a way that each one of the n data items belongs to a cluster with nearest possible mean/centroid.

K-Means is very convenient to understand and implement has a major drawback. In case of extreme valued data items, the distribution of data will get uneven resulting in improper clustering. This makes K-Means clustering algorithm very sensitive to outliers and noise, thereby reducing its performance too. K-means is also does not work quite well in discovering clusters that have non-convex shapes or very different size. This calls for another approach to clustering that is based

on similar lines, yet is robust to outliers and noise which are bound to occur in realistic uncontrolled environment. Even recently Bachem at el [12] talked at NIPS 2016 how critical seeding is and how the task of finding initial cluster centers – is in obtaining high quality clusterings for k-Means.

K-Medoids clustering [3] is one such algorithm. Rather than using conventional mean/centroid, it uses medoids to represent the clusters. The medoid is a statistic which represents that data member of a data set whose average dissimilarity to all the other members of the set is minimal. Therefore a medoid unlike mean is always a member of the data set. It represents the most centrally located data item of the data set. The working of K-Medoids clustering [3] algorithm is similar to K-Means clustering [1]. It also begins with randomly selecting k data items as initial medoids to represent the k clusters. All the other remaining items are included in a cluster which has its medoid closest to them. Thereafter a new medoid is determined which can represent the cluster better. All the remaining data items are yet again assigned to the clusters having closest medoid. In each iteration, the medoids alter their location. The method minimizes the sum of the dissimilarities between each data item and its corresponding medoid. This cycle is repeated till no medoid changes its placement. This marks the end of the process and we have the resultant final clusters with their medoids defined. K clusters are formed which are centred around the medoids and all the data members are placed in the appropriate cluster based on nearest medoid.

K-medians clustering algorithm is also an important clustering tool because of its well known resistance to outliers. K-medians, however, is not trivially adapted to produce normalized cluster centers. For some applications it is more desirable to use 1- norm distance (also known as Manhattan distance, denoted here as $\|.\|_1$ 1) to measure the distance between points. The reason stands behind it is the fact that cluster center that minimizes 1-norm distance to all points within that cluster is the median of that cluster, and using the median instead of the mean tends to be more robust to outliers.

5 Approach and numerical experiments

Our approach for finding the popular routes is to use multidimensional clustering algorithm in order to model the longitude and latitude points. K-Means minimizes within-cluster variance, which equals squared Euclidean distances. In general, the arithmetic mean does this. It does not optimize distances, but squared deviations from the mean.

The k-median problem has received significant attention for decades, primarily in computer science and operations research. k-medians minimizes absolute deviations, which equals Manhattan distance. In general, the median does this. It is a good estimator for the mean, if you want to minimize absolute deviations, instead of the squared ones. Therefore the way for one to choose the clustering method is to consider the metric distance. Whether, the distance is squared Euclidean distance, use k-means whether the distance is Taxicab metric, use k-medians or whether, other distance, one can use k-medoids.

Clustering methods like k-means require a-priori knowledge with regard to the number of clusters aka choosing K. Although our task is an unsupervised learning problem and we do not have any ground truth at all, we can use the clustering method under the assumption that we will set K as

the number of the popular routes desired. That way, we will overcome K a priori obstacle meaning we can reduce the problem to top N routes discovery.

For the clustering method, we tried all three. We chose to use the manhattan distance as the metric distance in the three clustering algorithms: K-means, K- medoids and K- medians. We used coordinate descent algorithm over all the points in the cluster for the route discovery. We found out that only four features from the dataset were enough for us to solve accurately the problem and therefore the latitude and longitude points (2 each) were chosen as an input.

Pre-processing: we will pre-multiply x by alpha and y by beta in order to avoid dealing with kernels and to obey to the data distribution which can be formulated as :

$$M((x_p, y_p), (x_q, y_q)) = \alpha|x_p - x_q| + \beta|x_p - x_q|$$

Now one can observe that the metric is the sum of L1 distances therefore, a desired path is a 4-dimensional vector (x_p, y_p, x_q, y_q)

The K-medians algorithm [4, 5] is thus a very powerful alternative. Theoretically, k-median exactly minimizes:

$$M(P_j, p_j) + M(D_j, d_j)$$

Therefore, the clustering final model leaned on the k-medians [7,8] . We also checked the k-medoids and the k-mean while the euclidean metric was replaced with manhattan or L1 metric. Those were equivalent k-means which finds centers for each path (with manhattan metric instead). Note that in a case we were using kmean with the manhattan distance we could get route which is an existing route from a rides' dataset.

In addition, we optimize over the hyperparameters of the clustering model : Delta, number of iterations. Finally we extracted the centers of the clusters found.

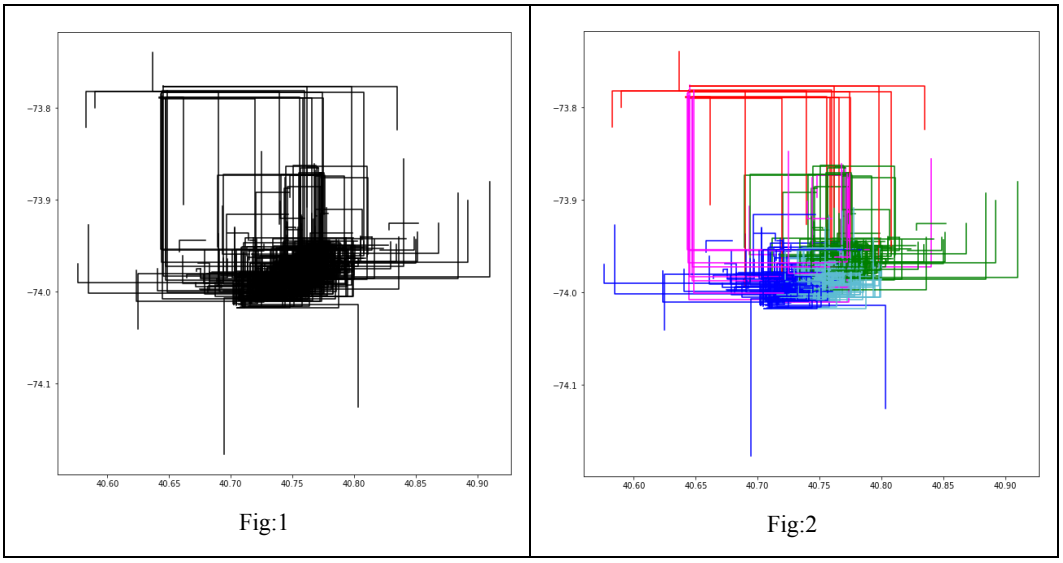

Fig:1

Fig:2

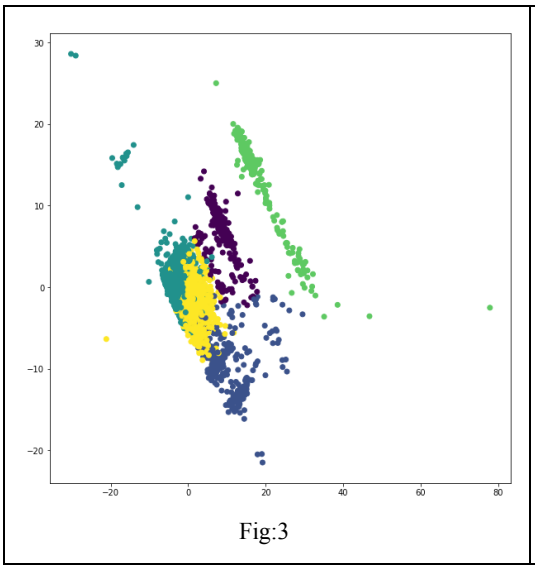

Fig:3

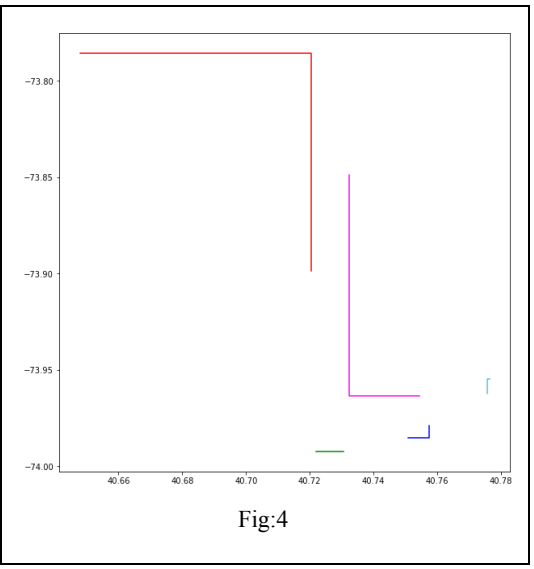

Fig:4

Fig:1 L1 routes before clustering, Fig:2 L1 routes after clustering by K-Medians, each color represents 1 of the 5 clusters, Fig:3 Each point represent a route and the color represents the cluster they belong to. Fig:4 The centroid routes for each of K-Median cluster

6 Conclusions and future work

k-medians gained both observable and fast results. In addition, k-medians be adjusted to maximize the sum of probabilities using the function f

$$f(H_j, r_i) = e^{-(M(P_j,p_i)+M(D_j,d_i))}$$

In order to do so we need to compute the derivatives of the exponent for the coordinate descent which are straightforward since it's an exponent. The order of the algorithm is O(1).

References

[1] Hartigan, J. A.; Wong, M. A. (1979), "Algorithm AS136: A K-Means Clustering Algorithm". Journal of the Royal Statistical Society, Series C 28 (1): 100–108.

[2] Bezdek, James C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms.

[3] Kaufman, L. and Rousseeuw, P.J. (1987), Clustering by means of Medoids, in Statistical Data Analysis

[4] P. S. Bradley, O. L. Mangasarian, and W. N. Street, "Clustering via Concave Minimization," in Advances in Neural Information Processing Systems, vol. 9, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, pp. 368-374.

[5] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data: Prentice-Hall, 1981.

[6] Mohammad Mahdian† Yinyu Ye‡ Jiawei Zhang ,Approximation Algorithms for Metric Facility Location Problems

[7] K-median Algorithms: Theory in Practice

[8] Yakar, Tal Ben, et al. "Bilevel Sparse Models for Polyphonic Music Transcription." ISMIR. 2013.

[9] Soland, Richard M. "Optimal facility location with concave costs." Operations Research 22.2 (1974): 373-382.

[10] Meyerson, Adam, and Brian Tagiku. "Minimizing average shortest path distances via shortcut edge addition." Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques. Springer Berlin Heidelberg, 2009. 272-285.

[11] Rossi, Federico, Rick Zhang, and Marco Pavone. "Congestion-Aware Randomized Routing in Autonomous Mobility-on-Demand Systems." arXiv preprint arXiv:1609.02546 (2016).

[12] Olivier Bachem. "Fast and Provably Good Seedings for k-Means." NIPS 2016.