Density Based Spatial-Clustering for Road Landmarks Using Taxi Trajectories

by Richard Li, Anthony Chu, Yifan Zhang

December 13, 2019

Abstract

Landmarks are the critical portions of roads that are most frequently traversed within the city. The aim of this project is to utilize density based clustering as landmark detection algorithms based on taxi trajectories within the city of Porto, Portugal. For our analysis, we compared the results of our own DBSCAN clustering models with the traditionally used naive count model for landmark evaluation. Using empirical methods and index based clustering evaluation methods, we derived optimal hyperparameters to distinguish significant, dense point clusters in our dataset from surrounding noise. Using these hyperparameters, our DBSCAN model provided a far better evaluation of inner-city road landmarks in comparison to the naive model. This landmark detection algorithm revealed critical road segments within our chosen dataset in Porto, and can be potentially applied in urban analysis for improved urban planning, advertising placement, and traffic prediction in other cities as well.

1 Introduction

As the world becomes increasingly urbanized, efficient traversal of road networks is critical for a city's overall success. Consequently, finding the most traversed road segments within a city would provide critical information about the overall topology of the city as a whole. These road segments (approximately the length of a city block), constitute 'landmark' streets within the city's network and may include critical road junctions, important urban locales, and frequently used highways. Landmark detection is a critical component for multiple studies in urban

computing and the detected landmarks is frequently used as the backbone for analyzing topological road networks in cities [1][2]. Thus there is an obvious need for a robust landmark detection algorithm to determine the flow of transportation within the city for urban improvement, city planning, road network analysis, and collection of transportation data.

Currently, the only method used for landmark detection rely on a naive counting algorithm that attempts to locate the roads with just the highest number of traversals. While this algorithm is successful We believe that this method often fails to take into account the intrinsic geospatial properties of road networks; these methods end up favoring longer road segments (IE highways) and are far more sensitive to noisy data, a significant issue in GPS geolocation. In response, we propose an alternative solution that relies on density based clustering algorithms. This method allows us select road 'zones' based on the density of GPS points of individuals that frequently traverse through these areas and select from these zones the best candidate road segments.

To compare the results between the traditional naive count method and our density based clustering method, we compared an opensource dataset of all taxi trajectories within Porto, Spain between 2013-2014. We believe that taxi drivers, due to the nature of their very occupation, would be expert navigators of their local road networks. Consequently, these drivers would most likely use the best road networks in order to arrive at their given destinations.

Figure 1: Map of Porto

2 Methods

2.1 Dataset Overview

The taxi dataset we used was taken from Kaggle.com [3] as part of an opensource data science competition. The data came from the trajectories of 442 taxis within the city of Porto between 07/01/2013 to 06/30/2014 in Porto, Portugal and its surrounding districts. The primary features we utilized from this data set were the *TRIPID, TIMESTAMP*, and *POLYLINE* features. In particular, we focused our analysis on the *POLYLINE* feature, which consisted of an array containing the longitude, latitude coordinates of each taxi trip sampled in 15 second intervals. For our project, we used the data from 1,000 total taxi trips for our analysis in order to ensure efficient analysis of our data due to both software and hardware limitations (See Methods and Results).

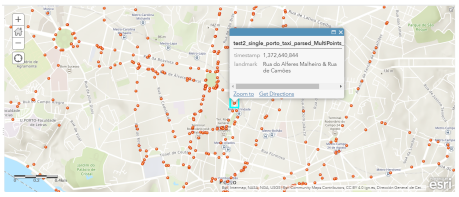


Figure 2: A Brief Overview of Data Points

2.2 Data Preprocessing

Before the we can properly analyze our data, various layers of preprocessing had to be performed to clean the raw dataset. Firstly, the dataset had a Boolean feature called *MISSINGDATA* which, if *TRUE*, signified that the *POLYLINE* feature was missing trajectory points IE points that were separated by more than 15 second intervals. However, the data set also had taxi trips where the *POLYLINE* data had empty arrays and trajectories with only one GPS sample point, signifying that certain trips were canceled and/or the GPS geolocator failed to work. These faulty data were all filtered out as the first stage of data preprocessing. The second stage of preprocessing was far more computationally intensive and required mapping each *POLYLINE* trajectory point to a road segment within Porto. Road segments were delineated by intersections between roads with other roads i.e. 33rd Street/7 Ave, 34th Street/8 Ave, 34th Street/9 Ave would all be considered consecutive but separate road segments. For this aspect of the project we used the geospatial

tool ArcGIS to reverse-geolocate each trajectory point in *POLYLINE* to a real world street intersection closest to that point. If consecutive trajectory points from the same taxi trip were mapped to the same road segment (i.e. long trajectories mapped consecutively to a highway or a overly long road with few intersections), then we keep the first point geolocated within the road segment and filter out the other consecutive points within that particular trajectory. This ensures that when a clustering algorithm is performed on the data set to find popular road segments, the clusters chosen from the data points have, at most, only one point mapped for each road for each taxi trip. This is critical to ensure that landmark selection will only select road segments with the most traversals over multiple taxi trips without having duplicates creating bias for longer roads.

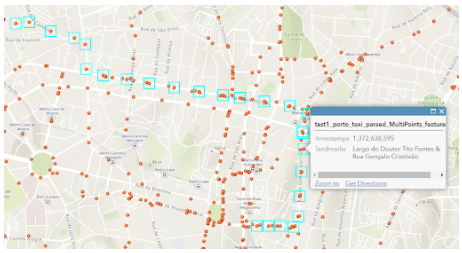


Figure 3: The Data Points after Preprocessing

2.3 Naive Count Modeling

The first method we used to identify the landmarks in our data set was to simply count which roads appeared most frequently in the taxis' GPS data, with no regard for other factors. This algorithm is currently the most frequently used landmark detection algorithm in urban trajectories research [1]. The landmarks we identified through this method will serve as a benchmark to compare with the landmarks identified in our density clustering model.

This method is quite limited, especially since we only paid attention to the closest intersections instead of the actual road segments using ArcGIS. It tends to be biased towards long roads and highways, or areas with one main intersection with no regard for the intersecting roads themselves. If the API we used to match a (latitude, longitude) points to an address that was more robust, we'd likely be able to remedy some of the pitfalls caused by just counting the frequency that roads appear in the data set.

Through the clustering model we develop, we hope to identify landmarks more accurately than we identified them through the naive method. The naive method can be summarized briefly as:

Algorithm 1: Query Modeling

Result: A set of landmarks V
A = Trajectory Archive;
C = Frequency dictionary for roads;
for *Trajectory T in A* **do**
  for *road segment R in T* **do**
    $C[R]$++
  end
end
$V = \text{Most}_T raveled(C,k)$

2.4 DBSCAN Modeling

Geospatial trajectory points are natural candidates for unsupervised clustering algorithms. Real world geospatial coordinates have a low Vapnik–Chervonenkis dimension since the data only requires two fields for analysis: longitude and latitude. DBSCAN modeling functions as a two step process. A DBSCAN model selects the top zone clusters and then runs the naive count algorithm only within these zone clusters to select the top landmark road segments from each zone. The rationale for using a density based clustering algorithm is that it addresses two major drawbacks to simply just using the naive counting model on its own: the naive model requires far more noise processing compared to DBSCAN modeling and is far less flexible in returning information about the road network in comparison to DBSCAN.

Firstly, since the naive model chooses landmarks based on counting which roads have the most traversals, it requires the GPS data to be heavily cleaned for noise. GPS coordinates are infamous for being noisy, oftentimes resulting in trajectory points that stray far from the roads that they traverse through. Consequently, map-matching algorithms are needed to clean these points to ensure that the naive counting algorithm can select the correct roads with the most traversals [4]. However, map matching algorithms oftentimes have multiple components, each with heavy polynomial runtimes: this becomes a significant issue when map match-

ing must be performed on all points within a dataset, especially in cases with big data analysis. In contrast, since DBSCAN selects small zone clusters that only make up small portions of the overall dataset *before* running its naive count algorithms, the number of overall points that must be map matched is far smaller if map matching is performed only on trajectory points within these zones, vastly improving overall runtime.

Furthermore, the naive model only returns the same top landmarks for its analysis. While this is beneficial in choosing the most efficient roads for city traversal, it lacks the flexibility in choosing different types of roads to perform its analysis on. In the case of the Porto dataset, the most traversed roads with the naive model were all highways. However if the user wanted to find information about most traversed local roads or inner-city streets, then any traversal data found on highways must be parsed out manually. In contrast, density based algorithms provides for a natural way to demarcate data between smaller roads and highways with its density based analysis (See Hyperparameter Selection). By applying specific parameters for the DBSCAN algorithm, we can find landmark roads from different road network clusters of similar density. For the explicit purposes of this experiment, we chose inner-city local roads as our point for analysis but modifications upon the type of density based clustering algorithms used can result in more varied types of analysis on different road networks (See Clustering Evaluation). Finally, while not a direct component in finding landmarks, the cluster zones found with density based clustering can possibly be used to gather further information about hotspot roads of travel within a city network.

For the DBSCAN algorithm (See Figures 4), we used the point clustering algorithm native to ArcGIS Python API which is a dedicated DBSCAN algorithm for geospatial data. This eliminated the need for us to convert our geocoordinates onto a three dimensional spherical surface for analysis. This DBSCAN algorithm took two parameters, a cluster radius ε and a value K that forced each cluster to have at least K minimum number of points within radius ε to be recognized as a cluster. Since ε and K are codependent values, we had to run clustering analysis to find the best parameter selection for both values. An example of an ideal cluster for our analysis is shown in Figure 5. This cluster contains about 4-5 road segments in their entirety, allowing us to select the top landmark roads from this far smaller sample size. This will be specified in more detail within the *Parameter Selection* section of this paper.

Figure 4: DBSCAN Example

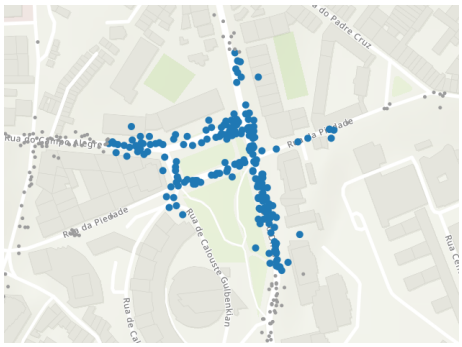


Figure 5: DBSCAN Optimal Result Example

when deriving landmarks from our trajectories, we can presume that for any j landmarks unknown to us, they will be found within z zone clusters where $j \leq z$ thus.

2.5 Optimal Number of Landmarks

If we want to find the top landmarks within a city, we also need to specify the number of landmarks we are searching for. This is especially critical because we want to approximately match the number of landmarks we are searching for with the number of clusters found by our learning algorithm. The quantity of clusters are critical for our analysis, as too many landmarks will result in us possibly interpreting noise as landmark locations, while too few landmarks will result in gaining little key information on the city layout. We propose to use cross-validation and our Naive Count Algorithm in order to evaluate the optimal number of landmarks to find. Our goal for clustering is to return an optimal amount of zone clusters $z \geq j$. Consequently, by finding the optimal number of landmarks, we will have our lower bound to our parameter z.

2.5.1 Landmark Appearance Rate

The most basic and intuitive method we developed to evaluate our landmarks is a metric we will call the *Landmark Appearance Rate*. We first identified a set of landmarks with an arbitrary method from a training set. Afterwards, we took a test set and counted how many times each landmark identified from the aforementioned training set appeared in this test set. By the end of this each landmark should have a percentage value associated with it that denotes how often it appeared in the test set.

The final step is averaging the aggregate value of these percentages to arrive at the Landmark Appearance Rate. By taking the average of each landmarks' percentage, we penalize the model for finding landmarks that do not appear very often in the test set. For example, with a set of 3 landmarks and their appearance percentages $L1$, $L2$, $L3$, the Landmark Appearance Rate I would be:

$$I = \frac{L1 + L2 + L3}{3} \tag{1}$$

This should give a rudimentary metric for how well we identified our landmarks as intuitively landmarks, as previously defined, are the areas that should show up most often in the data set. We performed this process multiple times on the data set as a form of cross validation to account for cases where the landmarks we identify in the training set never appear in the test set. We took the average of all the percentages we find in each "fold" to arrive at our final percentage value. In Figure 6 we have a plot for the average landmark appearance rate for different amounts of k landmarks picked.

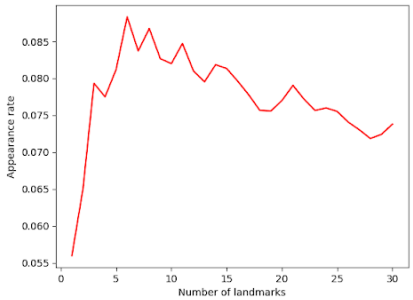


Figure 6: Landmark Appearance Rate VS. Number of Landmarks

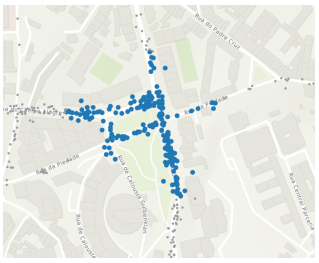
After finding landmarks in multiple training sets, we found that even the most frequent landmarks usually only appear around 10-12 percent of the time in the training set, which makes sense since even the most heavily traveled roads aren't used by disproportionate amount of taxis. The values of 6-8 percent seen in Figure 6 are relatively reasonable and in line with how often landmarks appear in the training set. Examining Figure 6, we see that the plot peaks around 5-7 landmarks chosen. So the optimal amount of landmarks to choose without the risk of selecting landmarks that might appear less is around 5. This metric serves more as a sanity check and a way to identify the optimal number of clusters more than anything, and other evaluation methods may be more suitable.

3 Hyperparameter Evaluation

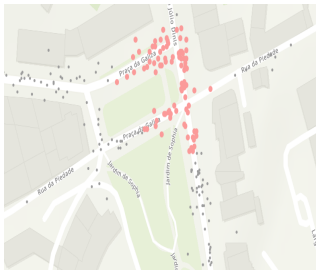
3.1 Hyperparameter Analysis

Let us define the minimum area density variable d where for a given distance unit of measurement,a minimum of d number of points will be found within it. For our project we used meters as our measurement unit with d defined as $d = \varepsilon/K$. Variable d is essential for our analysis because finding optimal values of d would help us derive significant dense clusters within our dataset without accidentally parsing noise as data.

If ε is too low or K is too high, then the clustering algorithm would not adequately detect neighboring points within our data. For small values of ε, the clusters generated would be smaller with many significant portions of clusters being deemed as noise. This is due to the limited radii surrounding each point so that potential core points would not form due to not having enough K minimum points in its vicinity. Consequently, higher values of K resulted in too many points needed for cluster generation per core point, and thus no clusters are formed even for dense, highly traversed areas; low values of ε and high values of K can create densities that are higher than the taxi data we are inputting, forcing large swaths of dense points to be mistakenly marked off as noise. This can be seen in Figures 7 and 8.

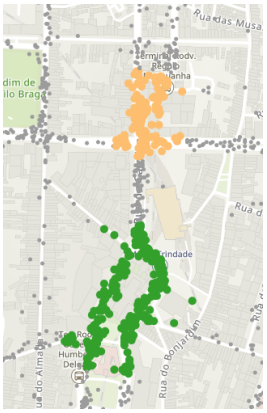


(a) Optimal Parameters of Cluster of Several Shorter Road Segments

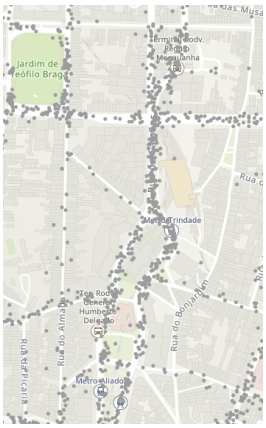


(b) Parameters That Mistakenly Resulted in an Overly Dense Cluster With Cut Off Road Segments

Figure 7



(a) Optimal Parameters of Clusters of Several Longer Road Segments



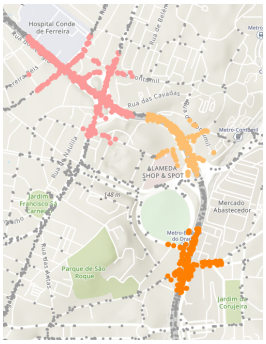
(b) Parameters That Mistakenly Result With All Clusters Marked as Noise

Figure 8

Inversely, if ε is too high or K is too low, then the clustering algorithm would more frequently select imprecise and overly large clusters, causing the algorithm to be far more sensitive to noise. Larger values of ε resulted in the radii of individual points to be too long, with many noise points being counted as core points due to the range of ε picking up enough noise points to reach threshold K. Lowering K resulted in lowering this threshold even further, allowing noise to be more easily labelled as core points.

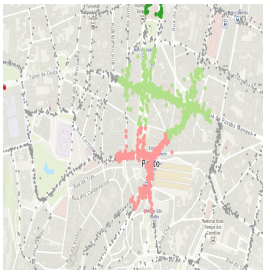
Evidently, both resulted in more frequent clusters as seen in Figures 9-10 below. One key consideration we had for choosing the best parameters was that for high ε values, distinct clusters would be connected together. The labelling of new points would have other core points within their extended ε range, often resulting in noise areas of similar density aggregated together as 'megaclusters'. Megaclusters form due to the minimum density d being too small of a value, thus resulting any noise areas with a given density higher than that of d being labelled as clusters. However, these areas with same minimum values of d are tricky to find optimal parameters, as if we decrease ε and/or increase K we risk marking actually significant points as noise. This makes finding significant roads with high densities over longer distances (such as highways) difficult to separate from noise. This can be seen in Figure 10b. The issue of solving this megaclusters issue explained in greater detail in the following section.

(a) Optimal Parameters that Correctly Label Highway as Noise

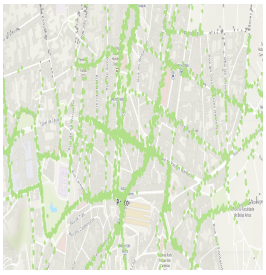


(b) Parameters that Mistakenly Result in Highway and Nearby Roads Labelled as Clusters

Figure 9



(a) Optimal Parameters of Clusters of Several Shorter Road Segments



(b) Parameters that Mistakenly Result in a Megacluster of All Neighboring Points

Figure 10

3.2 Clustering Evaluation

We decided the optimal values for ε and K based on three commonly used evaluation methods for clustering algorithms.

1. **Davis-Bouldin Index**
   The Davis-Bouldin index (DBI) measures the average similarity of each cluster with its most similar neighboring clusters, judging the clustering algorithm between the spatial distance between clusters. The score is determined by the ratio between intra-cluster distance with inter-cluster distance. Consequently the smaller the DBI score is, the better the clustering algorithm is at choosing specific clusters of data that are farther apart from each other.

2. **Silhouette**
   The Silhouette method follows a similar logic as DBI but instead compares the average of the differences between each cluster and their closest neighboring cluster. Silhouette ranges from $[-1,1]$ with the higher the silhouette score, the better the clustering algorithm is at deciding if each point is clustered correctly based on its distance to other neighboring clusters. Consequently, each point, even parsed noise points, are evaluated with this method.

3. **Calinski-Harabasz**
   The Calinski-Harabasz method is distinct from the previous methods in that it does not use cluster distances as a measure of evaluation. Rather, it relies on the ratio of intra-cluster variance to inter-cluster variance for its index.

The higher the Calinski-Harabasz index is, the better the clustering algorithm is at choosing specific clusters of data that have similar degrees of variance from the cluster centroid.

Using the information from the hyperparameter analysis as listed in our previous *Hyperparameter Analysis* section, we tested values of ε in the range $[50, 125]$ and K in the range $[75, 125]$. We chose these values because 50-125 meters best approximated the length of a road segment within the city of Porto. In addition, K was determined through empirical testing by evaluating different values until we settled on zone clusters that encompassed between 4-7 road segments, ensuring we have clusters that are meaningful in the information they return yet avoiding megaclusters from forming. Following this, we ran clustering evaluation on these datasets to receive the following results:

```
FINAL_DBSCAN_50M_75p_c
Davies-Bouldin Index:  1.6121343779595747
Calinski-Harabasz:  63.28197732406292
Silhouette Score:  -0.4640121982212512
```

Figure 11: epsilon = 50 Meters, K = 75 Points

```
DBSCAN 100M 100pc
Davies-Bouldin Index:  1.6943752140214121
Calinski-Harabasz:  97.38920638218863
Silhouette Score:  -0.46634464670734005
```

Figure 12: epsilon = 100 Meters, K = 100 Points

```
DBSCAN 100M 125pc
Davies-Bouldin Index:  1.674173775391041
Calinski-Harabasz:  113.01990434155451
Silhouette Score:  -0.4322127869708965
```

Figure 13: epsilon = 100 Meters, K = 125 Points

```
FINAL_DBSCAN_90M_125p_c
Davies-Bouldin Index:  1.6052963704179586
Calinski-Harabasz:  98.95322967659317
Silhouette Score:  -0.43400464707285913
```

Figure 14: epsilon = 90 Meters, K = 125 Points

For our analysis we chose DBI and Silhouette scoring to formally evaluate zone clusters that are adjacent to each other. However, we discovered that the DBI and Silhouette scores are only negligibly different from each other. In retrospect, this makes sense due to the 15 second interval geolocating for the taxi trips. Even after we preparse for one point per taxi trip for each road segment, the cluster zones we are analyzing will cover the entirety of the roads, resulting in generally larger clusters that cover entire portions of the road network. Furthermore, urban road networks naturally have its most frequently used roads closer to each other rather than sparsely spread throughout the city. Both of these factors resulted in the clusters we found in our analysis to have larger intra-cluster distances while also having smaller inter-cluster distances.

To provide a more adequate factor of analysis, we chose Calinski-Harabasz evaluation as the main form of evaluation of our clustering parameters. Due to certain road segments being used more than others, calculating the variations in cluster density is the most ideal form of concluding which hyperparameters worked best for our given data. However, we also still used DBI and Silhouette evaluations in order to provide a quantitative measure for megacluster detection, so as not to only rely on just empirical evidence.

From these clustering evaluation results, we observed that the hyperparameters $\varepsilon = 100$ meters and $K = 125$ yielded on average the best results with our evaluation methods. These hyperparameters had the highest value for the Calinski-Harabasz score of 113.01 and its DBI and Silhouette evaluations were agreeable when compared to the other hyperparameter evaluations as well. In particular, DBSCAN($\varepsilon = 100, K = 125$) yielded the ideal number of clusters according to our Landmark Appearance Rate method(approximately 5-6 clusters) and for selecting significant landmarks in context of location (See 'Results'). In contrast, a smaller ε and/or larger K resulted in clusters that select less than 3-5 blocks, often cutting off entire road segments. Larger ε and/or smaller Kvalues likewise resulted in megacluster formation.

Figure 15: epsilon = 100 Meters, K = 125 Points Visualization

These are the best results for road segments between 50-125 meters, the average length for local roads within an inner-city road network. To find longer, intra-city road segments typically found outside or surrounding cities, such as highways and interstates, the DBSCAN models need to be modified from our local road model. Since highways are longer than local roads, the density d for highway segments will be lower than that for local roads. However, if we set our hyperparameters to find such highways, the results returned will contain mega-clusters due to local roads having densities greater than equal to the smaller d given. Furthermore, since DBSCAN chooses points within a circle with radius ε, the clustering algorithm is optimal for finding multiple dense road segments (as the shape of a network of 5-7 connected road segments roughly form a circular shape), but not for finding single long dense roads, such as highways. In order to address this issue, we suggest alternative density based clustering algorithms that does not rely on radial based density measuring. An updated DBSCAN that searches for dense points through vectors rather than circles can be one alternative solution; the OPTICS clustering algorithm is also suggested to find specific dense road segments as individual clusters within a network. Alternatively HDBSCAN, a modified DBSCAN algorithm that selects varied distances of ε rather than utilizing a constant value, can be used for both highway landmark evaluation and local landmark evaluation in the same analysis (See Figure 15).

Figure 16: HDBSCAN of with K=125 Points. Francisco Sá Carneiro Airport and highway A20 are both chosen as landmarks, along with three other landmarks within downtown Porto as well.

Our choice of the conventional DBSCAN algorithm for our analysis is due to our project focus on inner-city road networks together rather than specific, extended roads typically found outside of cities.

3.2.1 Conversion from Clusters to Landmarks

Once the DBSCAN hyperparameters were selected, we returned the clustered points as members $p_1, p_2, ...p_n$ with n being the number of points within each cluster. We defined each cluster as sets $C_1, C_2 ... C_t$ with $t \geq j$ being our total number of clusters and let each $C_i = p_1, p_2, ..., p_n$. Next, for each C_i, we ran our Naive Query Method on each cluster and select the single most traveled landmark. If $j = t$, we returned the t landmarks. If $j < t$, then we choose the top J landmarks with the most counts from the Naive Algorithm out of t, and disregarded the rest. This allows neighboring clusters such as those shown in Figure 10a. to have their road segments counted fairly, even if the two clusters should optimally be considered a single cluster.

3.2.2 Evaluation Limitations

The main limitation for our analysis evaluation was that the ArcGIS API that we used to run our DBSCAN analysis relied on a credit based system and every time a clustering analysis was performed upon our data, it resulted in approximately a 3 USD fee. Thus, a budget for this project became a key issue that held us back from selecting the best parameters for our DBSCAN algorithm. If we were not limited within our budget, our group would have used cross-validation methods

for ε at a range between $[50, 300]$ meters in 10 meter intervals and for K at a range between $[50, 200]$ points within 5 point intervals. Furthermore, we would have implemented stops within the cross-validation if our total number of clusters falls under 5 clusters or over 15 clusters, as this is the best range as given by our landmark detection model.

4 Results

Here are the five landmarks returned by both the DBSCAN($\varepsilon = 100, K = 125$) and Naive Query algorithms:

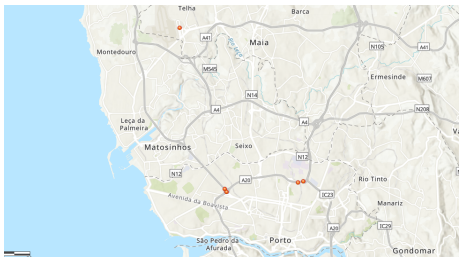


Figure 17: Naive Query Results

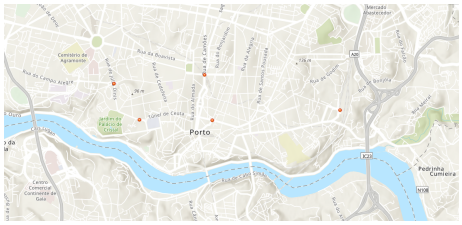


Figure 18: DBSCAN Results

The Naive Count algorithm chose four of its landmark positions on the major A20 highway that circles Porto. Its final, fifth point is located by Francisco Sá Carneiro Airport, the nearest airport to Porto. One of the most immediate issues with the naive model is its inability to choose landmarks based off the inherent geography of the roads, as it chose landmarks in very close proximity to each other, as seen in Figure 18.

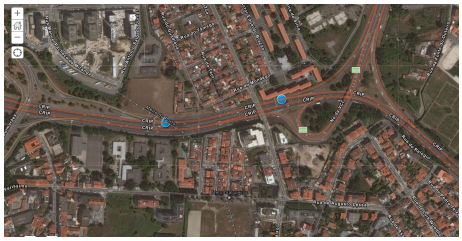


Figure 19: Naive Method Close Proximity Landmarks

Furthermore, while it is intuitive that the landmarks would be located on highways as they allow for higher densities of vehicles and higher speed limits, such information does not reveal much information regarding Porto's local urban network. Since the highways in Porto only runs across the border outskirts of the city, such data derived from this analysis reveals little regarding the actual local transportation options within the city of Porto itself.

In contrast, the DBSCAN algorithm returned varied points within the city of Porto, especially those in the downtown areas of Porto, which contain the major urban roads within the city. Of the streets returned from the DBSCAN model, we have road segments connected to the railray lines, critical metro stations, tourism/event sites, and important road junctions. Furthermore, we avoided the highway 'trap' because the $\varepsilon = 100$ meters forced the clustering algorithm to avoid long stretches of highways where the points were too sparse to total up to K points within our given radius. Similarly, due to the airport Francisco Sá Carneiro being connected via a highway, the DBSCAN at our chosen parameters failed to detect it due to the sparseness of the trajectories at the particular location. For the extent of this research though, where we are primarily focused on finding landmarks within urban areas and are less focused on critical highway junctions, the DBSCAN method provided with a far better overview of critical roads within the city of Porto itself.

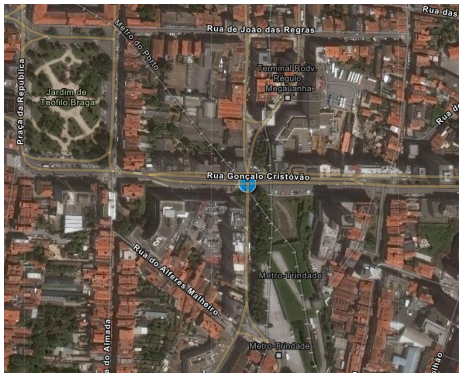


Figure 20: DBSCAN Landmark 1: Rua da Estação/Travessa de Miraflor

Landmark 1 is located directly at Campanha Railway, a historic railway and critical commuter station within Porto's Metro System. It is the first of the two railways within Porto.

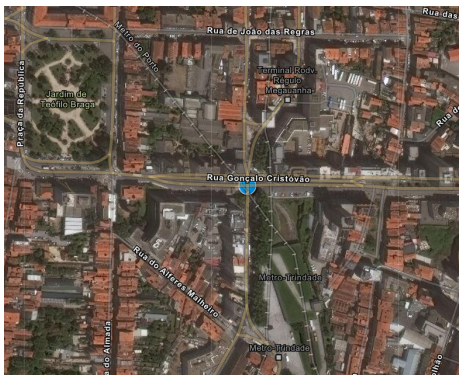
Figure 21: DBSCAN Landmark 2: Rua de Passos Manuel/ Rua de sá da bandeira

(a) Landmark 2 located nearby to Sao Bento Railway, another critical railway and commuter station within Porto's Metro System. It is the second of the two railways in Porto.

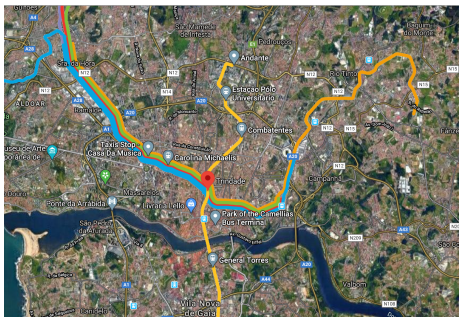
(b) Landmark 2 is also located nearby Liberdade Square (Liberty Square) and is Porto's main square. The location serves as the connecting point between the historic portion of the city and the modern portion of the city.

Figure 22

Figure 23: DBSCAN Landmark 3: Rua de Gonçalo Cristóvão/Rua de Camões



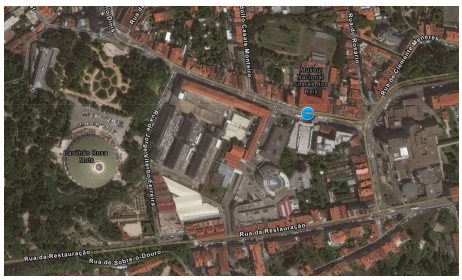
(a) This landmark is adjacent to Metro-Trinidade, a major metro stop within Porto. This stop accounts for 15 to 25 percent of all ticket purchases in Porto Metro.



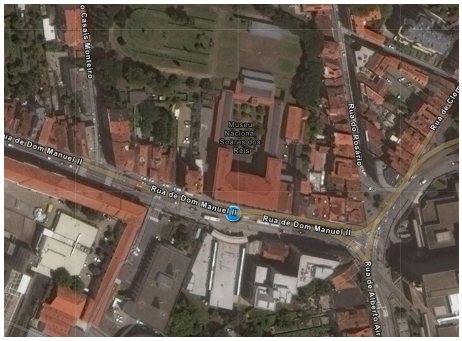
(b) As shown in this metro map of Porto, Trinidade Station is located as the sole transfer point for the two separate metro lines within the city.

Figure 24

Figure 25: DBSCAN Landmark 4: Rua de Dom Manuel II/Túnel de Ceuta



(a) This landmark is located close to Pavilhão Rosa Mota (Super Bock Arena) a major arena pavilion within the city.



(b) The landmark is also located directly beside Museum Soares dos Reis, one of the oldest and most well known museums within the entire city. This site marks a significant site for tourists of the city.

Figure 26

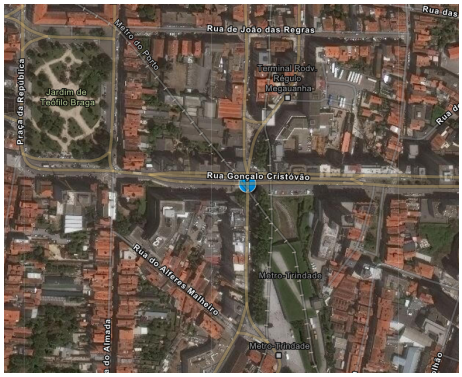


Figure 27: DBSCAN Landmark 5: Rua de Júlio Dinis/Rua da Piedade

This road landmark include roads that branch into significant highways surrounding Porto.

From these results, it is evident that in comparison to the traditional naive count model, DBSCAN analysis provides for a more robust landmark detection of road segments within Porto and is a better alternative model for analyzing local road usage.

5 Conclusion

Determining the flow of traffic within urban infrastructures is critical in gathering information about city networks. Our DBSCAN algorithm helps address this as it smartly evaluates road traversals of taxi trips by relying on the inherent geographical locations for its cluster evaluation, in contrast to our Naive Count model. Furthermore, the DBSCAN algorithm has an average case runtime of $O(log(n))$ and worst-case runtime of $O(N^2)$, making it faster than the Naive Count method, which has a $O(N^2)$ runtime.

For future research, we propose using a cross-validation parameter selection to have a more rigorous process of selecting the best parameters, as we discussed in our *Clustering Evaluation*. Furthermore, while this project was focused on locating urban landmarks within actual cities, evaluating frequently traversed highway segments on the outskirts of a city can also reveal critical urban movement patterns as well. For such an evaluation of inter-city highways, a traditional usage of DBSCAN and its constant radius based clustering may not be the best model for

evaluation. For such alternate cases, a modified DBSCAN algorithm that relies on vector based density evaluation or OPTICS is best suited to find clusters of sparser road segments. Alternatively, a hierarchical density-based spatial clustering of applications with noise (HDBSCAN) would provide density measures that return results for both urban centers and critical highways can be used as well. Lastly, the clustering upon multiple road segments may also reveal critical road network information besides that of landmark selection. Using a DBSCAN algorithm with larger parameters at a proper ratio of ε and K can also find critical urban road networks instead of simply important landmark road segments.

6 References

[1] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, Yan Huang T-Drive: Driving Directions Based on Taxi Trajectories Proceedings of 18th ACM SIGSPATIAL Conference on Advances in Geographical Information Systems, November 2010
[2] Yu Zheng,
Trajectory Data Mining: An Overview
ACM Transactions on Intelligent Systems and Technology, May 2015, Article No.: 29, https://doi.org/10.1145/2743025
[3] Kaggle.com Taxi Trajectory Data
https://www.kaggle.com/crailtap/taxi-trajectory/data
[4] Jing Yuan, Yu Zheng, Chengyang Zhang, Xing Xie, Guang-Zhong Sun
An Interactive-Voting Based Map Matching Algorithm, Eleventh International Conference on Mobile Data Management, May 2010
[5] Introducing Porto
Praça da Liberdade
https://www.introducingporto.com/praca-da-liberdade