Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 130 (2018) 359–367

Procedia Computer Science

www.elsevier.com/locate/procedia

The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018)

Spatio-Temporal Clustering of Time-Dependent Origin-Destination Electronic Trace Data

Daphne van Leeuwen[a,b,*], Joost Bosman[a], Elenna Dugundji[a,b]

[a] *Centrum Wiskunde & Informatica, Science Park 123, 1098 XG Amsterdam, Netherlands*
[b] *Vrije Universiteit, De Boelelaan 1081-1087, 1081 HV Amsterdam, Netherlands*

Abstract

In this study we identify spatial regions based on an empirical data set consisting of time-dependent origin-destination (OD) pairs. This OD data consists of electronic traces collected from smart phone data by Google in the Amsterdam metropolitan region and is aggregated by the volume of trips per hour at neighborhood level. In this study we cluster the pairs by space and time to gain insight in both aspects regarding travel characteristics. We show that spatially connected clusters appear when we use a performance metric called modularity on the OD data when directionality is incorporated.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Screen-lines; OD-matrices; Clustering; Modularity optimization; GPS traces; Travel behavior

1. Introduction

In a densely populated, compact city as Amsterdam where congestion in the city center is a main concern, it is of great importance to understand the commuting patterns of people. With the available sensor data nowadays, detailed information is available. This opens opportunities to analyze specific travel characteristics of for example districts within Amsterdam. It is of interest to analyze the typical traffic behavior in Amsterdam from both a time and spatial point of view. The aim of this research is to analyse whether travel patterns in Amsterdam can be aggregated in both space and time based on similar behavior. More specifically, can we determine clusters based on trips, and where and when these clusters arise. We are interested whether these clusters are spatially connected, and how strong is this connection.

In the literature various groups of clustering algorithms exists. A fairly complete review on this topic is written by Fortunato[4]. Clustering or graph partitioning is based on nodes that share common properties or behave in a similar manner. It is used to group nodes based on the graph topology only. A wide range of methods exist and various metrics are defined to determine the performance of the clustering result. A well-known metric to determine the

* Corresponding author. Tel.: +31 20 592 4301.
E-mail address: D.van.Leeuwen@cwi.nl

1877-0509
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2018.04.053

performance of the resulting clusters is called modularity, defined by Newman[8]. Modularity is a metric that computes the probability of trips occurring within a cluster compared to the probability of a trip belonging to one of the other clusters. Computing the partitioning of a graph that maximizes the modularity value is known to be NP hard. Various algorithms exist that are cluster based on optimizing the modularity value. In this paper we apply clustering by using one of these algorithms.

In a recent study, spatial clusters based on telephone calls have been examined by Blondel[1] and colleagues at the Université Catholique de Louvain. In their paper they developed a clustering algorithm which is based on maximizing the modularity value in an efficient manner. In a similar study this algorithm has been applied to telephone data in Great Britain by Ratti et al[10]. An interesting result that both of these paper show is the spatially connected clusters that appear naturally, while no spatial characteristics are considered in the algorithm. Both these datasets consists of a large amount of connections between the nodes of the network. This also applies in our dataset.

Another feature that is included in our dataset is directionality of the trips. In the original Louvain algorithm, analysis including directionality is not applied. However, the method is easily extensible to allow for directionality as is explained and elaborated by Dugué and Perez[3].

We will show that the Louvain method produces very good results to determine clusters based on origin destination data in the city of Amsterdam. The resulting clusters allow us to determine the answers to the following questions:
(1) Do people in Amsterdam tend to take more trips in their own neighborhood, and till what extend?
(2) What do the clusters that arise represent? Are they based on urban hierarchy, land use, or administrative regions?
(3) Do we observe specific time-dependent clusters that are not present during other periods?
Moreover, these results can be exploited to determine screen lines that support choosing sensor location to capture real-time traffic density in smart manner. A screen line is defined as a line segment consisting of one or multiple straight lines on a map. It is important to define determine good screen lines to determine where to place sensors that provide a good image of intra area traffic behavior. An overview of various screen line optimization methods was written by Yang et al[13]

2. Data analysis

2.1. Data specification

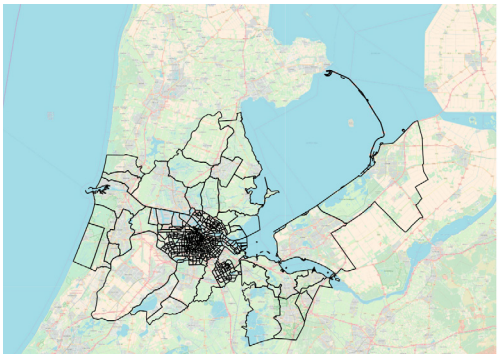
The travel data used for our analysis is based on travel movements registered by Google on android phones for the Amsterdam Metro region. This data spans a period of six months that starts 1 April 2016 until 30 September 2016. These trips are aggregated at neighborhood level for Amsterdam and the surrounding is aggregated at municipal level, both are grouped hourly. These neighborhoods of Amsterdam and surrounding muncipalities are based on the division made by the Netherlands Central Bureau of Statistics and splits the area in 512 small areas which are visualized for the Metro region in Figure 1a, and in more detail in Figure 1b. This results in a total of around 300 million data points, consisting of weights from each origin to each destination on a hourly basis. Due to privacy issues, the real intensity has not been disclosed, the intensity is given by a weight which represents a relative value. More specifically, all intensities have been divided by the largest hourly intensity over these 6 months, resulting in weight values between 0 and 1.

In Table 1 a summary of the weights observed in the data set is given based on the frequency. We observe that the total number of hours that contains a positive weight is close to 30%. As the data consists of all the destinations for each origin for every hour, we observe that during day time we have fully connected graphs. A large amount of the weights consist of small values, an overview is presented in Table 1.

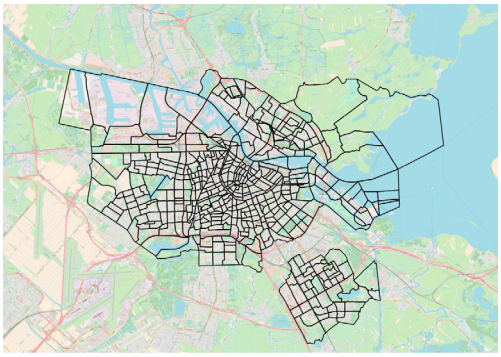
2.2. Filtering and preprocessing

In the analysis we focus on travel behavior in Amsterdam only. We analyze the behavior of people traveling within the city and from and to the city from the Metro region.

In Figure 2 the weekly pattern of trips within Amsterdam is visualized. It is interesting to observe that the rush hour is not so clearly present, and the number of trips in the weekend is nearly as large as during the weekdays. If we compare this to the trips from and to Amsterdam from the Metro region as shown in Figure 3, we do observe the rush



(a) The greater metropolitan area of Amsterdam.



(b) Detail of the municipality of Amsterdam.

Fig. 1: Division of the greater area of Amsterdam into municipalities surrounding Amsterdam and small neighborhoods within Amsterdam.

Weight	Perc. occurance	Perc. Total weight
0	71.62%	0%
0.000365764	17.67%	36.42%
0.000731529	6.66%	27.44%
0.001097293	2.49%	15.38%
0.001463058	0.93%	7.68%
> 0.001463058	0.63%	13.08%

Table 1: Table with the frequency for each weight.

hour clearly. In the morning a clear migration from the greater region of Amsterdam is observed to Amsterdam, and in the evening vice versa. In Figure 3b and 3d, the spatial spread of these trips is visualized. It is interesting to see that these two Figures do not show a similar pattern. It is interesting to observe that in Figure 3b, trips are homogeneously spread over Amsterdam, while in Figure 3d we see clear neighborhoods with an increased weight. The red areas in Figure 3d all contain large business districts which would be expected.

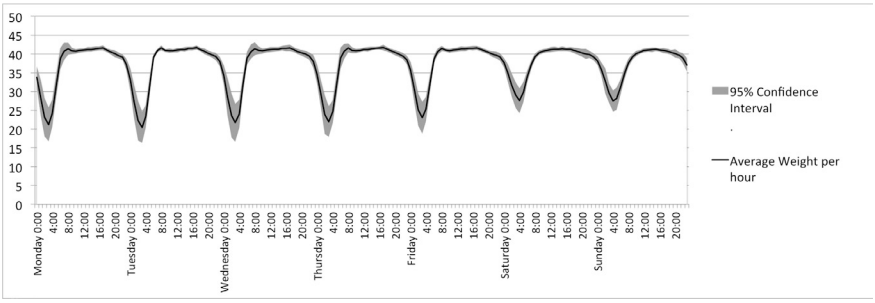
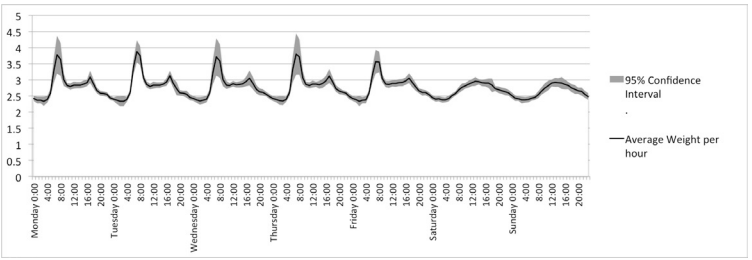


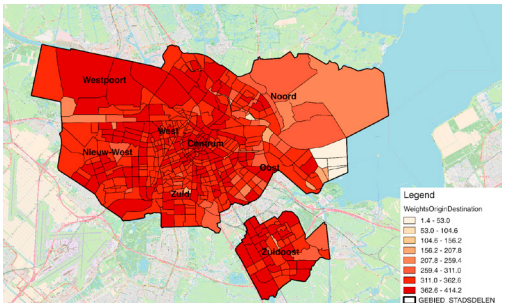
Fig. 2: Weekly pattern of weights per hour with a 95% confidence interval.

We continue our analysis by visualizing the trips for each neighborhood as an origin and as a destination in Figure 4. Here we can observe a similar pattern as we observed in Figures 3b and 3d. The destination figure shows a homogeneously spread pattern, while the origin Figure shows more variation between the areas. We observe that the certain parts of Amsterdam have more inflow than outflow. In Figure 4c the total inflow and outflow per neighborhood is shown. We can see that certain parts of Amsterdam have more inflow than outflow, except for the first 30 values which belong to the Metro region areas. This suggests that a transformation has been applied to sensor the data.

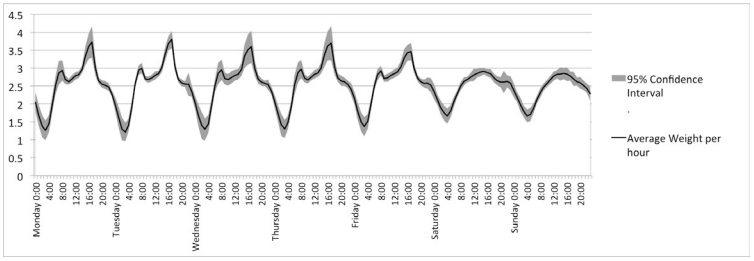
In order to restore the imbalance of the in and outflow we scale the rows of the OD matrix, such that the row and column sums become balanced. This is done by solving a system of equations where the OD weights are used as Markov chain weights[9]. The resulting stationary probability vector provides the scaling of rows such that the desired OD matrix property is restored. We use the origin weights as a reference and 'repair' the destination weights. In recent



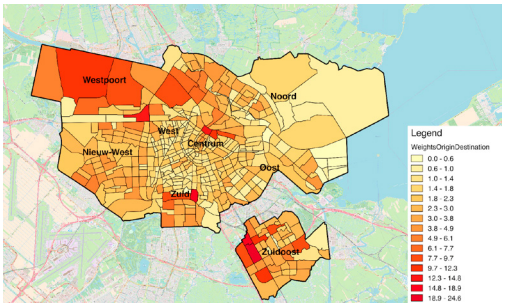
(a) Weekly pattern for Amsterdam as a destination.



(b) Spatial pattern for Amsterdam as a destination.



(c) Weekly pattern for Amsterdam as an origin.



(d) Spatial pattern for Amsterdam as an origin.

Fig. 3: Time and spatial intensity patterns to and from Amsterdam.

work by Tesselkin [12] this scaling method has been used to reconstruct the OD matrix from traffic flow observations on road segments.

In short, the computation consists of the following computation. We denote the OD matrix by an n by n matrix M. To restore the balance we simply have to solve

$$M\bar{x} = M^T\bar{e}, \tag{1}$$

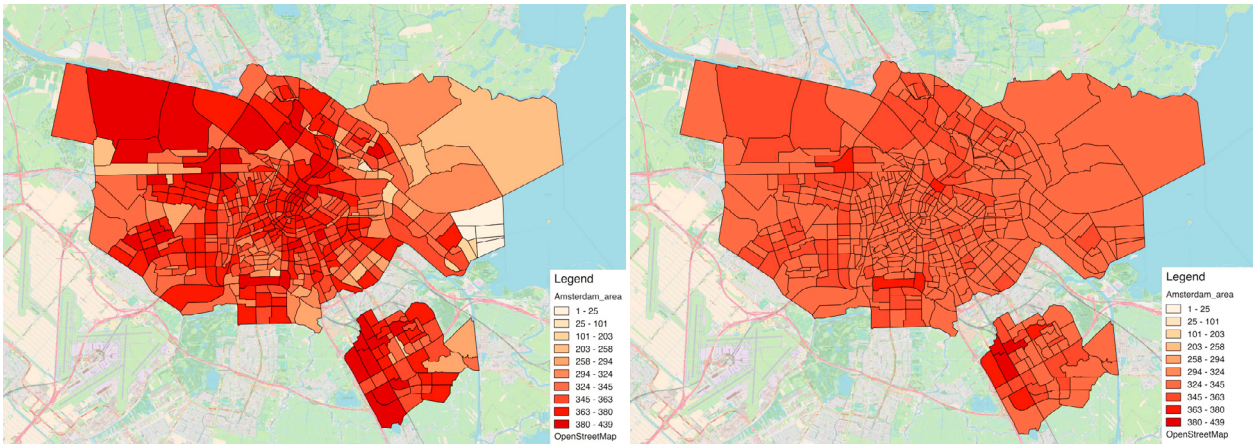
where $\bar{x}$ is the scaling vector and $\bar{e}$ is a vector of ones. The solution of $\bar{x}$ is than found by $\bar{x} = M^{-1}M^T\bar{e}$. The resulting scaling values are visualised in Figure 5a. It can be seen that a few areas have a scaling vector close to zeros, which is due to the small total outflow compared to the inflow of the specific neighbourhoods. These neighbourhoods are visualised in yellow in Figure 5b. We consider these areas as outliers. For analysis purposes these can be removed from the data, or the scaling factor can be used. In this paper, we ran all the clustering method with and without the scaling for comparison.

3. Aggregation of travel data

As we introduced in the beginning of the paper, the goal of our analysis is to discover travel characteristics within Amsterdam. More specifically, we want to determine whether district boundaries can be identified based on historical trips. By using clustering techniques we can determine whether clusters arise which are spatially connected, and whether these are similar to the districts that were defined by the municipality of Amsterdam. First, we explain the details of the metric modularity, which we use to evaluate the resulting clusters. We than explain a couple of heuristics based on optimizing this modularity metric. Finally, we show the results of these heuristic methods on various instances of the data.

3.1. Modularity optimization

Clustering based on optimization of the modularity value is a popular approach. Modularity is a well-known metric used for community detection in spatial travel data. The inspiration for this method of community detection is the

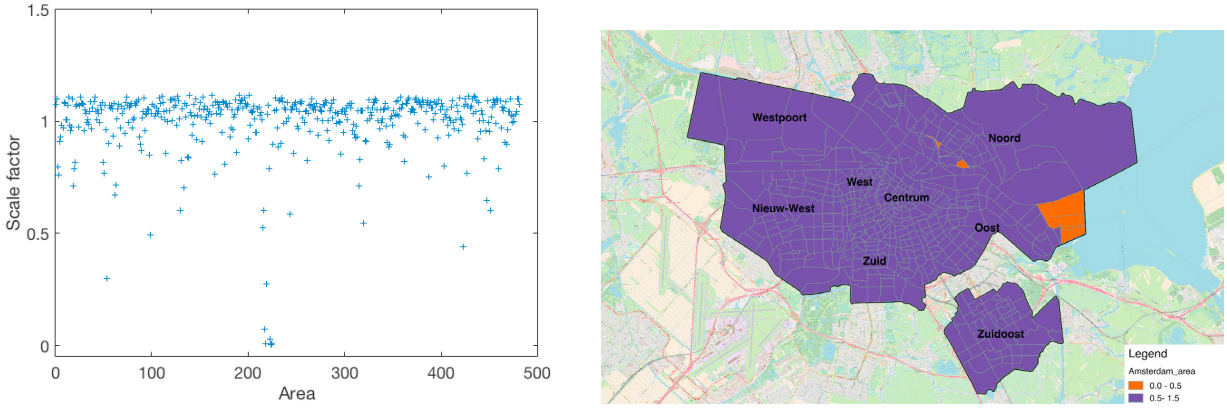


(a) Intensity per neighborhood as an origin.

(b) Intensity per neighborhood as a destination.

(c) Total inflow and outflow weight per neighborhood.

Fig. 4: Visualizations of the travel intensities within Amsterdam at each neighborhood.



(a)

(b)

Fig. 5: Scaling vector values to balance the inflow and outflow of each area.

optimization of modularity as the algorithm progresses. Modularity is a scale value between -1 and 1 and is defined by

$$Q = \frac{1}{2m} \sum_{vw} [A_{vw} - \frac{k_v k_w}{2m}] \delta(c_v, c_w), \tag{2}$$

where m is the total weight in the graph, A_{vw} is the weight from edge v to edge w, k_v defines the total weight of node v, and $\delta(c_v, c_w)$ determines whether node v and w share an edge. This formula measures the density of edges inside communities to edges outside communities, the value $A_{vw} - \frac{k_v k_w}{2m}$ defines the differences between the actual weight between nodes v and w and the expected weight given a random graph. Optimizing the modularity value theoretically, results in the best possible grouping of nodes of according to the inter and intra cluster trips for a given network, however going through all possible iterations of the nodes into groups is impractical so heuristic algorithms are used.

This computation can easily be extended to include directionality as was shown by Leicht and Newman in [7]. They show that instead of considering the total weight of the two edges, we now compare the in-degree of one edge with the out-degree of another edge the same computation can be applied. This results in the following equation

$$Q = \frac{1}{m} \sum_{vw} [A_{vw} - \frac{k_v^{in} k_w^{out}}{m}] \delta(c_v, c_w), \tag{3}$$

where A_{vw} now represents the total weight of an edge from v^{out} going to edge w^{in}.

3.2. Heuristic clustering technique

Many heuristic techniques exists for modularity optimization. A comparative study has been conducted by Lancichinetti [5]. Most of these heuristics are only implemented for undirected graphs, while our data consists of directed Origin-Destination pairs. In this paper we will not dive into all they clustering heuristics and their performances. We only focus on a method well-known for its computational efficiency, developed by Blondel et al [1] at the Université Catholique de Louvain. The Louvain method has been used to detect communities in geographical regions by means of telephone data. The result which captures our interest is the spatially connected clusters that were found, although no spatial characteristics were included in the algorithm.

3.3. Cluster results

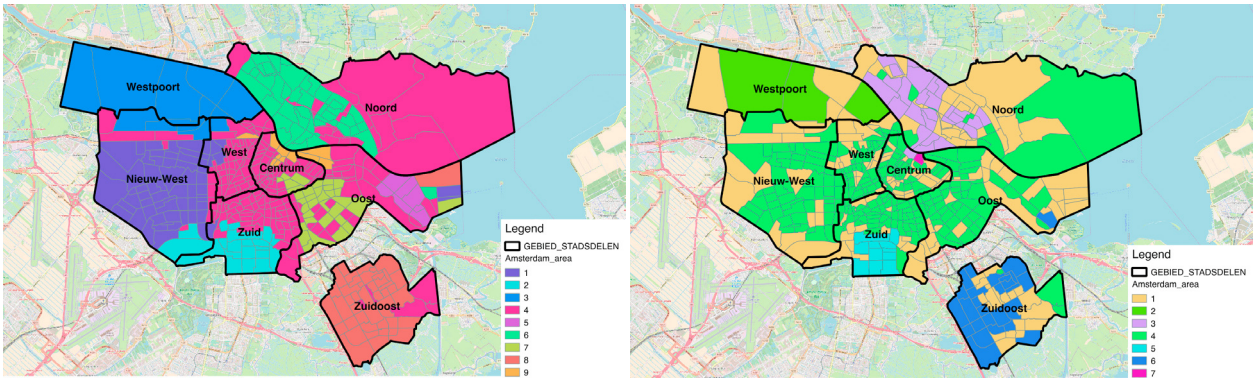
We have applied the Louvain clustering heuristic, and many others that have been proposed in the literature. All of them returned near to zero modularity values, which indicates that these clusters do not represent more than some random behavior. In Figure 6b shows a visualisation of the clustering output of the Louvain algorithm. It can be concluded that the resulting clusters show a non-coherent output. Although our data set does not consists of millions of nodes and edges, we do have a large #edges / #nodes value. The dataset consists of an almost fully connected graph, which is probably the reason that most clustering methods fail. Moreover, the directionality of the connections in the data was not included in most of these heuristics.

We continued the analysis by using an implementation of the directed Louvain method [11]. The output is visualized in Figure 6. As can be seen, the clusters that results from the Louvain method appear spatially close, although no spatial aspects are taken into account. It visually has a close resemblance with the districts of Amsterdam. However, the modularity value of the resulting clusters is 0.01, which is rather small.

The visually positive results of the directed Louvain method are interesting. We made a division of the data based on the trips during the week and the weekend and applied the Louvain clustering algorithm resulting in the clusters of Figure 7. Both show similar clusters, main differences are the cluster of 'Oost' and 'Westpoort' that appear only for the week data, and 'Amsterdam West' that pops up in the weekend data. The clusters at the outskirts of Amsterdam appear to be the most prominent. It is important to keep in mind that these visualisations are the result of a single output of the Louvain algorithm. The results vary due to a different initialisation of the algorithm, causing differences between each clustering partition.

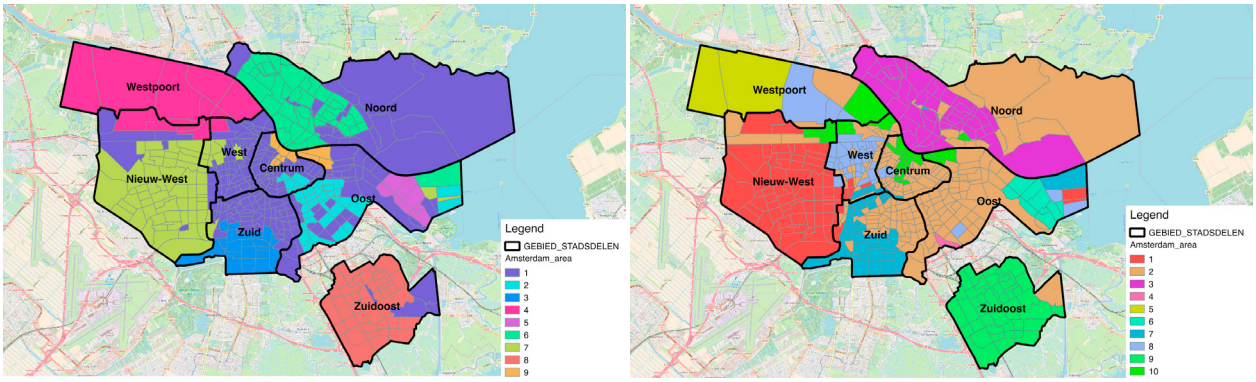
We compare the partitions by using a metric called Normalised Mutual Information (NMI) [2]. This metric is in the range of [0,1] and equals 1 if two partitions are identical. It is based on the entropy of each realisation, which is a value of the uncertainty in a partition. The mutual information gives the reduction in uncertainty by using the information of the first partition to estimate the second partition. In other words, it computes till what extend the realisations overlap. The NMI is defined as

$$NMI(Y, C) = \frac{2I(Y; C)}{H(C) + H(Y)}, \tag{4}$$



(a) Louvain directed. (b) Louvain undirected.

Fig. 6: Clustering with respect to destination for each district.



(a) Week. (b) Weekend.

Fig. 7: Clustering with respect to destination for each district using Louvain.

where $I(\cdot;\cdot)$ is the mutual information, $H(\cdot)$ the entropy value, C the class labels and Y the cluster labels. The value is normalised such that it corrects for differences in the total number of clusters obtained between realisations. This metric has been used to compute the quality of various clustering algorithms[6]. However, you need a benchmark graph of which you know the 'true' clustering result. In our case we have no 'true' clustering, but it allows us to compare the similarity between partitions. Table 2 shows the average similarities between the clustering realisations over the same dataset. We divided the data based on the total trips, column "Total", trips during the Week and Weekend. The resulting values show that the partitions are relatively similar when we take a large subset of the data. However, when we use the data per month and split between week and weekend, we can see that the resulting partitions are quite different between realisations. The data available is not sufficiently large to obtain consistent partitions. Therefore we will only consider the total weekly data to compare the realisations between months.

We want to determine whether we can observe whether there are large differences between months, and which ones give similar results. Therefore we compared the realisations of each month with all other months, and also compared it with the total data set partitions. The results are shown in Table 3. The last row compares the total data set with each month. We can see that the least similar months are August and September, however, we do not see large differences between the months.

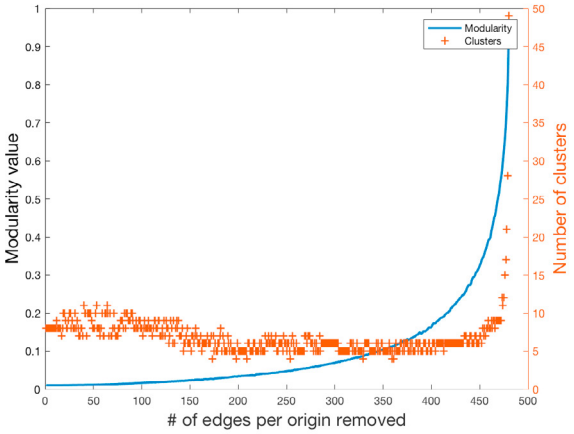
The small modularity value, combined with an almost fully connected network is a logical result. We therefore started a first analysis on edge removal to determine the impact of low weighted edges. We started with a simple analysis in which we removed the smallest x edges from each neighborhood, where $x \in 1, \ldots, n$ and n denotes the number of nodes of the network. In Figure 8a it can be seen that the modularity value increases when the number

Period	Total	April	Mei	June	July	August	September
Total	0.94	0.82	0.83	0.82	0.85	0.79	0.78
Week	0.94	0.84	0.88	0.91	0.83	0.85	0.87
Weekend	0.85	0.43	0.44	0.48	0.48	0.49	0.46

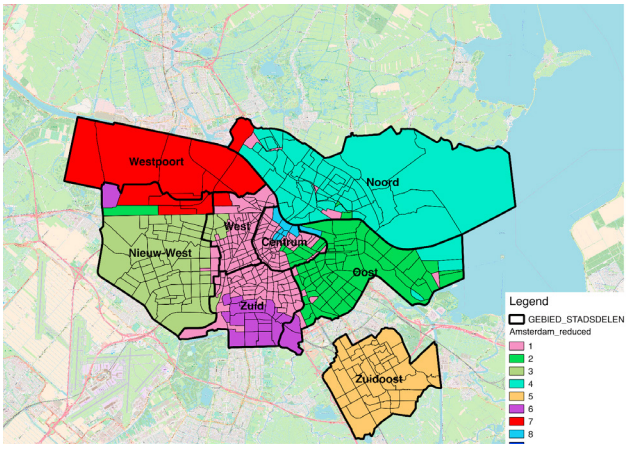
Table 2: Similarity (nmi) metric of the cluster realisations per dataset by using Louvain.

Period	April	Mei	June	July	August	September	Total
April	0.85	0.75	0.74	0.77	0.73	0.73	0.82
May	0.75	0.86	0.76	0.76	0.73	0.73	0.83
June	0.74	0.76	0.91	0.75	0.72	0.72	0.82
July	0.77	0.76	0.75	0.87	0.77	0.75	0.84
August	0.73	0.73	0.72	0.77	0.86	0.72	0.79
September	0.73	0.73	0.72	0.75	0.72	0.87	0.78
Total	0.82	0.83	0.82	0.84	0.79	0.78	0.94

Table 3: Values of the mutual information of the clustering results between different time slices in the data.



(a) Modularity value for increasing edge removal.



(b) Clusters analysis by filtering the 10% smallest weights of each area.

Fig. 8: Modularity analysis by edge removal.

of small edges removed increase, which is to be expected. More interestingly is that the number of clusters found remains relatively constant until almost all values are removed. In Figure 8b the clusters found when 10% of the smallest weights were removed is visualized. As can be seen these cluster are representing the regional boundaries even more closely than the complete set. This suggests that although the trips within Amsterdam are very well spread, trips within regional boundaries have higher weights in almost every district. Only part of the Center, West and Zuid remain connected as one cluster.

4. Conclusion

In this paper we analyzed travel behavior in Amsterdam based on Origin-Destination data. We analyzed both the spatial variation as well as the time-dependent variation of trips. Based on this analysis we transformed the data such that the total inflow equals the total outflow for each area. We proceeded our analysis by using clustering techniques based on modularity optimization to separate regions based on internal travel behavior.

The weekly pattern and spatial plots confirm expected behavior, such as the morning and evening commute. However, from this analysis we also discovered a gap between the total inflow and outflow. As there is no logical explanation for this behavior, we assume that this occurs due to some transformation to sensor the data. In order to properly analyze the data we restored this imbalance, and obtained scaling values for each neighborhood. This revealed a couple of outliers in the data. Especially in the east of Amsterdam a few neighborhoods which are mostly situated in the water showed a large difference between the total inflow and outflow value.

Our results show that we were able to identify clusters when the directionality is taken into account. These clusters happen to be very similar to the regional districts defined in Amsterdam. Especially at the outskirts of Amsterdam we can clearly identify clusters. The city center is represented by one large cluster, together with parts of the east of Amsterdam. When the method is separated into monthly periods, and a division between the weekend en weekday trips is introduced, we observe more prominent clusters in the weekend. While during the week these clusters are sometimes hard to obtain. This suggests that commuting trips are more spread around the city, while leisure trips are more often in people's own districts. These are results that would be expected. Lastly we observe that the trips taken from the metro region of Amsterdam are largely commuting trips. There are three areas that show a high density of trips, each of them contains large business districts. Finally, we analyzed the results when part of the data is removed. This revealed that a lot of small weight edges can be removed without losing the spatially obtained clusters. We can conclude from the above analysis that trips in Amsterdam are quite homogeneously spread in the city. However, we do observe clustering, although not so prominently. This analysis should be extended by including dynamic time-window clustering. Moreover, filtering the commuting trips from the regular trips can give additional insights into the travel behavior at each area for leisure.

References

1. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.
2. Thomas M Cover and Joy A Thomas. Elements of information theory. pages 13–55, 2012.
3. Nicolas Dugué and Anthony Perez. *Directed Louvain: maximizing modularity in directed networks*. PhD thesis, Université d'Orléans, 2015.
4. Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.
5. Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.
6. Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.
7. Elizabeth A Leicht and Mark EJ Newman. Community structure in directed networks. *Physical review letters*, 100(11):118703, 2008.
8. Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.
9. J. R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997.
10. Carlo Ratti, Stanislav Sobolevsky, Francesco Calabrese, Clio Andris, Jonathan Reades, Mauro Martino, Rob Claxton, and Steven H Strogatz. Redrawing the map of great britain from a network of human interactions. *PloS one*, 5(12):e14248, 2010.
11. Antoine Scherrer. Matlab louvain implementation. online, 2008.
12. Alexander Tesselkin and Valery Khabarov. Estimation of origin-destination matrices based on markov chains. *Procedia Engineering*, 178:107–116, 2017.
13. Hai Yang, Chao Yang, and Liping Gan. Models and algorithms for the screen line-based traffic-counting location problems. *Computers & Operations Research*, 33(3):836–858, 2006.