



# Network partitioning on time-dependent origin-destination electronic trace data

Daphne van Leeuwen<sup>1</sup> · Joost W. Bosman<sup>2</sup> · Elenna R. Dugundji<sup>3</sup>

Received: 23 July 2018 / Accepted: 15 February 2019  
© The Author(s) 2019

## Abstract

In this study, we identify spatial regions based on an empirical data set consisting of time-dependent origin-destination (OD) pairs. These OD pairs consist of electronic traces collected from smartphone data by Google in the Amsterdam metropolitan region and is aggregated by the volume of trips per hour at neighbourhood level. By means of community detection, we examine the structure of this empirical data set in terms of connectedness. We show that we can distinguish spatially connected regions when we use a performance metric called *modularity* and the trip directionality is incorporated. From this, we proceed to analyse variations in the partitions that arise due to the non-optimal greedy optimisation method. We use a method known as *ensemble learning* to combine these variations by means of the overlap in community partitions. Ultimately, the combined partition leads to a more consistent result when evaluated again, compared to the individual partitions. Analysis of the partitions gives insights with respect to connectivity and spatial travel patterns, thereby supporting policy makers in their decisions for future infra structural adjustments.

## 1 Introduction

In a densely populated, compact city as Amsterdam it is of great importance to understand the travel patterns of people, as congestion in the city centre is a main concern. With the rise of ubiquitous sensor data, detailed information with respect to mobility is available. Not only can we analyse the infrastructure performance more accurately, it also opens up new opportunities to for estimation, integration and validation of existing models.

For this study, we had access to origin-destination (OD) intensities for the metro region of Amsterdam. These ODs represent neighbourhoods within Amsterdam, and municipalities for the metro region of Amsterdam. The OD

intensities are based on electronic trace data collected from smartphone data by Google. These traces are aggregated at neighbourhood and municipal level by their volume of trips on an hourly basis for a 6-month period of time. We obtained these OD intensities by a project from Google called ‘Better Cities’ in which they provided this data based on the research proposals they received.

The aim of this research is to analyse whether travel patterns in Amsterdam can be aggregated into high-level patterns to detect flow trends in both space and time. In the literature, this is called community detection, where the high-level patterns are identified as communities. The results of such an approach can be exploited to analyse major flow patterns between areas based on the obtained communities. Moreover, the obtained communities can be used to support practitioners with strategic decisions, for example to identify or justify the expansion of public transport between specific areas.

We apply clustering to identify communities based on historical travel data. Clustering or graph partitioning is based on nodes that share common properties or behave in a similar manner. In this context, community detection is used to group nodes based on the edge properties only. We thereby want to identify the typical traffic behaviour in Amsterdam from both a temporal and spatial point of view.

In the literature, a wide range of community detections algorithms exist, as well as the number of metrics to evaluate the partition quality of the detection algorithms. A fairly

✉ Daphne van Leeuwen  
daphne.vanleeuwen@gmail.com

Joost W. Bosman  
joost.bosman@tno.nl

Elenna R. Dugundji  
e.r.dugundji@vu.nl

<sup>1</sup> Centrum Wiskunde & Informatica, Science Park 123,  
1098 XG, Amsterdam, Netherlands

<sup>2</sup> TNO, The Hague, The Netherlands

<sup>3</sup> VU Amsterdam Faculty of Science, De Boelelaan 1085,  
1081 HV, Amsterdam, The Netherlands

complete review of this topic is given in [12]. By far, the most popular metric to determine the performance of the resulting clusters is called *modularity*, introduced by [20]. Modularity is a metric to measure the strength of a network partitioned into communities based on the intra-inter community edge weight, i.e., the more weight captured within each community compared to the weight between communities, the stronger the connection and the larger the modularity value. The problem of finding the partitioning of a graph with the maximum modularity value is known to be NP-complete [7]. Various heuristics exist to optimise the modularity value. An overview of these methods can be found in [12, Chapter 6].

In a recent study, spatial clusters based on telephone calls have been examined by Blondel et al. [6], who developed an efficient heuristic procedure to find a partition of the network that maximises the modularity known as the Louvain algorithm. In a similar study, this algorithm has been applied to telephone data in Great Britain by Ratti et al. [23]. For this study, a geographical area is partitioned into small regions. These regions are translated to a graph where the regions are represented by nodes, and the intensity of phone calls by edges. In both of these papers, the resulting communities obtained from the clustering procedure are spatially connected, while no spatial characteristics are considered in the algorithm. Moreover, both these datasets consist of a large number of connections between the nodes of the network. This algorithm is of interest to us, as the geographical component and densely connectedness both apply to our dataset.

Another feature that is included in our dataset is directionality of the trips. In the original Louvain algorithm [5], analysis including directionality is not applied. However, the method is easily extendable to allow for directionality, as is explained in [10]. We will show that the Louvain

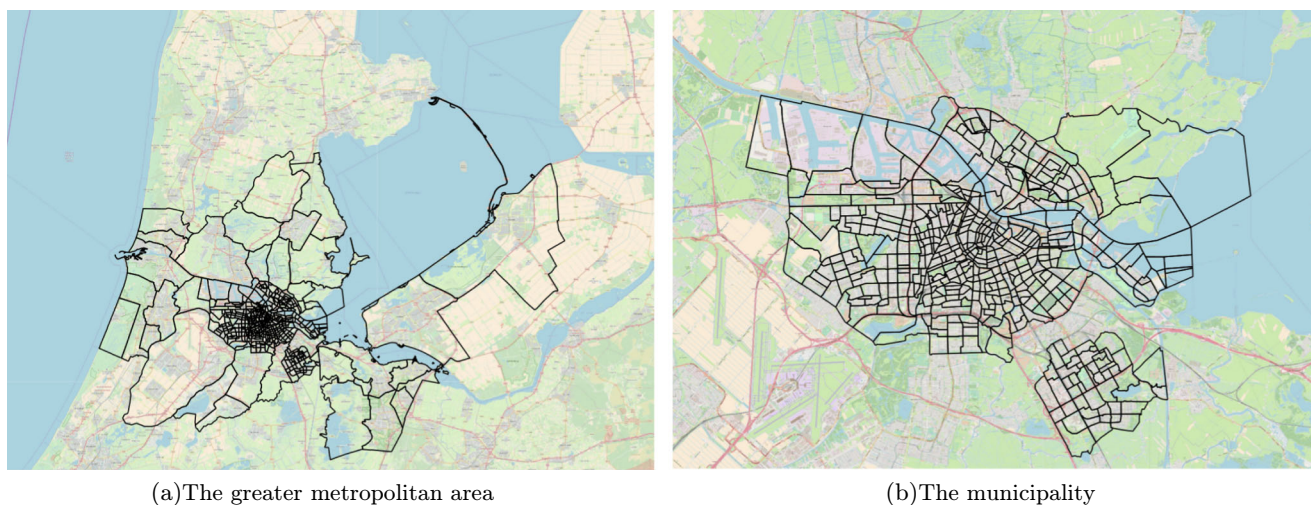
method produces very good results to determine clusters based on origin-destination pairs in the city of Amsterdam when directionality is included.

We extend the analysis for different time slices of the data (i.e., weekday, months, etcetera), and show that this results in variations in the obtained communities. However, the comparison is not straightforward as the Louvain method generates variations between each run for the same time slice as well. However, the efficiency of the Louvain heuristic with minimal computational effort allows for a more elaborative analysis on the variations between partitions of a network. To this end, we use a technique known as *ensemble learning* to obtain a more consistent partition, i.e., less variation between partitions resulting from the same algorithmic procedure. In [13], they explain this procedure applied to graph partitioning. A more consistent partition of the community structure for a specific time slice allows for a better comparison of partitions of other time slices.

The remainder of this paper is organised as follows. In Section 2, we give a detailed description of the data and specify the filter and preprocessing steps used for the model, which we introduce in Section 3. The preliminary results are then given in Section 4. From there, we proceed to characterise the obtained communities in terms of connectivity strength in Section 5 and consistency in Section 6. We conclude in Section 7.

## 2 Data analysis

Before we move to the step of clustering, we first give a more detailed description of the data. By applying some preliminary aggregation steps, we obtain initial insights into the value of the data. Some of these confirm assumptions,



**Fig. 1** Neighbourhood and district level division of the greater metropolitan area of Amsterdam

**Table 1** Example data of the Google data set

| Origin | Destination | Start interval (sec) | End interval (sec) | Weight                 |
|--------|-------------|----------------------|--------------------|------------------------|
| 438    | 310         | 1468918800           | 1468922400         | 0.00036576445563696325 |
| 438    | 310         | 1473652800           | 1473656400         | 0.00036576445563696325 |
| 438    | 310         | 1471942800           | 1471946400         | 0.0007315289112739265  |
| 438    | 2           | 1470542400           | 1470546000         | 0.0007315289112739265  |
| 438    | 403         | 1470542400           | 1470546000         | 0.00036576445563696325 |

such as daily fluctuations in travel density, while they also reveal some deficiencies of the data. Moreover, these preliminary results illustrate the motivation to direct the research of this paper to the use of clustering methods.

### 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 was obtained by an inquiry which was sent to Google to analyse aggregated travel behaviour. As a result, we received aggregated trip intensities at neighbourhood level for Amsterdam and at municipality level for the surrounding of Amsterdam, both are grouped hourly. The data set spans a period of 6 months that starts 1 April 2016 until 30 September 2016. The aggregation is based on the division made by Statistics Netherlands found in [1], who split the area into 512 small pieces as visualised in Fig. 1a, and in more detail in Fig. 1b. This division results in more than 300 million data points, consisting of weights from each origin to each destination on an 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 example of the data is given. In the columns, the origin and destination neighbourhood is indicated by a number. The time interval is given in seconds from UNIX time and consist of hourly intervals.

**Table 2** Frequency values for each weight in percentage of occurrence and total density

| Weight        | % Occurrence | % 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%         |

In Table 2, 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 weights larger than 0 is close to 30%. As the data consists of all destinations for each origin for every hour, we observe fully connected graphs during most peak hour periods. A large number of the weights consist of small values, an overview is presented in Table 2.

### 2.2 Filtering and preprocessing

For the clustering procedure, we restrict ourselves to the travel characteristics within Amsterdam. In this section, we analyse the behaviour of people travelling within the city, and the travelling behaviour from and to the city from the metro region (defined in Fig. 1a), to grasp the main traffic characteristics and identify deviating patterns.

In Fig. 2, the weekly pattern of trips within Amsterdam is visualised. As can be seen, the rush hour is not so clearly present, and the number of trips in the weekend is nearly as large as during the weekdays. Of course, this data contains not only car travel movements, but also walking and cycling which could explain the intensity of trips throughout the day. The rush hour of trips between Amsterdam and the metro region area visible in Fig. 3. In the morning, a clear migration from the greater region of Amsterdam is observed to the city of Amsterdam, and in the evening vice versa. In Fig. 4, the spatial spread of these trips is visualised. The dark red areas in Fig. 4b all contain large business districts, which could be expected. However, we observe that Figs. 4a and b do not show a similar pattern. In Fig. 4a trips are homogeneously spread over Amsterdam, whereas in Fig. 4b larger variations in weight between neighbourhoods is observed. A more detailed analysis on this aspect will be discussed below.

In Fig. 5 of the total trip, weight for each neighbourhood as an origin and as a destination for trips within Amsterdam is visualised. We again observe a similar pattern as in Fig. 3. The destination figure shows a homogeneously spread pattern, while the origin figure shows more variation between the areas. This suggests that certain parts of Amsterdam have more inflow than outflow over a large period of time, which does not make sense considering that these trip intensity are the sum of a half year period. In

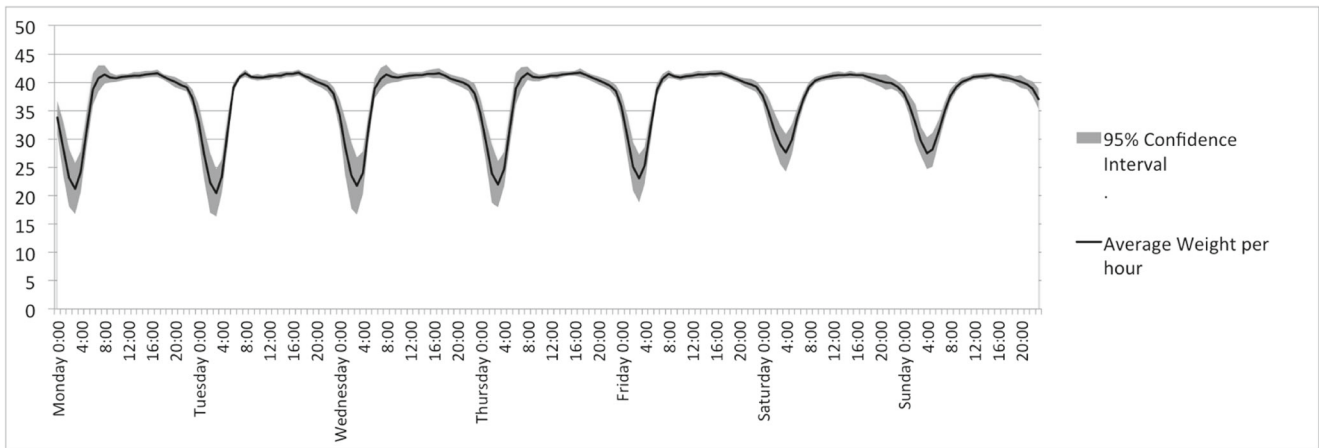
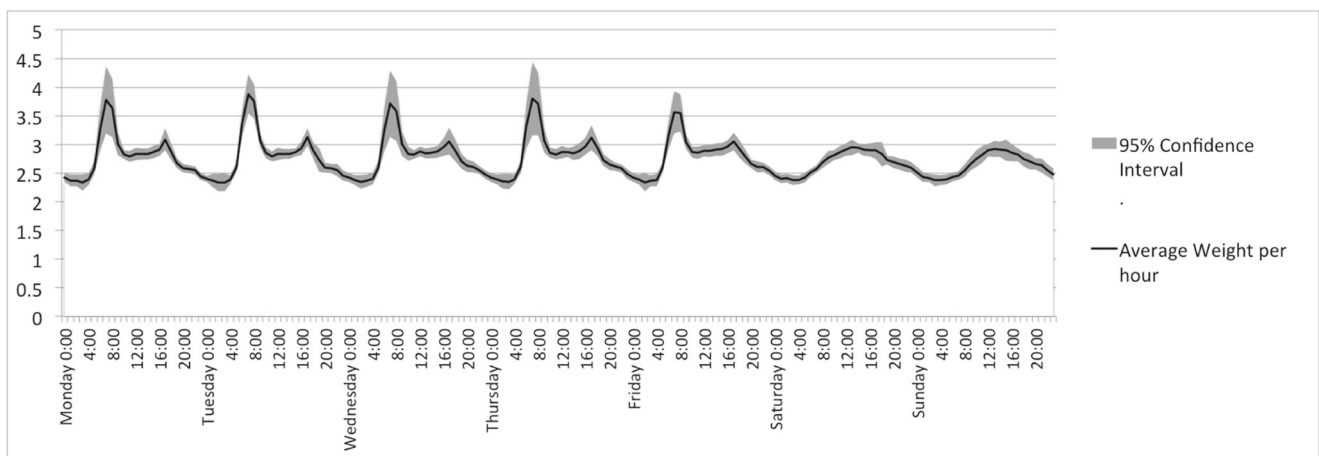


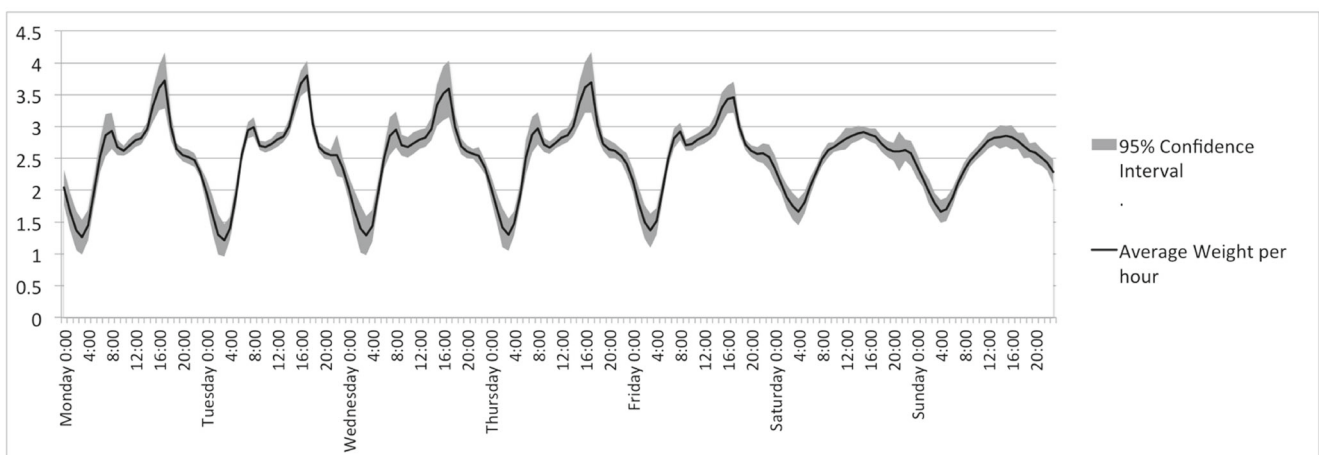
Fig. 2 Weekly pattern of weights per hour with a 95% confidence interval

Fig. 5c, the total inflow and outflow per neighbourhood are visualised. It shows that certain parts of Amsterdam have larger inflow than outflow, except for the first 30 values

which belong to the metro region areas. These observations suggests that a transformation has been applied to censor the data.



(a) Weekly pattern for Amsterdam as a destination



(b) Weekly pattern for Amsterdam as an origin

Fig. 3 Time intensity pattern to and from Amsterdam

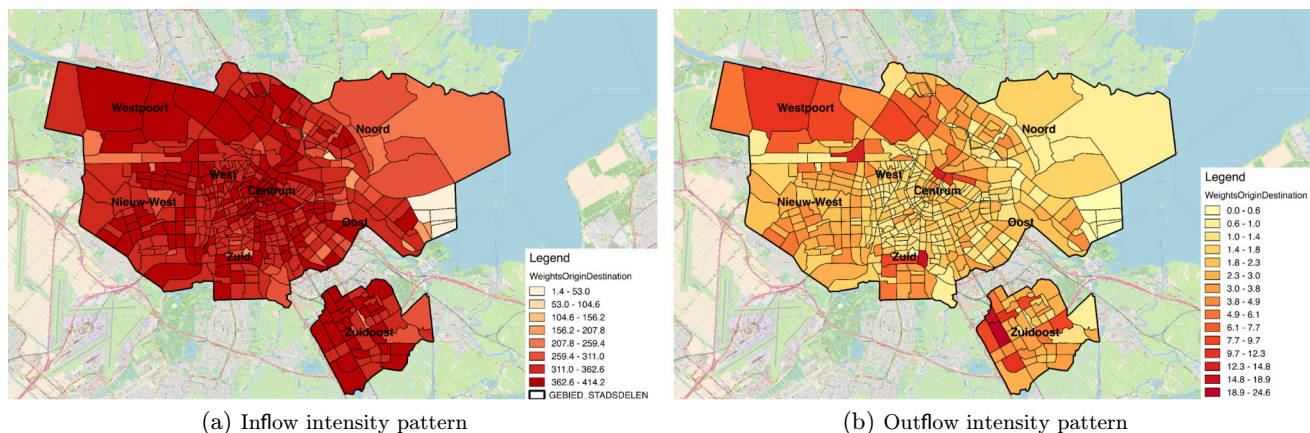
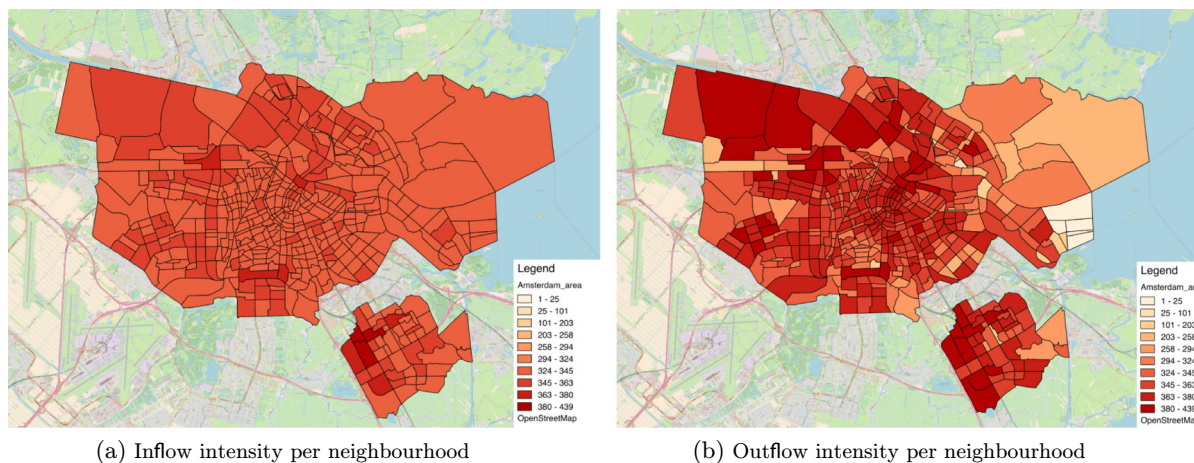


Fig. 4 Spatial intensity pattern of Amsterdam of trips from and to the surrounding metro region of Amsterdam spanning the 6-month period.

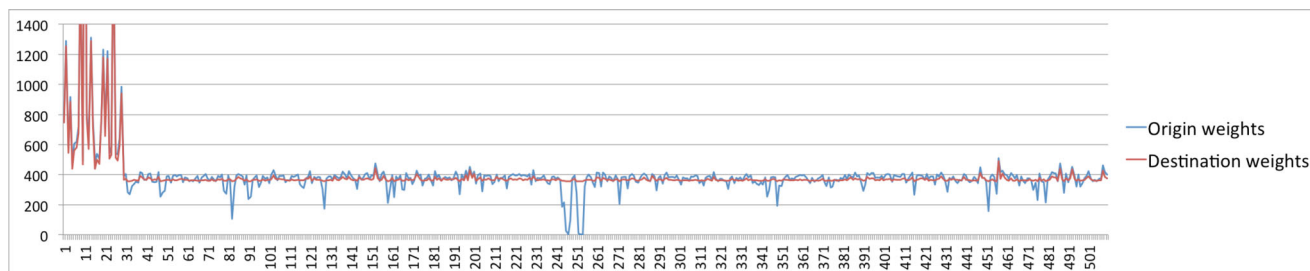
In order to restore the disbalance of the in- and outflow, we rescale the rows of the OD matrix such that the row and column sums become equal. This is done by solving a system of equations where the OD weights are used as Markov chain weights [21]. The resulting stationary probability vector provides the scaling of rows such that the OD matrix disbalance is restored. We use the origin weights as a reference and ‘repair’ the destination weights. In recent

work by Tesselkin [26], 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 steps. We denote the OD matrix by an  $n$  by  $n$  matrix  $W$ , where  $n$  denotes the total number of neighbourhoods, and  $W_{i,j}$  denotes the intensity of trips from neighbourhood  $i$  to neighbourhood  $j$ , for  $i, j = 1, \dots, n$ . To restore the

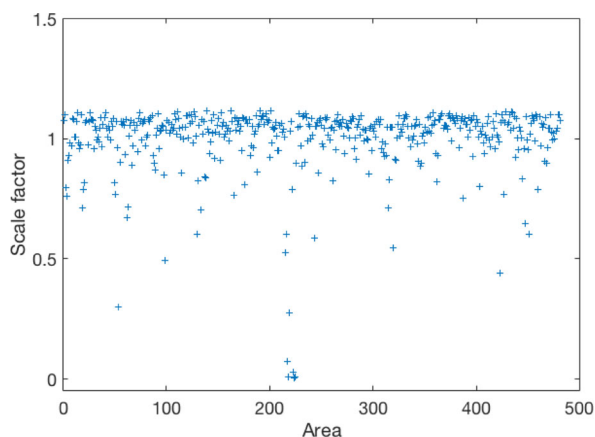


(a) Inflow intensity per neighbourhood (b) Outflow intensity per neighbourhood

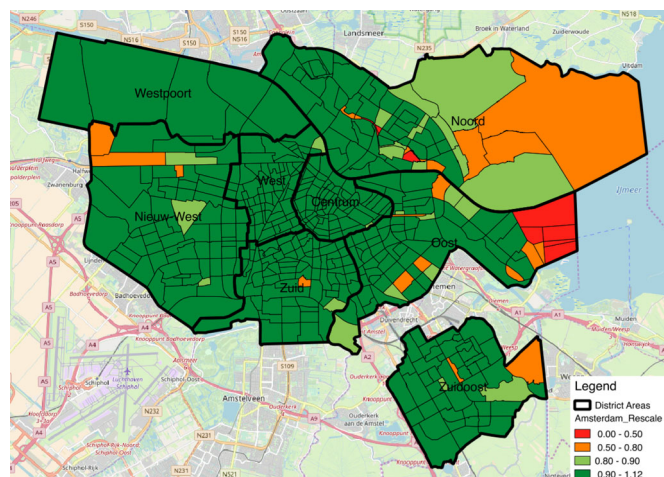


(c) Total inflow and outflow intensity per neighbourhood of trips within Amsterdam

Fig. 5 Visualisations of the travel intensities within Amsterdam at each neighbourhood spanning the 6-month period



(a) Scaling values of numbered neighbourhoods



(b) Geographical visualisation of scaling values

**Fig. 6** Rescaling values of each neighbourhood, visualising the imbalance of the data between inflow and outflow, where a value of 1 represents no imbalance

balance, we have to solve the linear set of equations as follows:

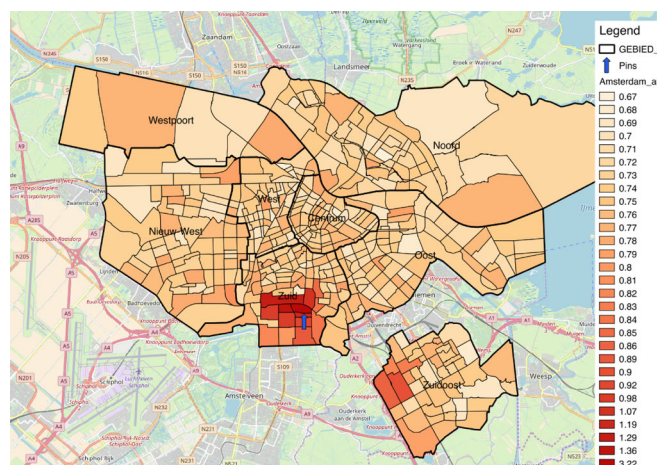
$$W\bar{x} = W^T\bar{e}, \tag{1}$$

where  $\bar{x}$  is the scaling vector and  $\bar{e}$  is a vector of ones.

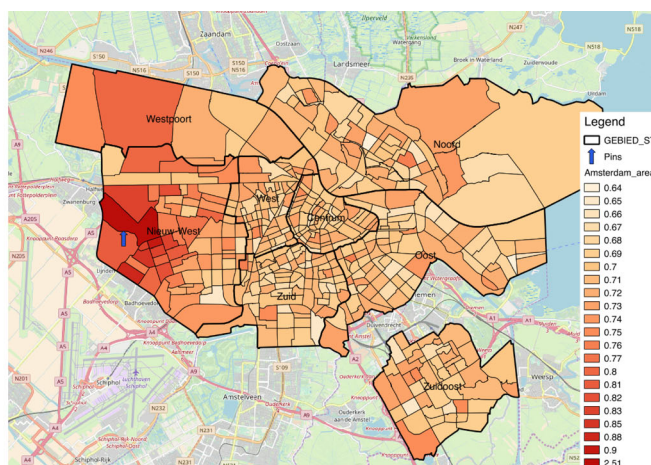
In a Markov setting, the vector  $x$  may be interpreted as the stationary vector that balances inflow and outflow for each node in the OD matrix. The left-hand side of Eq. 1 resembles the total outflow corresponding to vector  $x$ . The right-hand side corresponds to the total inflow for each node in the OD matrix by taking the sum over all columns of  $W$ . The solution of  $\bar{x}$  is then obtained by premultiplying (1) with the pseudo-inverse  $W^{-1}$  such that  $\bar{x} = W^{-1}W^T\bar{e}$ . The resulting scaling values are visualised in Fig. 6a. It can be seen that a few areas have a scaling vector close to zero, which is due to the small total outflow

compared to the inflow of the specific neighbourhoods. These neighbourhoods are visualised in yellow in Fig. 6b. 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 do not adjust or remove neighbourhoods to keep the analysis as clean as possible. Instead, we use the outlier analysis to explain behaviour caused by these deviations.

To give an indication of the travel characteristics per neighbourhood, we visualise the inflow intensity of two neighbourhoods in Fig. 7. We choose the inflow pattern, as the outflow shows a homogeneous pattern as observed in Fig. 5a. From both Figs. 7a and b, it is observed that travel intensities are larger around the area specified. This suggests that spatially connected communities might arise when neighbourhoods are clustered based on trip intensities.



(a) Neighbourhood in 'Zuid' pinpointed in blue



(b) Neighbourhood in 'Nieuw-West' pinpointed in blue

**Fig. 7** Visualisation of travel flow from a single destination

In this section, we analysed the travel patterns captured in the historical OD data from a spatial and temporal perspective. In general, we can conclude that these patterns match our expectations of travel intensity behaviour, except for the observed disbalance between in- and outflow. We investigated the extensiveness of this disbalance by equalising the in- and outflow. This led to the conclusion that, although we suspect an unexplained transformation on the data, we are confident that this transformation has minor impact on our results. Therefore, in the remainder of this paper, we perform our analysis on the original data set.

### 3 Model description

Before we provide all the details of our modelling approach, we first give a short introduction to network analysis and why we apply methods from social network analysis.

Network analysis in transportation systems is mainly concerned with the spatial and temporal nature of movements across the infrastructure and the infrastructural topology. Describing the network in terms of nodes and their linkage to each other, measures such as *accessibility* and *connectivity* can be extracted. Including the flow of movements across the infrastructure can be used to analyse the network performance. However, this is not an easy task as such detailed information is often not available.

A major area of research is concerned with the estimation of path flows, route choice decisions, and mode choice. A model incorporating the decisions and estimations into a framework is known as *discrete choice modelling* [4]. Currently, discrete choice models include an elaborate specification of dynamics and other elements. However, social influences are in general not taken into account in such models, which was first mentioned in [11]. In this case, models from social network analysis come at hand. In recent studies, this aspect is considered to be very important [27]. An overview of the current research is given in [19].

In this study, we apply methods from social network analysis to discover social interactions. Specifically, we use community detection. The aim of community detection is to divide the graph into components based on the topological information of the graph only [12]. These communities consist of groups of nodes that have a stronger connection to each other than to members of another community. Community detection algorithms give insight into the geographical connection and separation by grouping the regions into communities.

In the remainder of this section, we explain step by step the theory and procedure to use community detection in the OD dataset. First, we explain the transformation of the OD trip matrix to a connected graph. Then, we introduce the evaluation metric, denoted as *modularity*, to determine

the quality of a network partitioned into communities. We give an explanation of several models that heuristically optimise this modularity metric. Finally, we show the results of the heuristic method of our choice in which we obtain interesting results for various subsets of the data.

#### 3.1 Network description

We represent the OD trip matrix  $W$  in terms of a directed weighted graph  $G(\mathcal{V}, \mathcal{E})$ , where each node  $i \in \mathcal{V} = \{1, \dots, n\}$  represents a neighbourhood and each edge  $(i, j) \in \mathcal{E} \subset \mathcal{V}^2$  represents an OD pair. Each edge has a weight that corresponds to the travel intensity across the respective OD pair, denoted by  $w_{i,j} \geq 0$ , where  $i, j \in \mathcal{V}$ . We partition the graph into  $C$  communities, where, for each node  $i$  mapping index function  $c_i = k$ , for  $k = 1, \dots, C$  to its corresponding community. We define  $\mathcal{V}_k := \{i \in \mathcal{V} : c_i = k\}$  as the set of nodes that belong to community  $k$ . Moreover, we define  $\mathcal{C}_i := \{k \in \mathcal{V} : c_i = c_k\}$  as the set of nodes that belong to the same community as node  $i$ . The graph is initialised by either assigning each node to a unique community ( $C = n, c_i = i$ ), or by assigning all nodes to one community ( $C = 1, c_i = 1$ ).

#### 3.2 Modularity metric

*Modularity* is a well-known metric to determine the quality of a graph partitioned into communities. It is a measure of strength of the partition of the network into communities and is defined by a scalar value  $Q \in [-1, 1]$ . In the literature, the modularity value is often computed for undirected graphs. Therefore, we first present the undirected version before we explain the directed one. The modularity value  $Q$  for an undirected graph is defined by the following:

$$Q = \frac{1}{2m} \sum_{k=1}^C \sum_{i,j \in \mathcal{V}_k} \left[ w_{i,j} - \frac{w_i w_j}{2m} \right], \tag{2}$$

where  $m = \frac{1}{2} \sum_{i,j \in \mathcal{V}} w_{i,j}$  is the total weight in the graph, and  $w_i = \sum_{j \in \mathcal{V}} w_{i,j}$  defines the total edge weight attached to node  $i$ . This formula measures the density of edges inside communities to edges outside communities, the value  $w_{i,j} - \frac{w_i w_j}{2m}$  defines the differences between the actual weight between nodes  $i$  and  $j$  and the average node degree weight of  $i$  and  $j$ . Maximising 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.

The modularity metric of (2) can easily be extended to include directionality as was shown by Leicht and Newman [18]. They show that the total weight connected to these two edges should be split into the total in-degree weight of

one edge and the total out-degree weight of the other edge. Moreover, we specify the total weight by  $m_d = \sum_{i,j \in \mathcal{V}} w_{i,j}$  instead of  $2m$  as we now count each edge weight only once. This results in the following equation are as follows:

$$Q_d = \frac{1}{m_d} \sum_{k=1}^C \sum_{i,j \in \mathcal{V}_k} \left[ w_{i,j} - \frac{w_i^{\text{in}} w_j^{\text{out}}}{m_d} \right], \tag{3}$$

where  $w_i^{\text{in}} := \sum_{j \in \mathcal{V}} w_{j,i}$ , and  $w_i^{\text{out}} := \sum_{j \in \mathcal{V}} w_{i,j}$ .

### 3.3 Heuristic clustering technique

Clustering based on optimisation of the modularity value is a popular approach [12]. Many heuristic techniques exist for modularity optimisation. A comparative study has been conducted by in [15]. Most of these heuristics are only implemented for undirected graphs, while our data consists of directed OD pairs.

For this research, we will not dive into all the clustering heuristics and their performances. Instead, we only focus on a method well-known for its computational efficiency, developed in [6], throughout referred to as the *Louvain method*. This method was first 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. Moreover, this algorithm has shown to outperform many other heuristic methods for benchmark graphs. It has been ranked as second-best heuristic algorithm [15]. The infomap algorithm by Rosvall and Bergstrom [24], which is based on compression, has been ranked as first. Later on, we shortly mention its performance on our dataset.

We now briefly explain the partitioning procedure of the Louvain algorithm; a more detailed description is given in [6]. This algorithm can be classified as a greedy hierarchical approach for modularity optimisation and is known for its computational efficiency. The algorithm consists of a two-step procedure which is iterated until the modularity value is no longer improved. The first step is the ‘greedy’ assignment of nodes to communities, and the second step contains the hierarchical component, where the obtained communities are combined.

**Initialisation** The graph is initialised by a partition into singletons, meaning that each node represents a community.

**Step 1:** A loop initiates that runs through all the nodes in a random order. For each node  $i \in \mathcal{V}$ , the neighbouring nodes are identified, i.e.,  $w_{i,j} > 0$ . For each neighbouring node  $j$ , the modularity gain is computed when node  $i$  is added to the community  $c_j$  of neighbouring node  $j$ . The node  $i$  is then added to the neighbouring node  $j$ ’s community that creates the largest

positive increase in modularity, computed by (5). The first loop is re-initiated until the modularity gain is no longer improved.

**Step 2:** All nodes that belong to the same community are combined into one node representing the community. This means that the total weight to an external node is combined from all the nodes within the community, and the total weight of nodes within the community is summed, representing the total weight from the community to itself.

**Stopping criterium:** Repeat steps 1 and 2 until the final communities between the current and previous iteration are equal.

To speed up the above computation, we focus on the change in modularity when node  $i$  is moved to the community of node  $j$ , rather than recomputing the modularity by (2). For given modularity  $Q$ , the new modularity becomes  $Q' = Q + \Delta_Q(i, j)$ , where  $\Delta_Q(i, j)$  is defined by the following:

$$\begin{aligned} \Delta_Q(i, j) &= -\frac{1}{2m} \left[ \sum_{k \in \mathcal{C}_i \setminus i} (w_{i,k} + w_{k,i} - \frac{w_i w_k}{m}) + \left( w_{i,i} - \frac{w_i^2}{2m} \right) \right] \\ &\quad + \frac{1}{2m} \left[ \sum_{k \in \mathcal{C}_j} (w_{i,k} + w_{k,i} - \frac{w_i w_k}{m}) + \left( w_{i,i} - \frac{w_i^2}{2m} \right) \right] \\ &= \frac{1}{2m} \left[ \sum_{k \in \mathcal{C}_j} w_{i,k} + w_{k,i} - \frac{w_i w_k}{m} \right] \\ &\quad - \frac{1}{2m} \left[ \sum_{k \in \mathcal{C}_i \setminus i} w_{i,k} + w_{k,i} - \frac{w_i w_k}{m} \right]. \end{aligned} \tag{4}$$

Similarly, we can compute the *change in modularity* of (3) for the directed case by the following:

$$\begin{aligned} \Delta_{Q_d}(i, j) &= \frac{1}{m_d} \left[ \sum_{k \in \mathcal{C}_j} w_{i,k} + w_{k,i} - \frac{w_i^{\text{in}} w_k^{\text{out}} + w_k^{\text{in}} w_i^{\text{out}}}{m_d} \right] \\ &\quad - \frac{1}{m_d} \left[ \sum_{k \in \mathcal{C}_i \setminus i} w_{i,k} + w_{k,i} - \frac{w_i^{\text{in}} w_k^{\text{out}} + w_k^{\text{in}} w_i^{\text{out}}}{m_d} \right]. \end{aligned} \tag{5}$$

### 3.4 Evaluation technique

In this section, we explain the evaluation metric that we use to give an indication of the partition quality of the OD network, and to make a comparison of the obtained communities between various time slices. We explain how we can use the evaluation metric, as the ‘true’ partition of the network is not known.

In the literature, many evaluation techniques are proposed to determine the quality of the obtained network partitions. Almeida et al. [2] describe various metrics that exist to determine the quality. However, no straightforward



method exists to evaluate the quality of a partition when the ‘true’ partition is unknown. Some of the evaluation techniques can however be used to compare results and give an indication of their quality.

The most common quality metric is the normalised mutual information (NMI) [8]. We use this metric to compare our partition realisations based on their similarity. This metric is in the range of [0,1] and equals 1 if two partition realisations are identical. This value computes the mutual information between the two partitions and normalises it based on the entropy value of each realisation. The *entropy* is a value of the uncertainty present in a realisation. 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 to what extent the realisations overlap. The NMI is defined as follows:

$$NMI(P_i, P_j) = \frac{2I(P_i, P_j)}{\sqrt{H(P_i) \cdot H(P_j)}}, \tag{6}$$

where  $P_i$  and  $P_j$  denote the clustering labels,  $I(\cdot, \cdot)$  the mutual information and  $H(\cdot)$  the entropy value. 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 [16, 17].

The NMI compares two realisations, whereas we have a group of realisations and want to determine the overall similarity between these partitions. To obtain the mutual information over a group of partitions, we can compute the so-called *average-NMI*, as defined by Ana and Jain [3]

$$\text{average-NMI}(\mathcal{P}) = \sum_{i \neq j} NMI(P_i, P_j) / \binom{r}{2}, \tag{7}$$

where  $r$  denotes the number of cluster realisations,  $\mathcal{P} = \{P_1, \dots, P_r\}$  the group of partitions, and  $P_i, P_j$  the individual cluster realisations.

### 4 Preliminary cluster results

In this section, we show the resulting communities of the OD data in Amsterdam by using the Louvain algorithm. We partition the dataset based on time slices and discuss the observed differences in communities by using the evaluation metrics described in Section 3.4.

Various clustering heuristics are compared in [15] such as Fast Greedy, Walktrap, infomap and OSLOM. In contrast to the positive results on the benchmark sets used in [15], these methods proved to be unsuccessful when applied to the OD data of Amsterdam. These methods either failed to converge or returned near to zero modularity values. Near to zero modularity is an indication that the corresponding clusters do not represent any cohesion. In Fig. 8b, the clusters resulting from the undirected implementation of the Louvain algorithm are visualised, a close to zero modularity value is obtained. Visually, we observe that these clusters show a certain degree of spatial connectedness, although the low modularity value indicates that a high spatial connectedness in the network exists.

Although our data set does not consist of millions of nodes and edges, we do have a large number of edges to nodes ratio. The dataset consists of a fully connected graph, which is probably the reason that most clustering methods do not find good communities. Moreover, the directionality of the connections in the data was not included in most of these heuristics. Therefore, we continued the analysis by using an implementation of the directed Louvain method developed in [25]. An output of this method is visualised in Fig. 8a. As can be seen, the clusters that result from the Louvain method including directionality appear spatially close, although no spatial aspects are taken into account. Moreover, some of the communities have a close resemblance with the districts of Amsterdam. For example, the ‘Zuid-Oost’ district, which is more isolated from the rest of Amsterdam, is nearly covered by a single community.

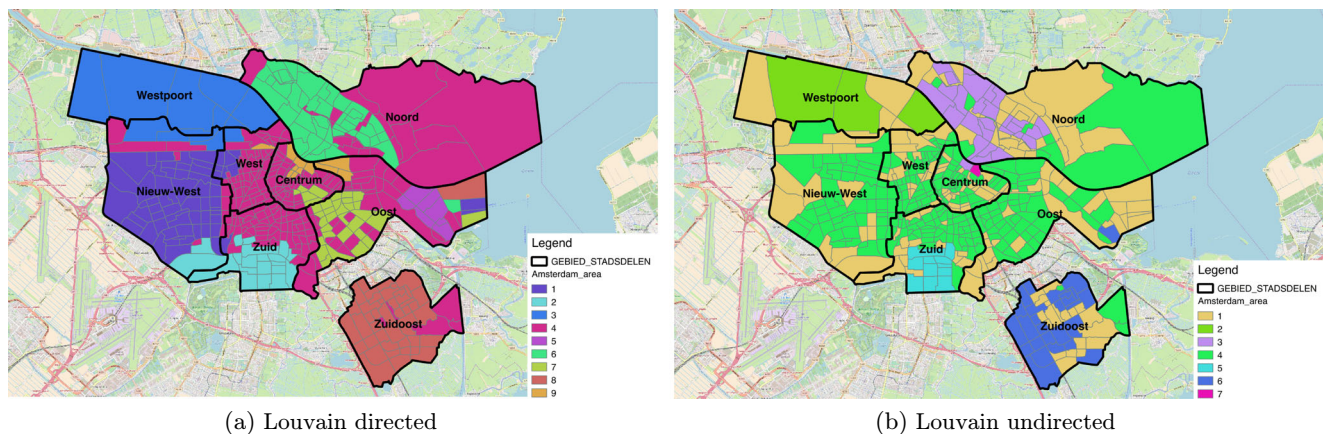


Fig. 8 Clustering with respect to destination for each district

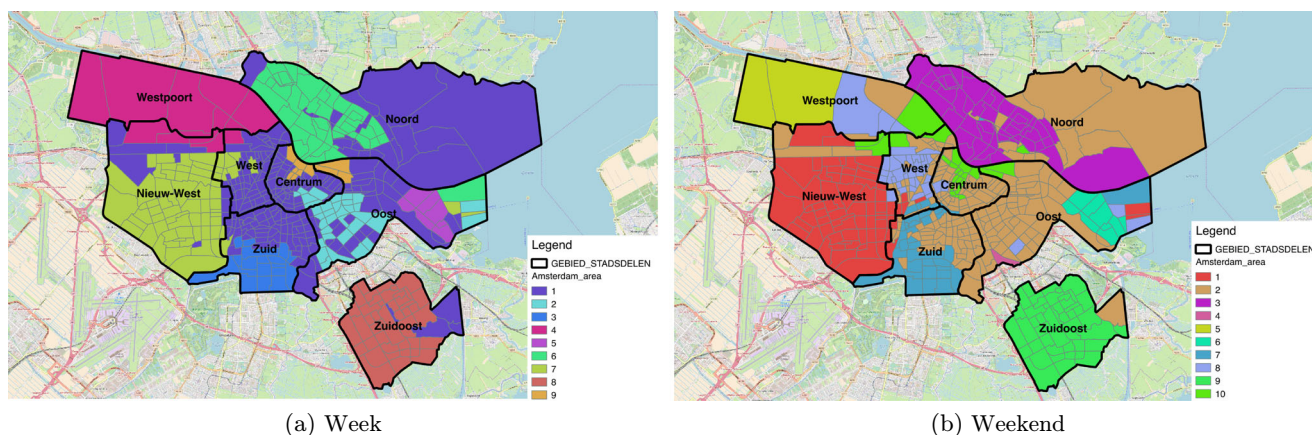


Fig. 9 Clustering with respect to destination for each district using the Louvain method

Nevertheless, the modularity value of the resulting clusters is 0.01, although larger than the undirected output, it is still rather small.

The geographical visualisation of the directed Louvain method show connected clusters and are grouped at locations that we would expect. Therefore, we explore continue to explore the results for this methods for subsets of the data. We divided the data based on the trips during the week and the weekend and applied the Louvain clustering algorithm. The results are shown in Fig. 9. In both figures, similar clusters appear. However, there are some clear differences. The main differences are the clusters in district ‘Oost’ and ‘Westpoort’ that appear only for the week data, and the cluster in the ‘Amsterdam West’ district that pops up in the weekend data. The clusters at the outskirts of Amsterdam appear to be the most prominent.

Table 3 shows the average similarities between the clustering realisations over the same dataset by using the average-NMI value of (7). We divided the data based on the ‘Total’ trips, trips during the ‘Week’ and ‘Weekend’ and the similarity results over the ‘Total period’ and on a monthly basis. The average-NMI values show that most subsets show consistent results between runs. However, the weekend data shows an overall smaller average-NMI value, especially when the monthly division is used. The consistency of the weekend data for each run is smaller compared to the week and total data sets. These variations can be caused by the smaller number of days covered, as well as less regular travel patterns in the weekend.

To analyse whether large differences and similarities between months are present, we again use the average-NMI value of (7) to compare the resulting partitions. The results are shown in Table 4. The average-NMI value of each month with itself is shown as well. The largest NMI value for each subset is with itself, denoted by the values on the diagonal. The last row compares the total data set with each month. We do not observe extreme differences between the months in this comparison.

The small modularity value appears for each of the subsets of the data. The small modularity combined with a fully connected network is not a surprising result. The fully connected graph indicates a well-connected network with a lot of interaction throughout the whole area of Amsterdam.

The results of the partitions in each of the have an overlap with the districts in Amsterdam. Although there are some differences between the municipalities and the clustering results, most of these differences can be explained with common sense and they provide insight in the structure of movements in Amsterdam. For example, the cluster IJburg which is a small neighbourhood is standing out, there are not a lot of exits causing people to stay within their neighbourhood more often. While on the other hand in the ‘Jordaan’ no clear cluster is present. This neighbourhood is very easy to reach from other parts of Amsterdam, as a results there is no separate cluster that appears.

Although we can explain the spatial appearance or abundance of certain cluster, we cannot determine whether there are clear variations between months. We therefore

Table 3 Comparison of the similarity between cluster realisations for different subsets of the data by using the similarity metric NMI

| Period  | Total period | April | May  | 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 4** Comparison of the similarity between cluster realisations for monthly subsets of the data by using the similarity metric NMI

| Period       | April | May  | June | July | August | September | Total period |
|--------------|-------|------|------|------|--------|-----------|--------------|
| April        | 0.85  | —    | —    | —    | —      | —         | —            |
| May          | 0.75  | 0.86 | —    | —    | —      | —         | —            |
| June         | 0.74  | 0.76 | 0.91 | —    | —      | —         | —            |
| July         | 0.77  | 0.76 | 0.75 | 0.87 | —      | —         | —            |
| August       | 0.73  | 0.73 | 0.72 | 0.77 | 0.86   | —         | —            |
| September    | 0.73  | 0.73 | 0.72 | 0.75 | 0.72   | 0.87      | —            |
| Total period | 0.82  | 0.83 | 0.82 | 0.84 | 0.79   | 0.78      | 0.93         |

continue our analysis to determine the strength of the communities.

### 5 Robustness of communities

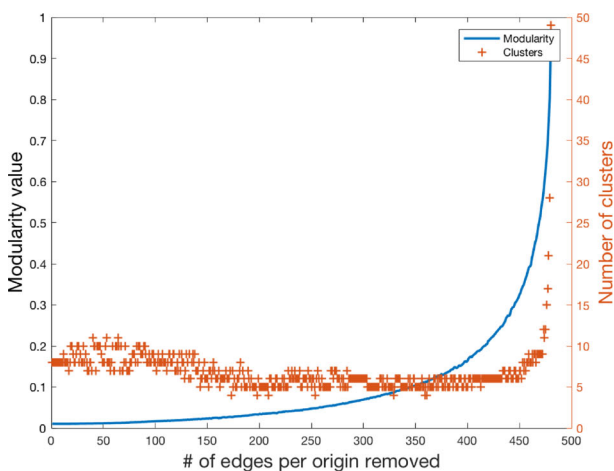
In the previous sections, we observed the spatially connected partitions of OD data in Amsterdam. To gain more insight in the generated communities, we analyse the strength of the communities relative to each other in terms of connectivity.

To analyse what fraction of all edges contributes to the detection of these district boundary clusters, we propose a simple method for analysis. We remove the smallest  $x$  weights edges from each neighbourhood, where  $x \in \{1, \dots, n\}$  and  $n$  denotes the number of nodes of the network. In Fig. 10a the results show that the modularity value increases when the number of smallest weights  $x$  removed increases, as would be expected. More interestingly, the number of clusters found remains relatively constant until almost all values are removed. In Fig. 10b the clusters found when 10% of the smallest weights were removed are visualised. It can be seen that

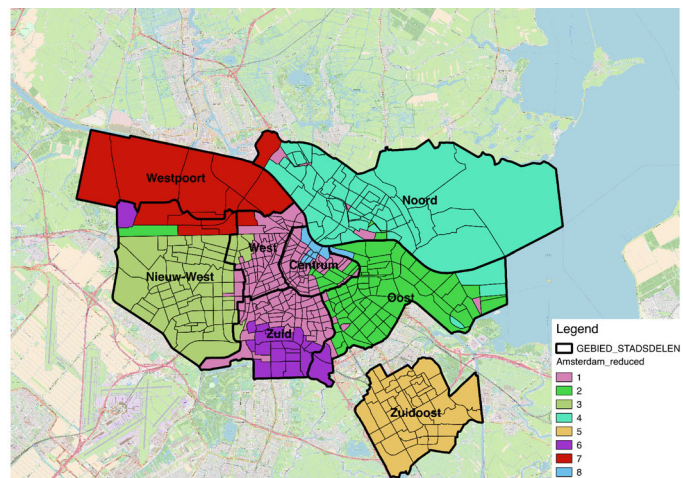
this partition represents the regional boundaries even more closely than the clusters of the complete set. This suggests that although the trips within Amsterdam are well spread, trips within regional boundaries have higher weights in almost every district. Only part of the ‘Centrum’, ‘West’ and ‘Zuid’ region remain connected as one cluster.

In addition to the question which edges contribute to the spatially connected clusters, another question that arises is the connectedness of the communities with respect to each other. Which are the most prominent communities in the dataset, and which communities are less prominent. Although there is no specific metric available in the literature to evaluate such property of connectedness [12, Chapter XIV], in [14], the authors evaluate the connectedness by adding random noise to the edge weights.

We applied the same methodology as in [14]. We add random weights to the edges with a predefined variance. Thus, for each edge in the network, we draw a random variable  $X$ , where  $X \sim N(0, \sigma^2)$ , add these values to the OD matrix, run the Louvain algorithm and visualise the obtained clusters. We gradually increase the value of  $\sigma$  and evaluate the resulting clusters between each increment until no coherent structure can be found. This gives an indication



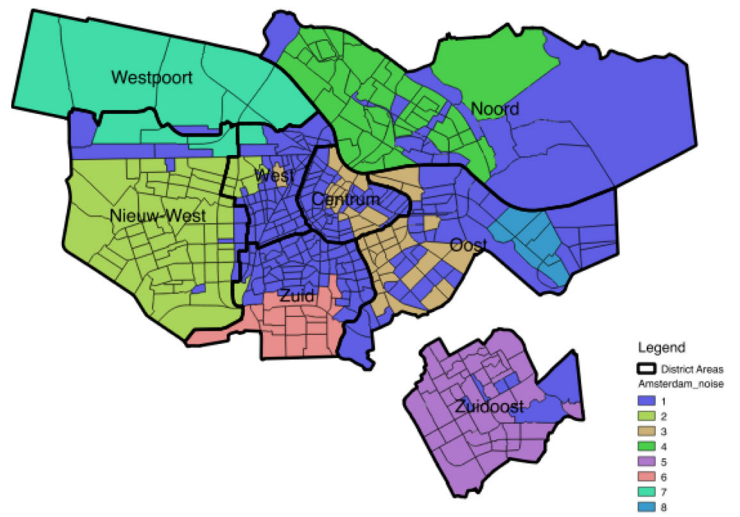
(a) Modularity and cluster results



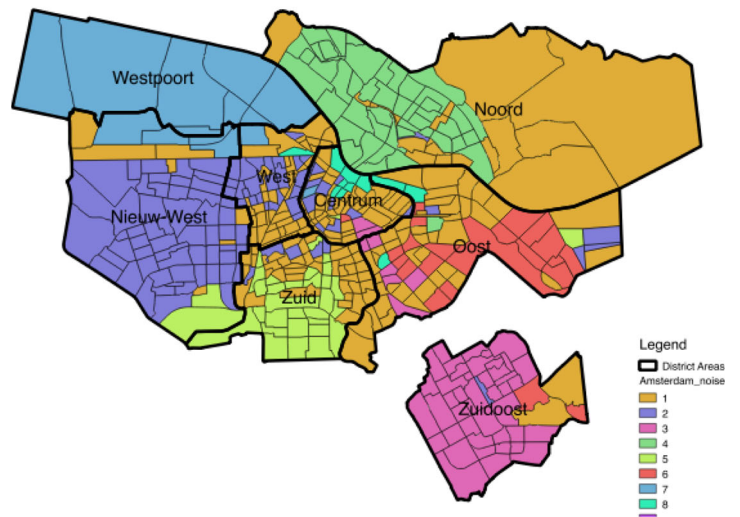
(b) Results based on removal of 10%

**Fig. 10** Cluster analysis for increasing number of removed edges per neighbourhood

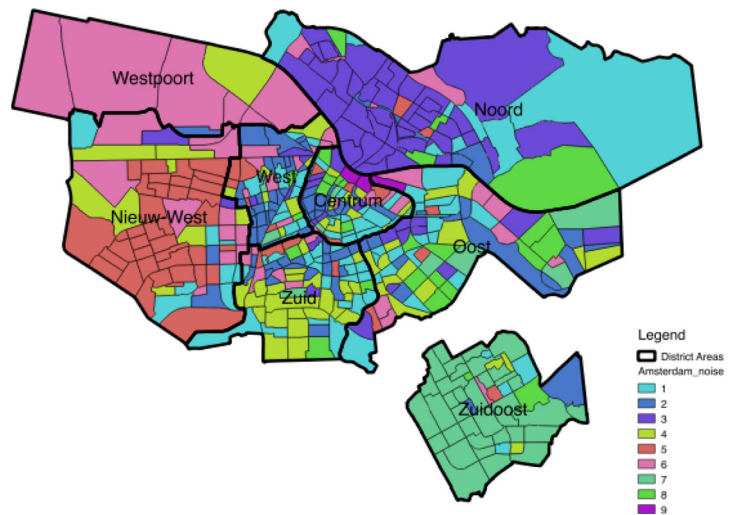
**Fig. 11** Clustering results under random noise addition



(a) Random noise with  $\sigma = 0.5$



(b) Random noise with  $\sigma = 1$



(c) Random noise with  $\sigma = 2$

of the connectedness of each community relative to the other communities in a visual manner.

In Fig. 11, a realisation for an increasing variability in random noise is shown. It should be kept in mind that these graphs only show the result of a single realisation and are only indicative of the impact of noise. We observe that the cluster ‘Zuidoost’ and ‘Noord’ remain visible although a large noise value is added to the edges. The cluster ‘Centrum’ is the first to disappear, and dissolves in the ‘Oost’ cluster. This is a first step towards analysis of connectedness of clusters with respect to each other. In [14], a more thorough analysis of the consistency of the individual nodes is applied. For the current analysis, we have so far only used visually indicative results. In Section 6.2, we continue introduce a method that reduces the variations between each realisation of the same subset to obtain more consistent results at each run, allowing for comparison between subsets.

## 6 Consistency of communities

So far, we analysed the community structure of the communities resulting from the Louvain clustering heuristic method applied on the OD data set. We observed variations between realisations of the same dataset and between subsets of the data. To compare the communities of the subsets relative to each other, we need a procedure that gives more or less consistent communities when applied on the same subset. We use a procedure called *consensus clustering* to obtain this consistency. Consensus clustering is an ensemble learning method that combines multiple realisations to create a more consistent final result. An example applied to graph clustering is explained in [22].

### 6.1 Consensus clustering procedure

To determine the consistency of each community, we analyse which neighbourhoods characterise the community. In this section, we explain the procedure to obtain such a characterisation for the current data set.

The Louvain method aims to maximise the modularity value in a greedy manner. The greedy approach makes it computationally efficient, and makes it applicable for clustering on large datasets. However, due to a randomisation in the approach, each realisation can deviate from a previously obtained realisation. The algorithm evaluates nodes based on their modularity gain when clustered. Due to randomisation in the order of which these nodes are evaluated, deviations in initial clusters occur. As the algorithm progresses, these initial clusters can result in a node ending up in another cluster than for other initial clusters. Moreover, some communities might not appear due

to initial clusterings of nodes which in other realisations belong to different communities. We want to exploit these variations to find the neighbourhoods that can be defined as the ‘core’ of the community, as well as the neighbourhoods that are on the boundary between communities.

A method known as cluster ensemble learning can be used to obtain the core cluster result, ensemble-based learning is a procedure that combines the results of a certain number of weak learners to obtain a final more robust result. In [13], an ensemble learning procedure is explained which they call evidence accumulation clustering. We use this procedure to obtain our final partition. The evidence accumulation method is composed of three steps. We will explain each step and specify the implementation that we choose to generate our results.

**Step 1 (Generating an ensemble):** A cluster ensemble is generated consisting of  $m$  clustering partitions, denoted by

$$\begin{aligned} \mathcal{P} &= (P_1, \dots, P_m) \\ P_1 &= (c_1^{(1)}, c_2^{(1)}, \dots, c_n^{(1)}) \\ &\vdots \\ P_m &= (c_1^{(m)}, c_2^{(m)}, \dots, c_n^{(m)}), \end{aligned}$$

where  $c_i^{(j)} \in \{1, \dots, C^{(j)}\}$  gives the cluster  $k \in \{1, \dots, C^{(j)}\}$  that node  $i$  of partition  $j$  is assigned to. These partitions can be obtained by either using different representations of the data, the choice of algorithms, or the algorithmic parameters. The randomised order of the node evaluations in the Louvain algorithm causes variations between each realisation in our dataset, which make it an appropriate method to apply the algorithmic parameter approach. The randomisation of the nodes is then the parameter adjustment.

**Step 2 (Determine the similarity):** The second step is to combine the cluster realisations by combining ‘evidence’ in the so-called *co-association matrix*  $A$ , with entries  $A = (a_{i,j})$ , with the following:

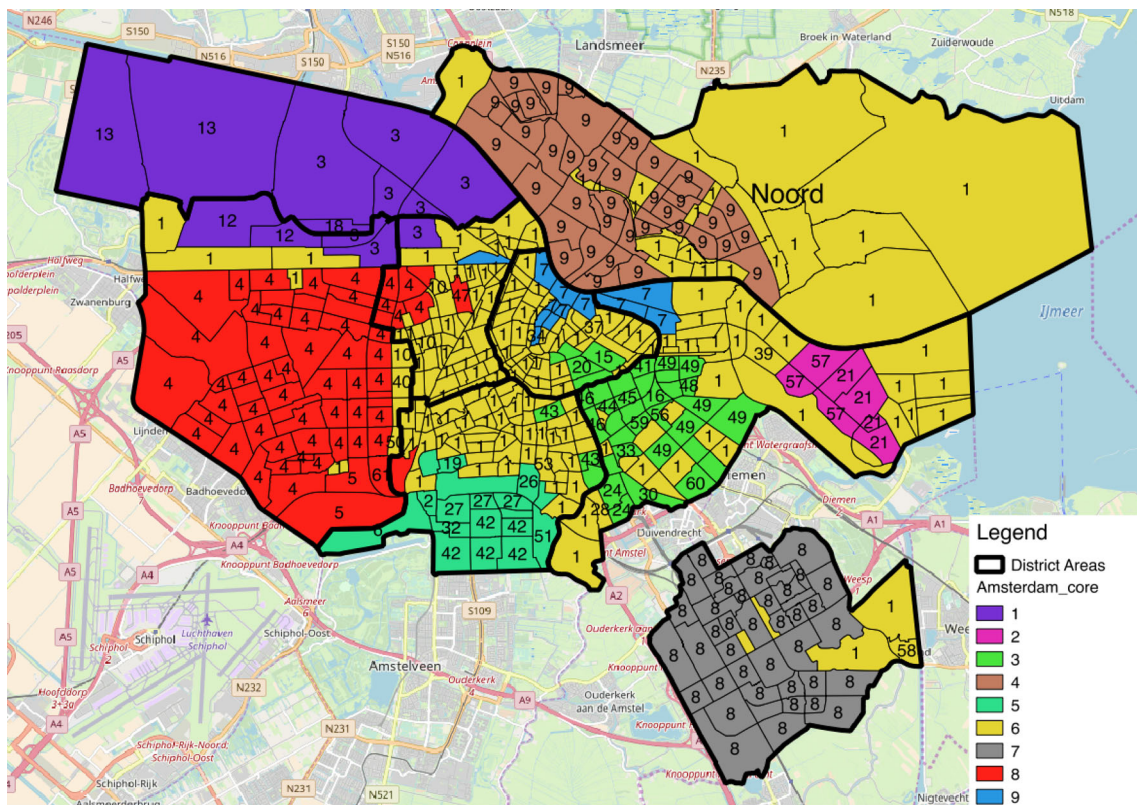
$$a_{i,j} = \frac{n_{i,j}}{m}, \tag{8}$$

where  $n_{i,j} = \sum_{k=1}^m \mathbf{1}_{\{c_i^{(k)}=c_j^{(k)}\}}$  represents the number of times that nodes  $i$  and  $j$  belong to the same cluster among the  $m$  partitions.

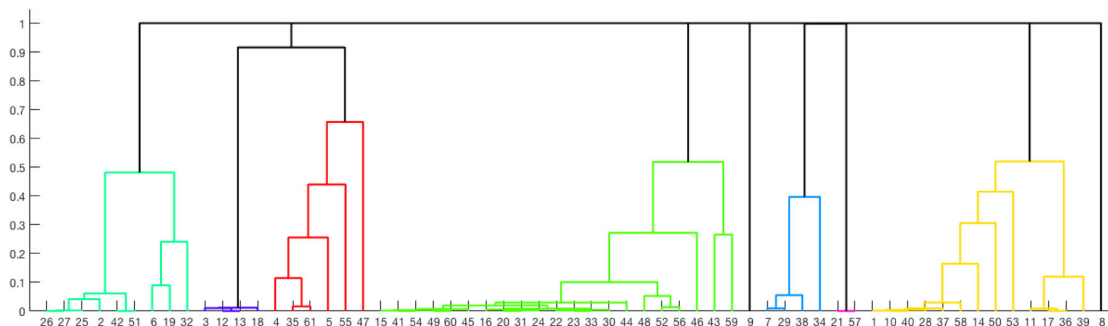
**Step 3 (Obtain the final partition):** The final step in the evidence-based clustering method is to obtain the final cluster partition from the generated similarity matrix  $A$ . Any clustering can be applied over this matrix to generate this partition. A hierarchical clustering algorithm is used to combine the nodes and generate the resulting dendrogram [9]. This last step can become complicated

when the number of nodes is large. However, in [13], Fred and Jain propose to group similar nodes together before generating the dendrogram. We use this approach to group the nodes that belong to the same partition in all iterations before generating the final dendrogram. We obtain our matrix  $A$  by applying the following three steps:

- (a) We combine the nodes which are in the co-association matrix  $A$  with the value 1, meaning that they are grouped in the same cluster for all realisations. This gives a reduced form matrix  $A' \subset A$ .
- (b) The subset  $A'$  is then used to obtain the dendrogram using the complete-link method [9]. The complete-link method computes the dendrogram based on the furthest neighbour method. This method is known to generate clusters that are well separated and compact and is one of the most commonly used methods for hierarchical clustering. As it computes the furthest neighbour, we have to determine the dissimilarity between nodes. This means that we use  $A'' = 1 - A'$ .
- (c) Having obtained the dendrogram, we then need to determine the cutoff value to disentangle the dendrogram into separate clusters. We determine the



(a) Core cluster



(b) Core cluster dendrogram

Fig. 12 Core clusters for the complete data set

cutoff value that leads to the identification of  $k$  clusters, where we set  $k$  equal to the closest integer value of the mean number of clusters from the cluster ensemble input  $\mathcal{P}$ .

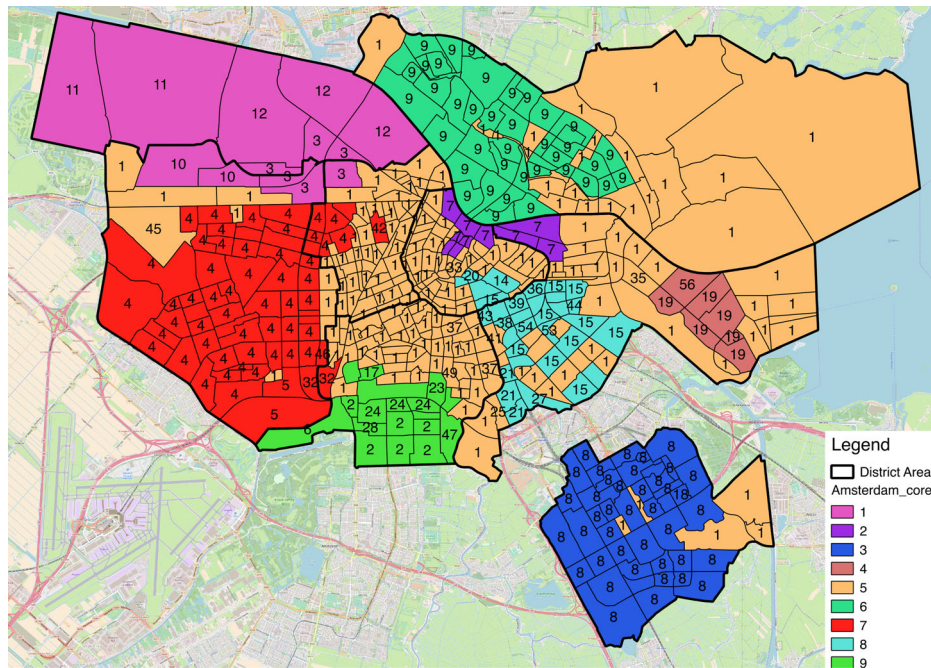
In the next section, we show the results obtained from the above procedure.

### 6.2 Experimental results

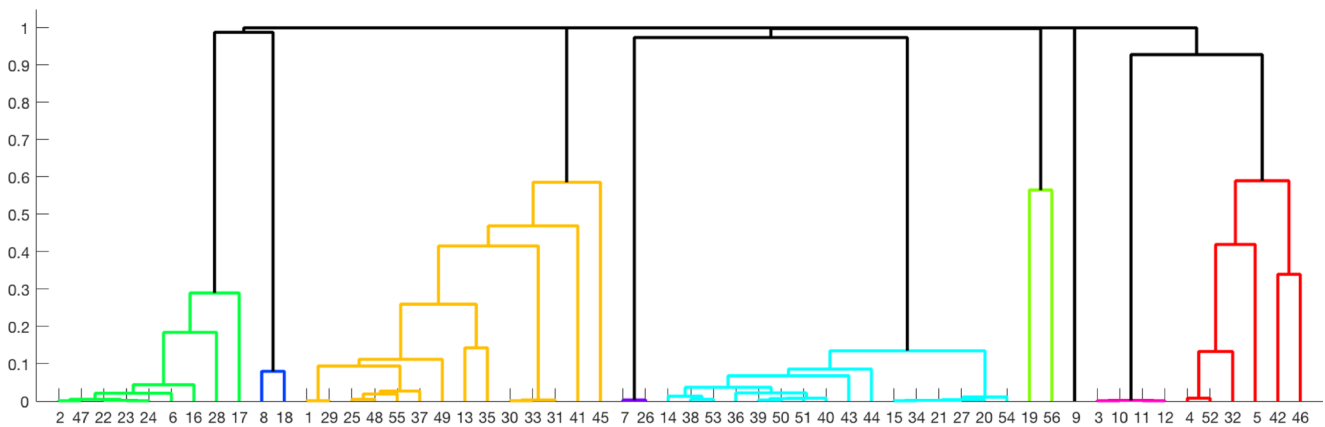
We apply the evidence-based learning algorithm to show the consistency and variation between communities and neighbourhoods. We use this approach to compare the monthly subsets in a more robust manner as well.

We applied the ensemble learning method initially over the whole data set. We computed the results based on  $N = 1000$  realisations of the Louvain algorithm and computed the co-association matrix. We grouped the nodes of the co-association matrix of (8) when they belong to the same community over all realisations. This results in a subset of the co-association matrix of size 6, of which 34 values are due to individual nodes. For the current analysis, the size of the dendrogram is still manageable. However, when many individual nodes occur, the reduction of the co-association matrix can also be performed based on a high similarity value.

The resulting core clusters of the total OD data are visualised in Fig. 12. Figure 12a of the final core cluster



(a) Core cluster business days



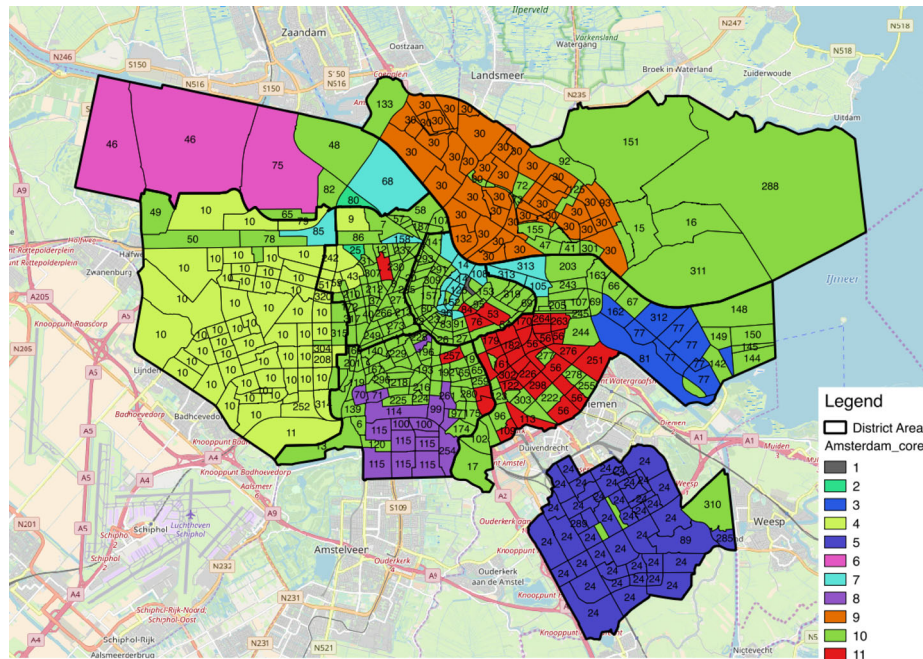
(b) Core cluster dendrogram business days

Fig. 13 Core clusters for the week data set

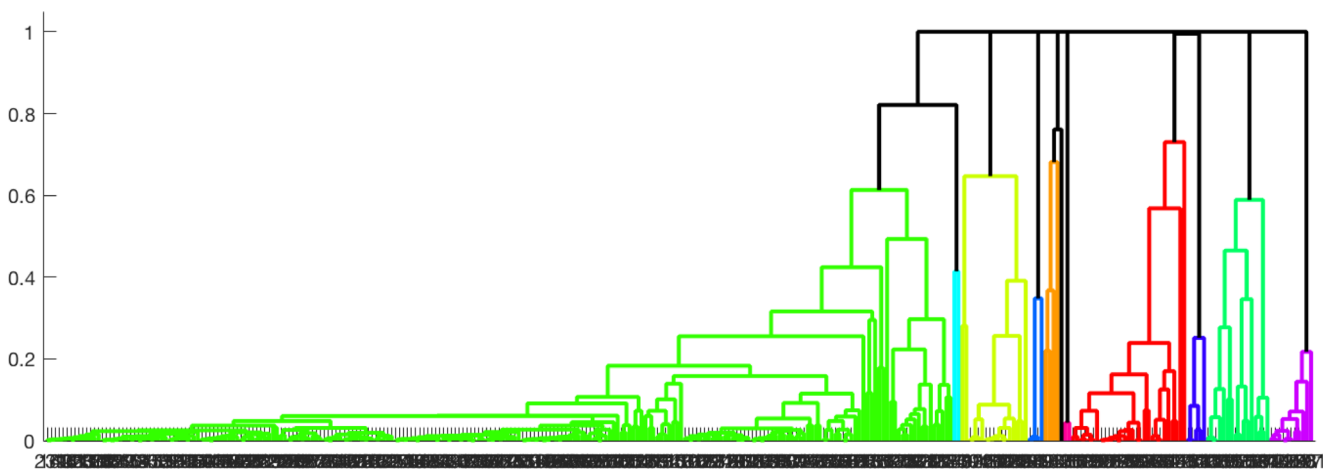
assignments by individual colours. The value in each neighbourhood represents the initial partition of 61 values in the co-association matrix. The dendrogram of Fig. 12b shows the dissimilarity between the node groups that are grouped together. For example, the pink group with node groups 21 and 57 have a dissimilarity value near to zero. This means that in only a few realisations of the algorithm they were not assigned to the same cluster. On the opposite side, we can observe that node group 47, denoted by red, does not have a large similarity value compared to the other

node groups in this cluster. It is interesting to observe that we have two core clusters consisting of only a single group of nodes. These two node groups were identified in all  $N$  realisations, meaning that the neighbourhoods in these groups were consistently grouped together.

We applied the same analysis for the weekdays and weekend subsets of the OD dataset. The resulting partitions and dendrograms are visualised in Figs. 13 and 14. Especially for the weekend subset, we observe more diversity between the clustering result. The dendrogram of Fig. 14



(a) Core cluster weekend



(b) Core cluster dendrogram weekend

Fig. 14 Core clusters for the weekend data set



**Table 5** Average NMI values of the consensus clustering result based on monthly data

| Period       | April | May  | June | July | August | September | Total period |
|--------------|-------|------|------|------|--------|-----------|--------------|
| April        | 0.97  | –    | –    | –    | –      | –         | –            |
| May          | 0.79  | 0.98 | –    | –    | –      | –         | –            |
| June         | 0.76  | 0.79 | 0.96 | –    | –      | –         | –            |
| July         | 0.81  | 0.83 | 0.76 | 1.00 | –      | –         | –            |
| August       | 0.76  | 0.78 | 0.72 | 0.81 | 0.95   | –         | –            |
| September    | 0.74  | 0.74 | 0.70 | 0.74 | 0.71   | 0.90      | –            |
| Total period | 0.85  | 0.88 | 0.82 | 0.90 | 0.81   | 0.77      | 0.98         |

**Table 6** Average NMI values of the consensus clustering result based on monthly data during business days

| Period       | April | May  | June | July | August | September | Total period |
|--------------|-------|------|------|------|--------|-----------|--------------|
| April        | 0.97  | –    | –    | –    | –      | –         | –            |
| May          | 0.76  | 0.97 | –    | –    | –      | –         | –            |
| June         | 0.77  | 0.79 | 0.99 | –    | –      | –         | –            |
| July         | 0.75  | 0.75 | 0.71 | 0.93 | –      | –         | –            |
| August       | 0.74  | 0.77 | 0.72 | 0.76 | 0.95   | –         | –            |
| September    | 0.75  | 0.74 | 0.72 | 0.71 | 0.72   | 0.95      | –            |
| Total period | 0.70  | 0.73 | 0.70 | 0.72 | 0.73   | 0.64      | 0.98         |

**Table 7** Average NMI values of the consensus clustering result based on monthly data during the weekend

| Period       | April | May  | June | July | August | September | Total period |
|--------------|-------|------|------|------|--------|-----------|--------------|
| April        | 0.64  | –    | –    | –    | –      | –         | –            |
| May          | 0.38  | 0.66 | –    | –    | –      | –         | –            |
| June         | 0.38  | 0.40 | 0.66 | –    | –      | –         | –            |
| July         | 0.40  | 0.45 | 0.44 | 0.69 | –      | –         | –            |
| August       | 0.38  | 0.40 | 0.41 | 0.42 | 0.69   | –         | –            |
| September    | 0.38  | 0.39 | 0.39 | 0.42 | 0.39   | 0.65      | –            |
| Total period | 0.49  | 0.54 | 0.55 | 0.57 | 0.52   | 0.53      | 0.99         |

**Table 8** Number of clusters in the final core cluster result

| Period    | All | Week | Weekend |
|-----------|-----|------|---------|
| April     | 8   | 9    | 8       |
| May       | 9   | 9    | 9       |
| June      | 9   | 10   | 8       |
| July      | 9   | 9    | 8       |
| August    | 10  | 10   | 9       |
| September | 7   | 8    | 7       |
| Total     | 9   | 9    | 9       |

**Table 9** Cutoff values, representing the highest dissimilarity between cluster branches that were combined to obtain the core cluster results

| Period    | All  | Week | Weekend |
|-----------|------|------|---------|
| Total     | 0.51 | 0.70 | 0.51    |
| April     | 0.63 | 0.67 | 0.95    |
| May       | 0.69 | 0.57 | 0.96    |
| June      | 0.62 | 0.57 | 0.92    |
| July      | 0.48 | 0.69 | 0.93    |
| August    | 0.66 | 0.67 | 0.94    |
| September | 0.57 | 0.61 | 0.92    |

b shows that in particular the centre cluster shows large fluctuations over the partitions. No major differences are observed between the week and weekend partitions, suggesting that travel behaviour shows similar groups for the week and weekend days.

We continue to use the cluster ensemble technique to obtain more robust results for the monthly subsets. In Tables 3 and 4, the NMI values of the cluster partitions of the same data set were relatively low. Making it hard to draw conclusions when compared to each other. We use the cluster ensemble method and run this method several times to compute the average-NMI values over the subsets. The results are shown in Tables 5, 6 and 7. It shows that the self-similarity is increased for the total and weekly data set, obtaining values close to 1. This allows for a better comparison between the several months as the monthly subset gives more self-consistent results.

The average-NMI values in Tables 5 to 7 show that in particular September gives a lower NMI value compared to other months. To determine whether specific results deviate, visual representations of the maps should be compared. However, we first analysed the number of clusters that were formed for each month. The average number of clusters for each monthly subset are shown in Table 8 for the weekdays, weekend and total set. Interestingly, the September month shows fewer clusters compared to the other months, possibly explaining the lower similarity value. June and August result in slightly more clusters compared to the other months. We expect that the main differences between the resulting core clusters are caused by the number of partitions.

An easy analysis of the partition differences observed in Table 9 of August and September is a geographical visualisation of the core result obtained from the ensemble method. In Fig. 15, both months are shown. We observe

that in Fig. 15b, representing September, the centre cluster disappeared. A possible explanation for the absence of this cluster is the end of the tourist season, generating fewer trips in the city centre. As for August, we observe a very prominent cluster at the border of district ‘West’ and ‘Zuid’, consisting of only one neighbourhood. This neighbourhood consists of the largest city park in Amsterdam, which is a famous hotspot during warm weather.

Finally, we can interpret the cutoff values that are used to determine the cluster results. The cutoff value is determined by the dissimilarity value for which we obtain a specific number of clusters, which is equal to the average number of clusters in the ensemble of partitions  $\mathcal{P}$ . The higher the cutoff value, the higher dissimilarity value to combine the correct number of partitions. As expected, the cutoff value is larger for the monthly weekend data, again confirming our observations that the weekend trip data shows less consistent clusters.

## 7 Conclusion

In this paper, we analysed travel behaviour in Amsterdam based on origin-destination travel intensity data. We analysed both the spatial variation as well as the time-dependent variation of trips. We proceeded our analysis by using clustering techniques based on modularity optimisation to separate regions based on internal travel behaviour. We used a heuristic technique to separate these regions and exploit this to analyse the consistency of the obtained results. Finally, we discovered deviations in the patterns in time.

The weekly pattern and spatial plots confirm expected behaviour, such as the morning and evening commute. We observe that the trips taken from the metro region of

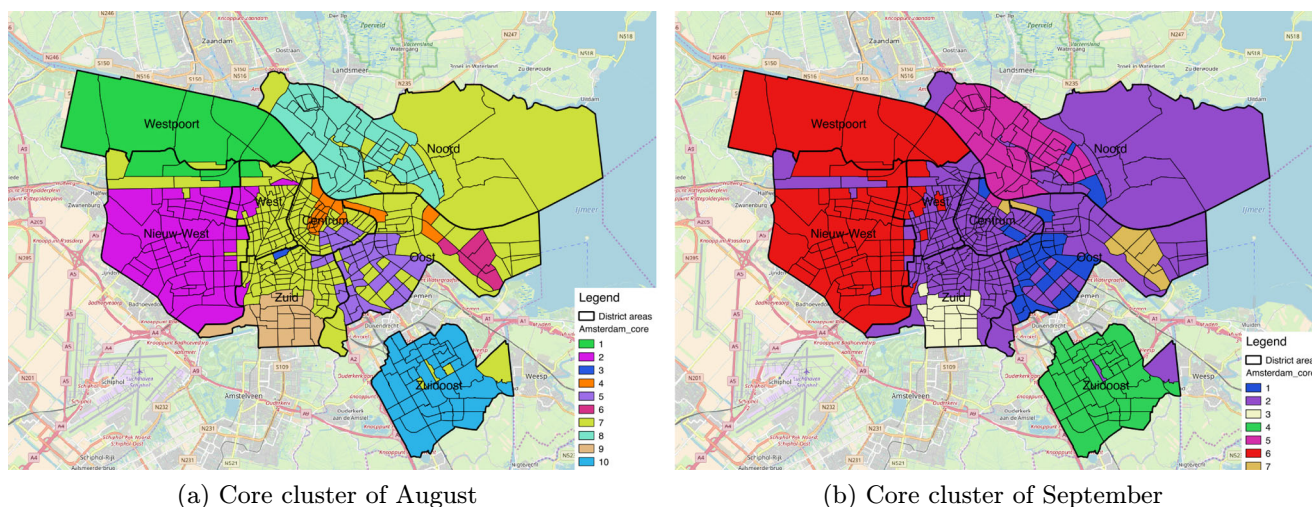


Fig. 15 Core clusters for the complete data set

Amsterdam are largely commuting trips. There are three areas that show a high density of trips coming from the metro region, each of them contains large business districts. However, from this analysis, we also discovered a gap between the total inflow and outflow. As there is no logical explanation for this behaviour, we assume that this occurs due to some transformation to censor the data. In order to properly analyse the data, we restored this imbalance and obtained scaling values for each neighbourhood. This revealed a couple of outliers in the data. Especially, in the east of Amsterdam, a few neighbourhoods which mostly consist of water showed a large difference between the total inflow and outflow value. With these outliers in mind we continued our analysis.

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 centre 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 and weekday trips is introduced, the result suggest that we observe slightly different clusters in the weekend compared to the week data. Although this is hard to conclude, given to the inconsistency between results of the same data partition.

We analysed 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 over 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 behaviour at each area for leisure.

Finally, we used a cluster-ensemble technique to obtain more consistent results, allowing for a better comparison. The results from the cluster ensemble method show minimal deviations between the obtained clusters over the time-dependent subsets, regarding week and weekend. The monthly subsets revealed some differences in the number of partitions obtained in each month. Nevertheless, no big differences in clusters were found between these subsets, suggesting that the partitioning is quite robust over the entire period.

The partitioning results give an indication of the connectedness through the city. It provides high-level partitions that can be used to analyse major flows through the city. This can help policy-makers deciding where the road network of public transport network is insufficient. Moreover, major changes in city planning can be analysed in this aggregated manner. A main limitation of the current

work is that the trip information involves all modalities together. It would be interesting to analyse each modality separately to be able to determine the clustering aspect for specific modalities.

For the long run, city planners could use this type of analysis to see the city structure and plan accordingly. They can compare the results to the structure in other cities to see similarities. A time-dependent analysis for a group of cities would allow them to predict the changes in the future for cities with similar characteristics in different points in time.

Further research could involve the quantification of the strength of each cluster. In this paper, we focussed on the overall strength, an extension would be to quantify this strength per cluster. Another approach would be to determine the variety of each cluster by means of an NMI value per cluster. This would give a representation of the strength of a cluster in terms of consistency.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

1. Cbs wijk- en buurtkaart, 2017
2. Almeida H, Guedes D, Meira W, Zaki MJ (2011) Is there a best quality metric for graph clusters? In Joint European conference on machine learning and knowledge discovery in databases, pp 44–59. Springer
3. Ana LNF, Jain AK (2003) Robust data clustering. In: Proceedings in computer IEEE computer society conference on computer vision and pattern recognition, 2003, vol 2, pp II–II. IEEE
4. Ben-Akiva ME, Lerman SR (1985) Discrete choice analysis: theory and application to travel demand, vol 9. MIT press
5. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2011) The Louvain method for community detection in large networks. *J Stat Mech: Theory Exp* 10:P10008
6. Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. *J Stat Mech: Theory Exp* 2008(10):P10008
7. Brandes U, Delling D, Gaertler M, Görke R, Hoefler M, Nikoloski Z, Wagner D (2006) On modularity- $np$ -completeness and beyond. Univ., Fak. für Informatik, Bibliothek
8. Cover TM, Thomas JA (2012) Elements of information theory. pp 13–55
9. Day WHE, Edelsbrunner H (1984) Efficient algorithms for agglomerative hierarchical clustering methods. *J Classif* 1(1):7–24
10. Dugué N, Perez A (2015) Directed Louvain: maximizing modularity in directed networks. PhD thesis, Université d'Orléans
11. Dugundji E, Walker J (2005) Discrete choice with social and spatial network interdependencies: an empirical example using mixed generalized extreme value models with field and panel effects. Transportation Research Record: Journal of the Transportation Research Board 1921:70–78
12. Fortunato S (2010) Community detection in graphs. *Physics reports* 486(3):75–174

13. Fred ALN, Jain AK (2005) Combining multiple clusterings using evidence accumulation. *IEEE Trans Pattern Anal Mach Intell* 27(6):835–850
14. Gfeller D, Chappelier J, De Los Rios P (2005) Finding instabilities in the community structure of complex networks. *Phys Rev E* 72(5):056135
15. Lancichinetti A, Fortunato S (2009) Community detection algorithms: a comparative analysis. *Phys Rev E* 80(5):056117
16. Lancichinetti A, Fortunato S, Kertész J (2009) Detecting the overlapping and hierarchical community structure in complex networks. *New J Phys* 11(3):033015
17. Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. *Phys Rev E* 78(4):046110
18. Leicht EA, Newman MEJ (2008) Community structure in directed networks. *Phys Rev Lett* 100(11):118703
19. Maness M, Cirillo C, Dugundji ER (2015) Generalized behavioral framework for choice models of social influence: behavioral and data concerns in travel behavior. *J Transp Geogr* 46:137–150
20. Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. *Phys Rev E* 69(2):026113
21. Norris JR, Chains M (1997) *Cambridge series in statistical and probabilistic mathematics*. Cambridge University Press, Cambridge
22. Ovelgönne Michael, Geyer-Schulz Andreas (2012) An ensemble learning strategy for graph clustering. *Graph Partitioning and Graph Clustering* 588:187
23. Ratti C, Sobolevsky S, Calabrese F, Andris C, Reades J, Martino M, Claxton R, Strogatz SH (2010) Redrawing the map of Great Britain from a network of human interactions. *PloS one* 5(12):e14248
24. Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. *Proc Natl Acad Sci* 105(4):1118–1123
25. Scherrer A (2008) Matlab Louvain implementation. online
26. Tesselkin A, Khabarov V (2017) Estimation of origin-destination matrices based on Markov chains. *Procedia Eng* 178:107–116
27. Wichmann B, Chen M, Adamowicz W (2016) Social networks and choice set formation in discrete choice models. *Econometrics* 4(4):42

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.