Journal of Traffic and Transportation Management
Volume 2, No. 2 (2020) pp. 35-42

An iterative k-means clustering approach for identification of bicycle impediments in an urban traffic network

Johan Holmgren[a] *, Luk Knapen[b,c], Viktor Olsson[a], Alexander Persson Masud[a]

[a]*Internet of Things and People Research Center & Dept. of Computer Science and Media Technology, Malmö University, Malmö, Sweden, 205 06 Malmö*
[b]*Hasselt University, Hasselt, Belgium, 3500 Hasselt*
[c] *VU Amsterdam, Amsterdam, The Netherlands, 1081 HV Amsterdam*

Abstract

The bicycle has many positive effects; however, bicyclists are more vulnerable than users of other transport modes, and the number of bicycle related injuries and fatalities are too high. We present a clustering analysis aiming to support the identification of the locations of bicyclists' perceived unsafety in an urban traffic network, so-called bicycle impediments. In particular, we present an iterative k-means clustering approach, which in contrast to standard k-means clustering, enables to remove outliers and solitary points from the data set. In our study, we used data collected by bicyclists travelling in the city of Lund, Sweden, where each data point defines a location and time of a bicyclist's perceived unsafety. The results of our study show that 1) clustering is a useful approach in order to support the identification of perceived unsafe locations for bicyclists in an urban traffic network and 2) it might be beneficial to combine different types of clustering to support the identification process. Furthermore, using the adjusted Rand index, our results indicate high robustness of our iterative k-means clustering approach.

Keywords: *Cluster analysis; k-means; iterative k-means; DBSCAN; Click-point data; bicycle impediment*

1. Introduction

The bicycle is considered a sustainable, fast, cost efficient, and healthy alternative to the car for urban transport. The list of positive effects of the bicycle can be made long; however, a negative aspect is that bicyclists are unprotected, hence more vulnerable than the users of other transport modes, in particular car and public transport. Still, it has been shown that the positive health effects of bicycling significantly overshadow the risk of getting injured or dying from an accident. For example, Andersen et al. [1] argue that bicyclists are expected to live longer even though they are exposed to an increased risk of getting injured or dying in an accident.

Even though the number of road fatalities are steadily decreasing, there are still too many accidents involving bicyclists. For example, approximately 2000 bicyclists died in road accidents in the European Union (EU) countries during 2016 [2]. The total, yearly, number of road fatalities in EU has significantly decreased from approximately 43.000 to approximately 26,000 from 2007 to 2016. The number of bicycle fatalities dropped from approximately 2500 to approximately 2000 from 2007 to 2010; however, since 2010, the number of bicycle fatalities has remained on the same level. It is, therefore, obvious that there is a need for improvements in order to provide a safer traffic environment for bicyclists, hence contributing to increasing the attractiveness of the bicycle. This is important as the bicycle plays an important role in the strive towards a sustainable society, where urban transport to a larger extent than today is carried out by green transport modes.

We let the term *bicycle impediment* refer to a location in an urban traffic network where bicyclists tend to feel unsafe. By identifying bicycle impediments, it is possible for the responsible authorities to focus their safety improving investments on the locations that are perceived unsafe by bicyclists. However, it is not always straightforward to identify the (location of) bicycle impediments in an urban transport network. Historic accident statistics is an important source to support the identification of bicycle impediments; however, we argue that the accident statistics does not give a complete view on the locations of impediments as it does not take into consideration the bicyclist's own perception of unsafety.

Holmgren et al. [3] contribute a clustering analysis, aiming to identify bicycle impediments in the city of Lund, Sweden, where each of the identified clusters represents a potential bicycle

* Corresponding author. Tel.: +4666576888
E-mail: johan.holmgren@mau.se

DOI: 10.5383/JTTM.02.02.005

impediment. The clustering analysis is based on a set of so-called *click-point* data collected by bicyclists, who were instructed to push a button mounted on their bicycle handlebars, each time they experienced unsafety in the traffic situation. The analysis by Holmgren et al. suggests that bicycle impediments typically occur at particular places (e.g., in an intersection) or along roads, corresponding to compact and stretched clusters, respectively. The compact clusters are identified by applying k-means clustering, and the stretched clusters are identified using DBSCAN. In particular, Holmgren et al. introduce an iterative k-means clustering approach, which, in contrast to standard k-means clustering, enables to suggest 1) solitary points (including outliers) from the data set and 2) a value of k, that is, the number of clusters to be generated by the k-means algorithm. We define a *solitary point* as a point that is not located sufficiently close to sufficiently many other points, according to specified thresholds.

In the current paper, we extend the paper by Holmgren et al. [3], by providing an in-depth analysis of the iterative k-means clustering approach. In particular, we extend the work by Holmgren et al. by providing a stability analysis of the suggested approach in order to show its applicability for spatial clustering.

The identification of accident locations, often referred to as accident hotspots, is widely discussed in the literature, see, for example, Cheng & Washington [4] and Montella [5] for overviews. Several studies use clustering in order to identify accident hotspots, for example, Anderson [6] use k-means clustering and kernel density estimation in order to identify accident hotspots and typical groups of road users involved in accidents, and Xu & Tao [7] use ensemble clustering in order to identify accident hotspots. The existing studies differ from our work in that they mainly focus on historic data on accidents, whereas we focus on perceived unsafety of bicyclists. One exception is the study of Persson Masud & Olsson [8] which makes use of k-means clustering for the same data set as we used in the current study. The focus of their work is to compare different approaches for controlling the size of the generated clusters.

Each impediment (i) is bound to a particular yet unknown location that does not necessarily coincide with one of the observations and (ii) has a limited spatial extent. Therefore, the current paper introduces the idea to add an upper bound for the cluster diameter together with a lower bound for the number of cluster members and, hence, adds the possibility to disregard the solitary points.

Rodriguez & Laio [9] develop the *density peak clustering* (DPC) method. The basic idea is that cluster centers are located in regions of high density and far apart from each other. The method is based on ranking of distance values (as opposed to distance values themselves). The authors claim that their method is therefore less sensitive to the density threshold value. For each element the number of elements within a given threshold are counted. This is defined as the density ρ for the element. Then a value δ is defined as the minimum distance to an element having a larger density. δ and ρ are shown in a decision diagram. Each element having a large ρ and a large δ represents a cluster center. Finally, each element is assigned to the same cluster as its nearest neighbor of higher density. The method has only a single parameter that specifies the region around each observation used to determine the density ρ. The number of clusters is a result of the algorithm. The method considers some observations as outliers and may deliver stretched clusters (there is no restriction on the geometric extent of a cluster).

Lord et al. [10] focus on k-means and k-medoid clustering and discuss *cluster validity indices* (quality of clustering) and *solution stability* (effect of removing part of the data). New *stability indices* related to individual objects, clusters (cluster stability) and *partitions* (global stability) are introduced. The most unstable objects are removed from the dataset, hence introducing the concept of *noise*. Furthermore, the *global stability score* is used to determine the number of clusters. The technique has a large memory footprint and handling our problem on a machine having 8GB of memory would not be possible. Furthermore, noise can be removed but there is no guarantee about limited spatial extent of the resulting clusters.

Knapen and Holmgren [11] use click point clustering in order to select the optimal set of impediments to solve under budget constraints. They combine the same dataset of click points with GPS traces collected from a different set of bicyclists. The authors deliberately allow for stretched clusters and use DBSCAN. Each cluster of click points represents an impediment. The geometric distance between cluster members on one hand and recorded trip positions on the other hand determines whether the impediment affects the trip. An impediment resolution cost is assigned to each cluster and the study aims to determine the set of impediments to resolve under a given budget constraint so that the maximum number of trips becomes impediment free. Both the size and the shape of the clusters affect the resolution cost as well as the geometric relationship between impediments and trips. Hence it would be interesting to investigate how the optimization behaves when DBSCAN is replaced by compact k-means clustering. The current study aims to discover a phenomenon for which spatial density is an essential property. Therefore, we look for spatially dense groups of click points. It should be emphasized that the solitary points are not interpreted as measurement errors; instead they are discarded because they, by definition, are not member of an impediment.

As indicated above, the result of the presented clustering approach aims to support the interactive urban designer through visual inspection of the generated clusters, hence enabling to incorporate the clustering in the decision-making on how to improve the urban road infrastructure.

The remainder of the current article is organized in the following way. Section 2 describes the click point data set used in our study. In Section 3, we discuss the use of traditional k-means clustering for the considered application. In Section 4, we present and analyze our iterative k-means approach. Finally, we conclude the paper in Section 5, with a discussion and concluding remarks.

In our study, we used python 2.7.18 with the machine learning package scikit-learn 0.19.2.

2. Click point data

We used a set of, so-called, click-point data, which was collected by 78 bicyclists traveling in Lund, Sweden, in the autumn of 2018. The bicyclists where instructed to click a handlebar mounted button when feeling unsafe in the traffic situation. Each of the clicks, that is, a data point, contains the unique button identifier, the GPS coordinate (latitude and longitude), the received GPS accuracy, and a time stamp of the click. See Fig 1 for an overview of our click-point data set and geographic focus area.

As the data set contained several repeated (most likely unintended) clicks, duplicates, as well as geographic outliers, we filtered the data set prior to conducting our cluster analysis. Repeated clicks were those that were provided so fast that we considered it to be very unlikely that they could be provided by a bicyclist pushing the button several times. Even if the clicks that we considered to be repeated would be intended, a sequence of repeated clicks most likely refers to the same impediment, still making it safe to filter them out. Duplicates are sequences of data points provided by the same button, but with the same

latitude and longitude. By applying the following, sequential filtering steps, we reduced the data set from 3101 to 1622 data points:

- **Remove repeated clicks**. For each individual, we considered a click point with timestamp t to be an intended click point if and only if it has no predecessor in the period $t - \Delta t$, where $\Delta t = 1$[s]. This reduced our data set from 3101 (the collected number of data points) to 2142 data points.
- **Remove duplicate clicks.** This further reduced our data set from 2142 to 1914 data points.
- **Remove clicks outside focus area.** We filtered all data points outside the area defined by the longitude interval [55.68,55.729] and the latitude interval [13.153,13.254]. This reduced our data set from 1914 to 1774 points.
- **Remove inaccurate clicks.** We filtered all data points with GPS accuracy larger than 50 meters. This further reduced our data set from 1774 to 1622 data points.

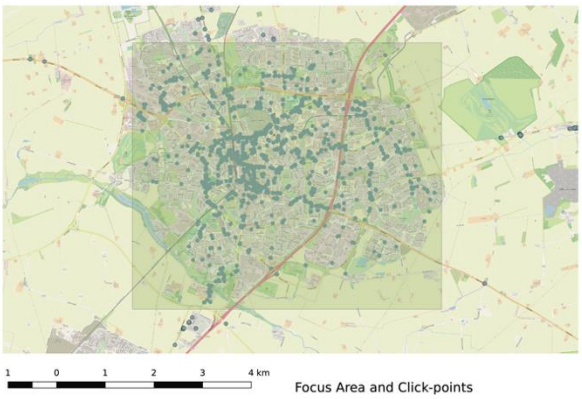

Fig 1: The click-point data set and considered focus area.

3. K-means clustering

In k-means clustering [12], the number of clusters that should be produced is required as input, and the algorithm operates by iteratively assigning points to clusters represented by cluster centroids, which are iteratively updated. A cluster centroid is calculated by taking the average in each dimension of all data points included in the cluster. The algorithm is initiated by assigning k randomly chosen points in the data set as centroids, and it iterates between the two following two steps until the clusters have stabilized:

1. Assign each data point to the closest centroid.
2. Calculate new positions of the centroid of each cluster by taking the mean value of all data points assigned to the cluster.

As mentioned above, k-means clustering requires the number of clusters (k) to be provided as input; however, it is typically not straightforward to identify a suitable value of k. Different approaches have been proposed in order to identify an approximate optimal value of k, including *silhouette analysis*, *rule-of-thumb*, and the *elbow method* [13]. In the elbow method, a decreasing cost function $c(k)$, for example, the average distance from any point in the data set to its centroid, is analyzed for $k = 1, \dots, n$. As the value of k is increased, the cost function will typically drop faster for lower values of k and slower for larger values of k. For some applications, the decrease rate of the cost function will, for some value of k, distinctly slow down, and this value of k is referred to as the elbow point. For our click-point data set, we illustrate in Fig 2 the application of the elbow method, where $c(k)$ is the average distance from any point to its centroid. By visual inspection, the elbow appears to be somewhere in the interval [35,40].

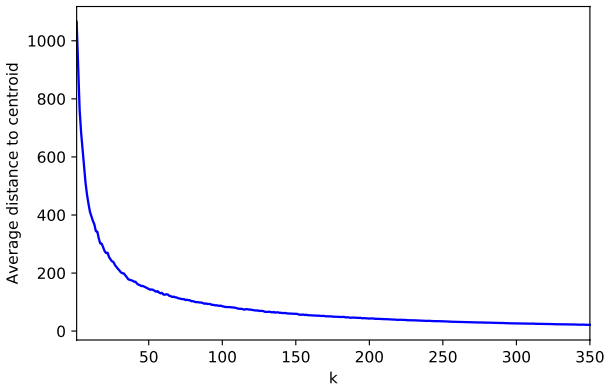

Fig 2: The average distance from any point to its centroid for different values of k. The elbow appears to be somewhere in the interval $[35, 40]$.

However, a further analysis of the k-means clustering for $k = 40$ reveals that the types of clusters produced is clearly not suitable for identification of bicycle impediments, as the clusters vary significantly in terms of their spatial extension. This is illustrated in Fig 3, where we plot the largest and average distance from any point to its centroid for each cluster. In general, the clusters are spatially too big for k:s around 40, and to obtain clusters that are small enough to be connected to bicycle impediments, a significantly higher k-value is required. However, this would instead give us many small clusters containing solitary points, and some clusters that are still spatially larger than a typical bicycle impediment. For $k = 200$, we illustrate this in Fig 4.

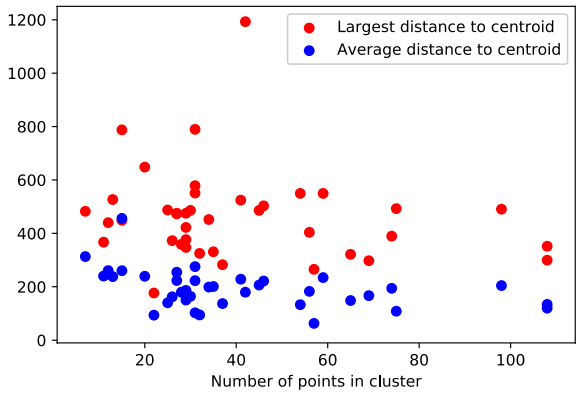

Fig 3: For each k-means cluster, where $k = 40$, the average distance from any point to its centroid (in blue) and the largest distance from any point to its centroid (in red).

By analyzing Fig 3 and Fig 4, it appears that it is mainly the solitary points that prevent us from finding an appropriate value of k. Indeed, k-means clustering has no noise concept similar to DBSCAN and is therefore unable to identify solitary points. In addition, k-means clustering does not allow setting a limit on the minimum number of points to include in a cluster, or a maximum spatial cluster size. These aspects are important in our application, where a minimum number of clicks within a rather small area is required in order to define a bicycle impediment. Furthermore, it is well known that k-means clustering suffers from the problem of converging to a local optimum; hence, it has been suggested to restart k-means using different starting

solutions. For the considered, filtered data set, we applied k-means clustering five times, with random initialization, for each $k = 1, \dots, 350$. For each value of k, we plot in Fig 5, the silhouette score for 1) the clustering with the lowest average distance to centroid (in red), 2) the clustering with the highest average distance to centroid (in green), and 3) the clustering with the median average distance to centroid (in blue). This confirms, for our data set, how sensitive the performance of k-means clustering is for the initialization.

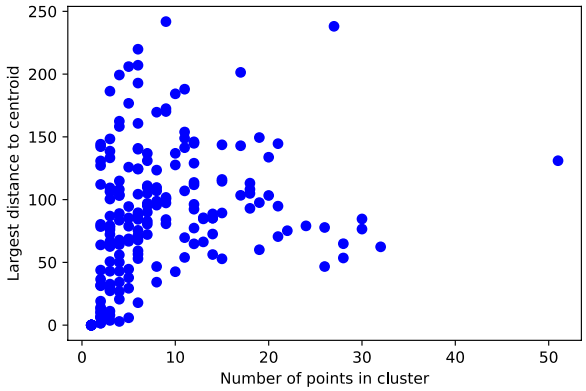

Fig 4. For each k-means cluster, where $k = 200$, the largest distance from any point to its centroid.

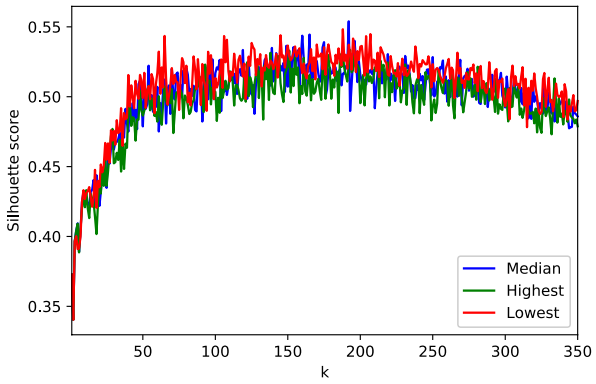

Fig 5: Silhouette score where we applied k-means clustering with five random initializations. For each value of *k*, we plot the silhouette score for the clustering with the lowest, median, and largest average distance from a point to its centroid.

It should be emphasized that the Silhouette score (or Silhouette coefficient**)** [14] is often used in order to evaluate the quality of a clustering. The Silhouette coefficient $s(i)$ of a data point i ranges between -1 and 1, where a higher value indicates that the data point quite well matches its own cluster, whereas a low value indicates that the data point poorly matches its own cluster. We let $a(i)$ denote the mean distance between a point i and all other data points in the same cluster, and $b(i)$ denote the smallest mean distance between i and all points of a cluster, where the minimum is taken over all clusters except the one that i belongs to. The Silhouette coefficient of a point i is defined as

$$s(i) = \begin{cases} \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \text{ if } |C_i| > 1 \\ 0 \text{ if } |C_i| = 1 \end{cases}.$$

The mean $\bar{s}(i)$ taken over all data points in the data set provides an indication of the quality of the generated clustering.

4. Iterative k-means clustering

In the current section, we present our *iterative k-means clustering approach with outlier detection*, which was originally presented by Holmgren et al. [3]. The approach aims to 1) identify an appropriate value of *k* (for the k-means algorithm) and 2) remove solitary points from our data set. The approach, which is specified below (in Algorithm 1), makes use of iterative updates of *k* and removal of solitary points, and it requires the following input parameters:

- min_$samples$. The minimum number of points to allow in any cluster.
- $max\_dist\_to\_centroid$. The maximum distance a point is allowed to be from the centroid of its cluster.
- $outlier\_threshold$. A point is considered to be a solitary point if and only if the distance to its centroid is larger than $outlier\_threshold$ times the median of the distance to the centroid for all points in the same cluster.
- $k^{init} \in \mathbb{Z}^+$. The initial k-value used in the approach.

Furthermore, we let *P* denote the input set of click points.

Algorithm 1: Iterative k-means clustering with outlier detection

Step 0: Set $\boldsymbol{P^{cur}} = \boldsymbol{P}$ and $\boldsymbol{k^{cur}} = \boldsymbol{k^{init}}$.

Step 1: Generate a k-means clustering $\boldsymbol{C}$ for $\boldsymbol{k} = \boldsymbol{k^{cur}}$ and data point set $\boldsymbol{P} = \boldsymbol{P^{cur}}$.

Step 2: Identify the set $\boldsymbol{C'} \in \boldsymbol{C}$ of all clusters with less than $\boldsymbol{min\_samples}$ data points. If $|\boldsymbol{C'}| = \boldsymbol{0}$: Go to Step 3. Otherwise: Go to Step 4.

Step 3: For all clusters $\boldsymbol{c'} \in \boldsymbol{C'}$ remove the point $\boldsymbol{p'}$ with longest distance to its cluster centroid, i.e., set $\boldsymbol{P^{cur}} = \boldsymbol{P^{cur}} \setminus \boldsymbol{p'}$. Set $\boldsymbol{k^{cur}} = \boldsymbol{\max}\,(\boldsymbol{1}, \boldsymbol{k^{cur}} - \boldsymbol{1})$ and go to Step 1.

Step 4: Identify the point $\boldsymbol{p'} \in \boldsymbol{P^{cur}}$, with distance $\boldsymbol{d(p')}$ to its cluster centroid, with largest value $\boldsymbol{q} = \frac{\boldsymbol{d(p')}}{\boldsymbol{\text{median}_{p \in C'}\, d(p)}}$. If $\boldsymbol{q} > \boldsymbol{outlier\_threshold}$: Go to Step 5. Otherwise: Go to Step 6.

Step 5: Remove $\boldsymbol{p'}$, i.e., set $\boldsymbol{P^{cur}} = \boldsymbol{p^{cur}} \setminus \boldsymbol{p'}$. Set $\boldsymbol{k^{cur}} = \boldsymbol{\max}\,(\boldsymbol{1}, \boldsymbol{k^{cur}} - \boldsymbol{1})$ and go to Step 1.

Step 6: If $\boldsymbol{\max_{P^{cur}} d(p)} > \boldsymbol{\max\ \_dist\_to\_centroid}$: Set $\boldsymbol{k^{cur}} = \boldsymbol{k^{cur}} + \boldsymbol{1}$ and go to Step 1. Otherwise: Terminate with approximate optimal $\boldsymbol{k^*} = \boldsymbol{k^{cur}}$, filtered data point set $\boldsymbol{P^{cur}}$, and clustering $\boldsymbol{C}$.

It should be emphasized that the proposed approach contains several (greedy) heuristic elements:

- Step 1: Since k-means clustering is heuristic it cannot be guaranteed that the best clustering can be identified. In addition, choosing the best clustering is not a guarantee to find the optimal point set and optimal k value at termination.
- Step 3: We chose to remove one point from each of the clusters with less than $min\_samples$ data points.
- Step 4: The identification of potential solitary points is based on the chosen definition of what is a solitary point. It should be emphasized that different definitions might give different candidate sets.
- Step 3, 5 and 6: The update of k^{cur} is heuristic, based on our perception of what is reasonable in order to converge to a proper k value at termination.
- The choice of input parameters also influences the performance of our iterative approach.

Considering the indication (see Section 3) that the optimal k value should be somewhere in the interval [35,40], we applied our iterative kmeans approach with initial k value slightly below this interval. In particular, we applied our algorithm with $min\_samples = 8$, $max\_dist\_to\_centroid = 50$, $k^{init} = 30$, and $outlier\_threshold = 2.0$. This resulted in a final k value of 41 (i.e., $k^* = 41$), where 1053 (i.e., all but 569) of our 1622 data points were discarded as solitary points. See Fig 6 for an illustration of the generated clusters.

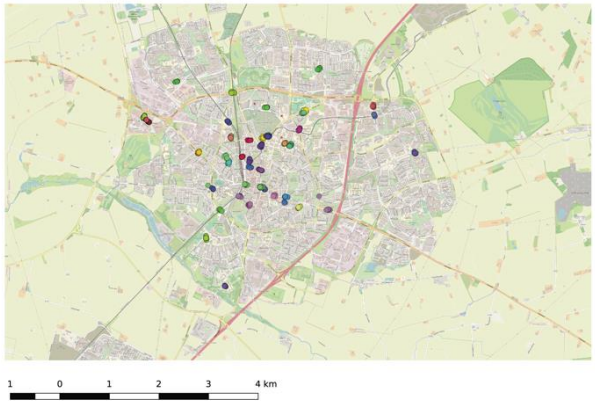

Fig 6: Generated clusters using our iterative k-means clustering approach with $k = 41$ and 1053 solitary points removed.

For comparison, we present in Fig 7 the clusters generated with the DBSCAN algorithm [15] for three different values of ϵ. By comparing Fig 6 and Fig 7, it can be seen that the DBSCAN clusters typically cover the k-means clusters; however, the k-means clusters are in general more compact. The reader is referred to Holmgren et al. [3] for further details of the DBSCAN analysis.

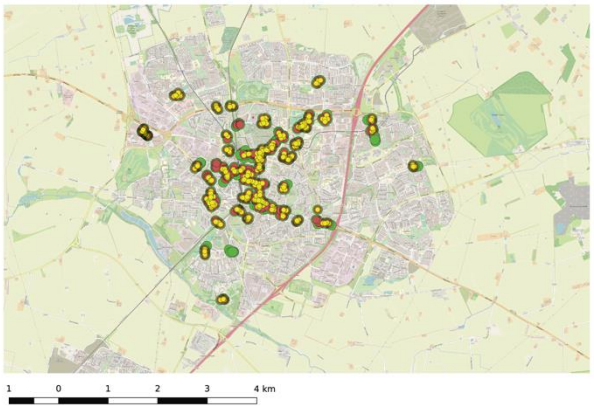

Fig 7: DBSCAN clusters using $\epsilon = 53$ (yellow), $\epsilon = 63$ (red), and $\epsilon = 73$ (green).

It should be emphasized that we evaluated the generated clusters by presenting them to traffic management practitioners of the Lund municipality, who confirmed that the identified bicycle impediments, represented by clusters, appear to be reasonable. They were able to explain several of the impediments, whereas some of them were previously unknown.

In order to evaluate the robustness of our iterative k-means approach, we applied it for $k^{init} \in \{5,10,\dots,100\}$, with $max\_dist\_to\_centroid = 50$, $min\_samples = 8$, and $outlier\_threshold = 2.0$. For each value of k^{init}, we conducted five runs, were we used the kmeans++ initialization approach [16] in each of the iterations. It should be mentioned that the *kmeans++* initialization algorithm, which is the default initialization method in scikit-learn, has been suggested in order to propose an initialization that increases the possibility that k-means will find a good local optimum. It should, however, be emphasized that kmeans++ is stochastic.

The results of our robustness analysis clearly indicate that our iterative approach inherits the issue that the k-means algorithm tends to get stuck in different local optima, depending on how the initialization is done. In Fig 8, we plot the final k values for each of our five series of runs with $k^{init} \in \{5,10,\dots,100\}$. For k^{init} values up to around 15-20, it appears that the final value of k is lower than for higher values of k^{init}. Similarly, we plot in Fig 9, the number of solitary points for each value of k^{init} for our five series of runs. Obviously, there is a negative correlation between the final k value and the number of solitary points. Furthermore, we present in Fig 10, the silhouette score for each value of k^{init} for our five series of runs.

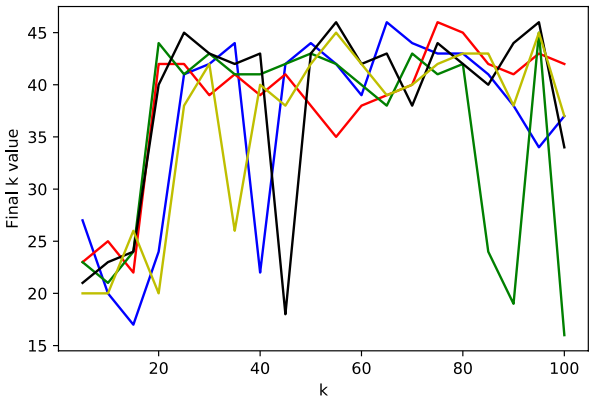

Fig 8: Final k value for $k^{init} \in \{5, 10, \dots, 100\}$ and the kmeans++ initialization method, where we conducted five runs of our iterative k-means approach for each value of k^{init}. Each color represents one series of runs.

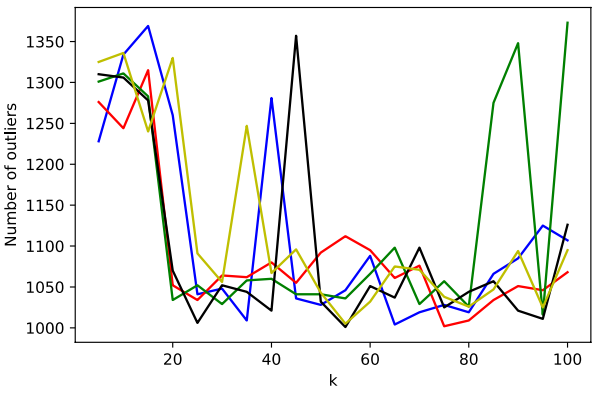

Fig 9: Number of solitary points for $k^{init} \in \{5, 10, \dots, 100\}$ and the kmeans++ initialization method, where we conducted five runs of our iterative k-means approach for each value of k^{init}. Each color represents one series of runs.

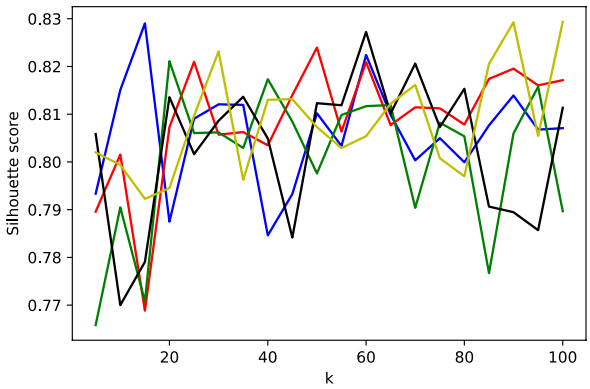

Fig 10: Silhouette score for $k^{init} \in \{5, 10, \dots, 100\}$ and the kmeans++ initialization method, where we conducted five runs of our iterative k-means approach for each value of k^{init}. Each color represents one series of runs.

It can be seen that the Silhouette score for the final clusterings also vary significantly; however, it can be argued that a higher silhouette score indicates a better clustering. Hence, we plot for each value of k^{init} in Fig 11 and Fig 12 the final k value and number of solitary points, respectively, for the runs with the highest Silhouette score. This results clearly indicate that the final k values and number of generated solitary points stabilize if we choose the clusterings with higher silhouette score. In turn this could indicate that a proper final k value, for the considered application with solitary points removed, is approximately in the range [40,45], a value that can be reached for values of k^{init} ranging from 20 at least up to 100. This also indicates that the silhouette score might be a proper indication of how well our iterative k-means approach manages to identify solitary points, and to identify a proper k value.

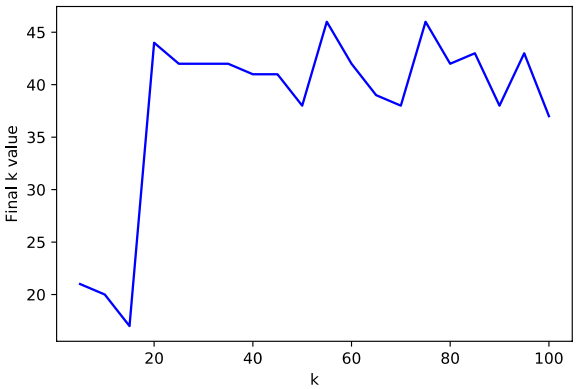

Fig 11: Final k value for $k^{init} \in \{5, 10, \dots, 100\}$ and the kmeans++ initialization method, for the runs with best silhouette score.

For the interested reader, we present in Fig 13 the evolution of k for two runs of our iterative k-means clustering approach, where $k^{init} = 40$. It can be seen that both of the runs converge towards the same final k value even though the evolution of k is different.

Furthermore, we present in Table 1 the adjusted Rand index for pairwise comparisons of the clusterings generated for the runs with best silhouette score for $k^{init} \in \{30,35,40,45,50\}$. The adjusted Rand index [17] is a measure that can be used in order to compare the partitions (i.e., clusters) included in two clusterings. Basically, the adjusted Rand index gives a measure of the degree of agreements between two clusterings.

As the set of solitary points varies for the different runs (see Fig 12), we had to include the solitary points in the adjusted Rand score calculation. In particular, we assigned all solitary points for each of the runs to a separate cluster.

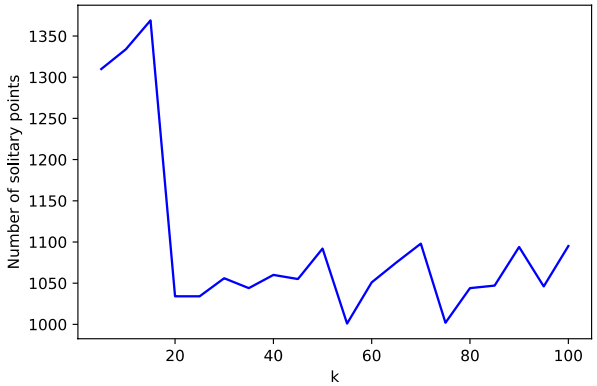

Fig 12: Number of identified solitary points for $k^{init} \in \{5, 10, \dots, 100\}$ and the kmeans++ initialization method, for the runs with best silhouette score.

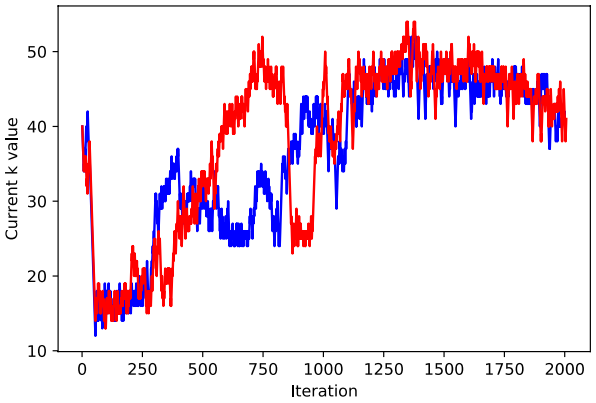

Fig 13: Evolution of k for two runs, represented in blue and red, respectively, of our iterative k-means clustering approach, where $k^{init} = 40$.

Table 1: Adjusted Rand index for $k^{init} \in \{30, 35, 40, 45, 50\}$, for the runs with best silhouette score.

k^{init}	35	40	45	50
30	0.862	0.896	0.914	0.878
35		0.858	0.892	0.837
40			0.879	0.871
45				0.844

5. Discussion and concluding remarks

In the current paper, which is an extension of Holmgren et al. [3] we have presented a clustering analysis for the identification of so-called bicycle impediments, which are locations in the traffic network where bicyclists tend to feel unsafe in the traffic situation. We present an iterative k-means clustering approach with outlier detection, which is tailored for the considered application. In particular, we extend the work by Holmgren et al. [3] with an in-depth analysis of our iterative k-means clustering approach.

The presented iterative k-means approach is based on repeated application of the k-means algorithm, where it aims to 1)

identify a proper k value and 2) identify solitary points (including outliers). Hence, it aims to address two of the main weaknesses of the k-means algorithm, that is, that it requires that a proper k value is provided as input and that it is not able to identify outliers in the considered data set.

As mentioned above, it is well known that the k-means clustering algorithm is sensitive towards the choice of starting solution, that is, the initial choice of cluster centroids, and it typically converges towards a local optimum. In order to analyze whether the characteristic that k-means tends to converge towards a local optimum has an influence of our iterative k-means clustering approach, we conducted a robustness analysis consisting of five series of runs with our approach. In each of the series, we considered k^{init} values ranging from 5 to 100, that is, $k^{init} \in \{5,10,\dots,100\}$. The results from our robustness analysis clearly indicate that our iterative approach inherits the sensitiveness towards the initialization from the standard k-means algorithm.

However, our results indicate that the Silhouette score (even though it is proposed for visual analysis of clusterings, can be used as an indicator of the quality of a clustering. By choosing the clustering with the highest Silhouette score for each of the runs for a particular value of k^{init} (at least if $k^{init} \geq 20$), it appears that the outcome of the algorithm stabilizes (see Fig 11). Hence, we recommend, just as for standard k-means clustering, to run our iterative clustering approach multiple times and choose the clustering with highest Silhoutte score.

Our results further indicate that our iterative approach is not very dependent on choosing a proper initial k value (i.e., k^{init}). For our data set, where it appears that a proper final k value approximately lies in the range $[40,45]$, it appears that k^{init} ranging from 20 up to at least 100 (we did not test values higher than 100) gives reasonable results.

In order to compare the generated clusterings for the runs with best silhouette score, we generated the adjusted Rand index for pairwise comparison of the runs with $k^{init} \in \{30,35,40,45,50\}$. The adjusted Rand index values for these comparisons vary between 0.837 and 0.914, which we consider to be a rather strong indication that the clusterings generated for different values of k^{init} are rather consistent.

As mentioned above, the Rand index compares partitions of a dataset (clustering results). Hence, it defines a value for each pair of results. A graph can be created where each vertex represents a clustering result and the edges are labeled with the Rand index. Two vertices are connected if and only if the Rand index for the pair exceeds a given threshold. Cliques in such a graph may identify *families* of more or less equivalent clusterings. In many problems there is no single best clustering representing the *base truth*; however, there may be a family of similar clusterings that solve the impediment identification problem in a sufficient way. This is suggested by the diagrams because of the plateaus for k and for the number of solitary points. Future research will focus on identifying maximal cliques having sufficiently large average silhouette value in the *Rand index based graph* and on summarizing the resulting cluster family for use by the officer in charge for impediment resolution.

An interesting observation for the considered data set is that the elbow method, applied on the data set including the solitary points, suggested that the optimal value k is approximately in the range $[30,40]$. Our iterative approach suggests that an appropriate k value is just slightly higher, that is, in the range $[40,45]$. Perhaps this could be an indication that the elbow method could be used on the original data set in order to find an appropriate value of k^{init} for our iterative approach.

As mentioned above, our iterative approach contains some heuristic steps, which obviously might influence the performance of the approach. For example, we use kmeans++ in Step 2; however, kmeans++ does not always provide the best clustering, as it is to some extent stochastic. In order to analyze whether finding better clusterings in each of the iterations tends to lead to a better final clustering, we made, in addition to our first kmeans++ series of runs, three additional series of experiments, where we added 10, 20, and 30, random initializations, respectively in each of the iterations of our approach. Then we chose to use the clustering with the lowest average distance from each point to its centroid, assuming that the lowest average distance is a proper measure of the clustering quality. One could believe that choosing the best clustering in each of the iterations should lead to a better final clustering; however, the results of our experiments does not support this. For the details, see Fig 14 and Fig 15, where we plot the final k value and number of iterations where a random initialization is better than kmeans++, respectively, for each of the considered k^{init} values for our four runs.

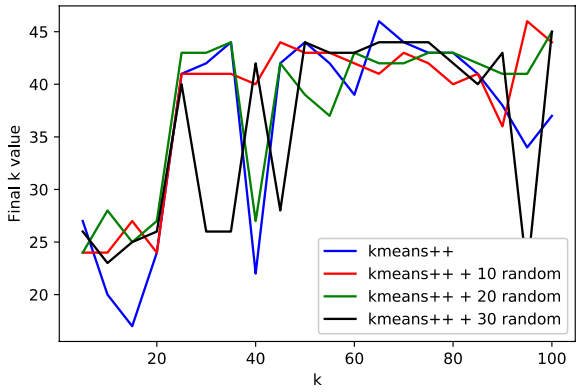

Fig 14: Final k value when only using kmeans++, and when adding 10, 20, and 30 random initializations, respectively, in each of the iterations of our iterative approach.

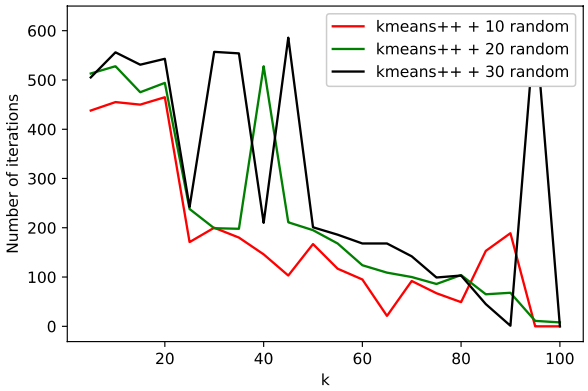

Fig 15: Number of iterations (of approximately 2000) where a random initialization gives a better clustering than kmeans++, when adding 10, 20, and 30 random initializations, respectively, in each of the iterations of our iterative approach.

References

[1] Andersen L, Riiser A, Rutter H, Goenka S, Nordengen S, Solbraa A. Trends in cycling and cycle related injuries and a calculation of prevented morbidity and mortality. Journal of Transport & Health 2018;9:217-225. https://doi.org/10.1016/j.jth.2018.02.009

[2] European Union. Traffic Safety Basic Facts 2018. 2018.

[3] Holmgren J, Knapen L, Olsson V, Persson Masud A. On the use of clustering analysis for identification of unsafe places in an urban traffic network. Proceedings of the 11th International Conference on Ambient Systems, Networks and Technologies (ANT). Procedia Computer Science 2020;170;187-194. https://doi.org/10.1016/j.procs.2020.03.024

[4] Cheng, W, Washington, SP. Experimental evaluation of hotspot identification methods. Accident Analysis & Prevention 2005;37;870-881. https://doi.org/10.1016/j.aap.2005.04.015

[5] Montella A. A comparative analysis of hotspot identification methods, Accident Analysis & Prevention 2010;42(2); 571-581. https://doi.org/10.1016/j.aap.2009.09.025

[6] Anderson TK. Kernel density estimation and K-means clustering to profile road accident hotspots}. Accident Analysis & Prevention 2009;41(3);359-364. https://doi.org/10.1016/j.aap.2008.12.014

[7] Xu Q, Tao G. Traffic Accident Hotspots Identification Based on Clustering Ensemble Model. Proceedings of the 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)}, 2018, pp 1-4. https://doi.org/10.1109/CSCloud/EdgeCom.2018.00010

[8] Person Masud A, Olsson V. Cyclists' perceived insecurity in urban environment - An unsupervised machine learning study. Bachelors Thesis. Department of Computer Science and Media Technology, Malmö University; 2019.

[9] Rodriguez A, Laio A, Clustering by fast search and find of density peaks, Science 2014;344;1492-1496. https://doi.org/10.1126/science.1242072

[10] Lord E, Willems M, Lapointe FJ Makarenkov V. Using the stability of objects to determine the number of clusters in datasets. Information Sciences 2017;393;29-46. https://doi.org/10.1016/j.ins.2017.02.010

[11] Knapen L, Holmgren J. Identifying bicycle trip impediments by data fusion. Proceedings of the 11th International Conference on Ambient Systems, Networks and Technologies (ANT). Procedia Computer Science 2020;170;195-202. https://doi.org/10.1016/j.procs.2020.03.025

[12] MacQueen J. Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. 1967, pp 281-297

[13] Kodinariya T. Makwana PR. Review on Determining of Cluster in K-means Clustering. International Journal of Advance Research in Computer Science and Management Studies 2013;1;90-95.

[14] Rousseeuw, PJ. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics. 1987;20;53-65. https://doi.org/10.1016/0377-0427(87)90125-7

[15] Ester M, Kriegel HP, Sander J, Xu X. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. Portland, OR, 1996.

[16] Arthur D, Vassilvitskii S. k-means++: the advantages of careful seeding". Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms. Philadelphia, PA, 2007.

[17] Hubert L, Arabie P. Comparing partitions. Journal of Classification. 1985;2(1);193–218. https://doi.org/10.1007/BF01908075