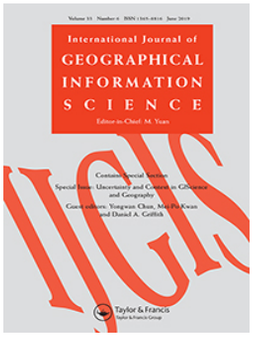

International Journal of Geographical Information Science

ISSN: 1365-8816 (Print) 1362-3087 (Online) Journal homepage: https://www.tandfonline.com/loi/tgis20

Exploring the uncertainty of activity zone detection using digital footprints with multi-scaled DBSCAN

Xinyi Liu, Qunying Huang & Song Gao

To cite this article: Xinyi Liu, Qunying Huang & Song Gao (2019) Exploring the uncertainty of activity zone detection using digital footprints with multi-scaled DBSCAN, International Journal of Geographical Information Science, 33:6, 1196-1223, DOI: 10.1080/13658816.2018.1563301

To link to this article: https://doi.org/10.1080/13658816.2018.1563301

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE
2019, VOL. 33, NO. 6, 1196–1223
https://doi.org/10.1080/13658816.2018.1563301

Taylor & Francis
Taylor & Francis Group

RESEARCH ARTICLE

Exploring the uncertainty of activity zone detection using digital footprints with multi-scaled DBSCAN

Xinyi Liu, Qunying Huang and Song Gao

Department of Geography, University of Wisconsin-Madison, Madison, WI, USA

ABSTRACT

The density-based spatial clustering of applications with noise (DBSCAN) method is often used to identify individual activity clusters (i.e., zones) using digital footprints captured from social networks. However, DBSCAN is sensitive to the two parameters, *eps* and *minpts*. This paper introduces an improved density-based clustering algorithm, Multi-Scaled DBSCAN (M-DBSCAN), to mitigate the detection uncertainty of clusters produced by DBSCAN at different scales of density and cluster size. M-DBSCAN iteratively calibrates suitable local *eps* and *minpts* values instead of using one global parameter setting as DBSCAN for detecting clusters of varying densities, and proves to be effective for detecting potential activity zones. Besides, M-DBSCAN can significantly reduce the noise ratio by identifying all points capturing the activities performed in each zone. Using the historic geo-tagged tweets of users in Washington, D.C. and in Madison, Wisconsin, the results reveal that: 1) M-DBSCAN can capture dispersed clusters with low density of points, and therefore detecting more activity zones for each user; 2) A value of 40 m or higher should be used for *eps* to reduce the possibility of collapsing distinctive activity zones; and 3) A value between 200 and 300 m is recommended for *eps* while using DBSCAN for detecting activity zones.

ARTICLE HISTORY
Received 2 May 2018
Accepted 14 December 2018

KEYWORDS
Activity zones; Twitter; spatial clustering; social networks; human mobility

1. Introduction

Human mobility study is a significant research thrust in GIScience by contributing to various applications, such as examining human behaviors and mobility patterns, revealing underlying urban spatial structure and dynamics, or understanding the evolution of epidemics and spatial spread of diseases (Kang *et al*. 2012, Noulas *et al*. 2012a, Kwan 2013, Richardson *et al*. 2013, Gao 2015, Xu *et al*. 2016, Huang 2017). While surveys, GPS data, and mobile phone records have been primarily used to explore human movement behaviors and patterns, social media now is widely used as a new source that captures human regular activity patterns (e.g. commuting time) and population dynamics (Gao *et al*. 2014, Huang and Wong 2015, Steiger *et al*. 2015, Luo *et al*. 2016). However, digital footprints recorded by social media and represented as a series of spatiotemporal (ST) points are highly sparse and irregular in both spatial and temporal dimensions, and thus various spatial data processing and data mining methods have to be applied before

CONTACT Qunying Huang qhuang46@wisc.edu

meaningful mobility patterns can be discovered. Herein, spatial clustering is one of the important methods to make sense of digital footprints by grouping the sparse and irregular ST points to clusters (Mennis and Guo 2009). During the work of modeling, visualizing and mining individual digital footprints captured through social media, it is a paramount preprocessing component for various data mining algorithms and models, such as home location reference and user future location prediction models (Mahmud *et al*. 2012, Mathew *et al*. 2012, Lin and Cromley 2018), to discover the regular activity zones and points of interest (POIs) that an individual or a group of users with similar movement behaviors regularly visit or stay.

Among all spatial clustering algorithms, the density-based spatial clustering of applications with noise (DBSCAN; Ester *et al*. 1996) is very popular with the capability of detecting clusters of arbitrary shape with noise, and only needs to supply two input parameters: *minpts*, and *eps*; where *eps* is the search radius of a point, i.e. the neighborhood of a point, and *minpts* is the minimum number of points that the neighborhood should include to form a cluster (Moreira *et al*. 2013). In DBSCAN, points with a neighborhood more than *minpts* are categorized as core points, and points reachable by a core point are border points of the cluster. Points not within the *eps* search radius of any core point are treated as noise. Both core points and border points are considered as the clustered points.

At present, a few studies have provided some general guidelines on setting up a *minpts* and an *eps* value for various datasets (Ester *et al*. 1996, Zhou *et al*. 2004). For example, Ester *et al*. (1996) reveal that using a *minpts* value <4 may capture noisy points (outliers) in clusters and the clustering results most likely are similar with a *minpts* value ≥4. While clustering dense GPS trajectories, Zhou *et al*. (2004) use an *eps* value of 20 meters (m), approximating to the uncertainty in GPS positioning. With regard to geo-tagged social media data clustering analysis, Hu *et al*. (2015) compare the DBSCAN results with 25 different parameter combinations and demonstrate that *eps = 200 m* search radius can help identify the well-known urban areas of interest neighborhoods in cities using geo-tagged Flickr photos. However, the activity zone and trajectory detection is highly uncertain and sensitive to the DBSCAN parameter values with sparse trajectory points collected from social networks: while the clustering results produced by smaller *minpts* and/or *eps* values likely collapse footprint points in one activity zone into smaller clusters and capture clusters of random activities, the derived trajectories are likely more precise in space; the results generated by larger *minpts* and/or *eps* values include fewer but larger clusters, more likely merging footprint points in different activity zones as one cluster. Correspondingly, the resulting trajectories are likely less precise in space.

In addition, while DBSCAN is able to identify clusters of arbitrary shapes, it is insufficient to deal with data with clusters of different densities (or neighborhoods) as it fails to detect the core points of varying density clusters with a single *eps* value (Ertöz *et al*. 2003). For example, the points in the bottom left cluster (Figure 1) are more compacted or denser in space and with 25 as *eps* value, these points are grouped within one cluster. However, the points in the right-side cluster are sparser and the *eps* value should be increased to 76 to ensure these points are all included in the cluster. Further, if we increase the *eps* value to 90, the two distinct clusters will be merged as one cluster. At the same time, people's travel trajectory densities captured through social media over different places differ based on the size of a place and the nature of their activities in the

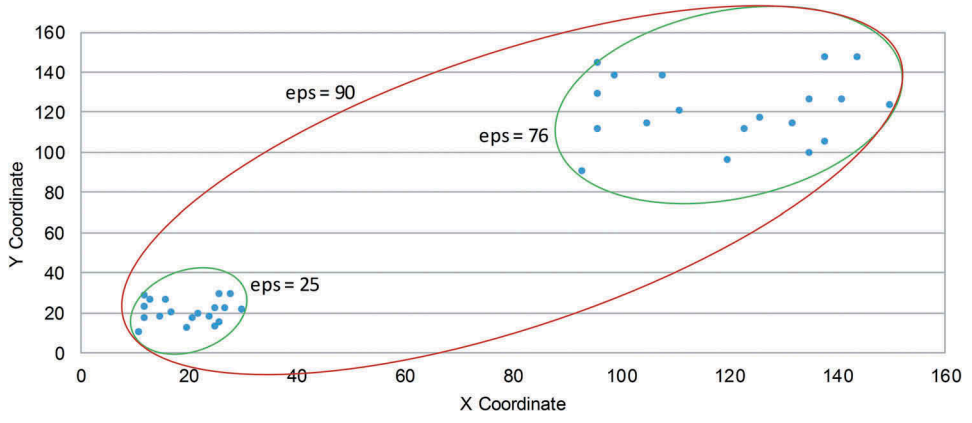

Figure 1. Randomly generated two clusters with different densities.

place. For example, people may have a relatively compact digital footprints at their work place such as their office, and dispersed footprints in a large shopping mall. What is more, the two locations (office building and shopping mall) could be very close to each other as the example (Figure 1). Therefore, an ideal clustering algorithm for digital footprints should not only be able to detect, but also separate different clusters with varying densities.

To mitigate the uncertainty of the clustering results produced by DBSCAN for digital footprints of human movements, this paper develops an improved density-based algorithm based on DBSCAN, named as Multi-scaled DBSCAN (M-DBSCAN) to detected activity clusters at different scales of density and cluster size. Given the historic geo-tagged tweets of an individual, M-DBSCAN can automatically produce a set of clusters and detect appropriate local *eps* and *minpts* values, instead of using one global parameter setting (i.e. the same *eps* and *minpts* value) for each cluster indicating the activity location (zone) the individual visits frequently. The experimental studies demonstrate that the proposed M-DBSCAN algorithm outperforms the classic DBSCAN method and its improved variation, varying DBSCAN (VDBSCA; Liu *et al*. 2007), for activity zone detection from individual geo-tagged tweets with varying-density distributions. Especially, while DBSCAN with inappropriate *eps* and *minpt* values may treat some points (i.e. tweets) in the same activity zone as noise (e.g. tweets posted at the parking lot or pool area of an apartment; Figure 4), our algorithm can significantly reduce the noise ratio (the proportion of tweets not included in any cluster) by identifying all points capturing the activities performed in each zone. Finally, using the proposed M-DBSCAN algorithm on different geo-tagged tweets produced by a large number of users in Washington, DC (DC), this paper explores the range and distribution patterns of appropriate parameter (*eps* and *minpts*) settings for activity zone detection of individuals.

In the next section, we introduce relevant work of using social media data for human mobility studies and state-of-the-art work on DBSCAN for spatial data clustering. Section 3 discusses the uncertainty during the process of regular activity zone detection. Section 4 introduces the datasets and Section 5 presents the M-DBSCAN algorithm and demonstrates how the proposed algorithm can effectively detect activity zones of varying densities. Section 5 compares the results of M-DBSCAN and DBSCAN to develop a general guideline

to select the values of DBSCAN parameters for highly varying densities of digital footprints. In the conclusion, we summarize and provide insights about the effectiveness of the M-DBSCAN, and a recommendation of selecting *eps* value for DBSCAN.

2. Related work

Social media recently emerges as a new data source for examining human movement behaviors and mobility patterns. This new trend is driven by the fact that social media data have several unique advantages (Huang and Wong 2016, p. 1) a large number of study subjects. For example, Twitter data have recorded surprising snapshots of human daily movements at a large scale (e.g. 500 million registered users publishing 400 million tweets per day at the year of 2013; Morstatter *et al*. 2013, p. 2) publicly accessible. Using the Twitter application programming interfaces (APIs), we are allowed to retrieve up to 1% of Twitter data (Morstatter *et al*. 2013); and 3) long-term trajectories are available. These data are generated continuously, reflecting the on-going societal situations. They cover extensive temporal scales and provide near real-time information.

As a result, social media is largely used in various human mobility studies (Huang and Wong 2015, Steiger *et al*. 2015, Luo *et al*. 2016, Shaw *et al*. 2016). However, the nature of social media imbues its data with certain blind spots. For example, such data rarely offer much information on the background (e.g. socioeconomic status, demographic information, or home location) of users. This triggers another research line focusing on social media data mining that intends to infer the background of social media users. In those studies, an important step is detecting the regular activity regions using spatial clustering method that a social media user frequently visits. Among various spatial clustering algorithms, DBSCAN (Ester *et al*. 1996) is often used to detect clusters from digital footprints. However, while we do not need to supply the number of clusters before clustering begins like the K-means method, DBSCAN still requires *minpts* and *eps* values to be provided. While previous studies use a value of 4 and 20 for *minpts* and *eps* respectively (Ester *et al*. 1996, Zhou *et al*. 2004, Hu *et al*. 2015), whether they are optimal for sparse digital footprints are unknown.

In addition, onefold DBSCAN fails to identify clusters from datasets of varying density. To overcome such a limitation, Liu *et al*. (2007) introduced a new algorithm named VDBSCAN to detect clusters with varied densities using the following steps:

(1) Given a *k* value, compute the distance of each point to its kth nearest neighbor (k-dist), and generate a k-dist plot for all points based on their k-dist value sorted ascendingly (Figure 3).
(2) Determine the number of densities intuitively based on the k-dist plot.
(3) Select parameter *eps* for each density.
(4) Scan all the points and cluster points of varied densities with corresponding *eps* using DBSCAN.
(5) Finally, display the valid clusters corresponding with varied densities.

However, the value of parameter *k* has to be manually supplied and no logical procedure or principle is used for determining its appropriate value. As a result, Chowdhury *et al*. (2010) developed a new method to detect the value of parameter

k automatically based on the characteristics of the datasets. Specifically, given a set of points (P_i), six steps are used to calculate the parameter *k*:

(1) For each point P_i, calculate average distance d(P_i) from P_i to all other points.
(2) Compute the average of all d(P_i) as avg(d).
(3) For each P_i in the datasets, draw a circle with P_i as the center and avg(d) as the radius.
(4) For every circle, find the point nearest to the circumference of each circle, and tag this point as T_i.
(5) Identify the position of T_i as T_i(Pos) relative to the P_i for that particular circle.
(6) Determine the mode of T_i(Pos) and use this value as the value of parameter *k* in the k-dist plot.

While the method works well on a small number of points that are relatively close with each other (e.g. Figure 1), our experiment results indicate that it cannot identify a reasonable *k* value for digital footprints. On one hand, such data could be geographically distributed in a large area, resulting in a large average distance d(P_i) and overall average distance avg(d) at the first and second steps, respectively. As a result, a very large *k* would be detected. On the other hand, trajectory data could be generated at several places with each place including a set of points very close or even overlapping with each other, and accordingly a very small *k* value may be identified. However, the number of points (cluster size) in each activity zone are highly different. In fact, it is observed that most of the users have a primary cluster including a large number of the digital footprint points (46.4% of clustered points on average), and several relatively smaller clusters (Section 6.1). While a large *k* is expected for detecting the primary cluster, relatively small *k* values should be used for the remaining clusters. Therefore, we argue that different local *k* values instead of a global value should be used for detecting clusters with points of varying densities and multiple scales, and heterogeneous spatial distribution. In addition, many existing methods (Liu *et al*. 2007, Chowdhury *et al*. 2010) manually identify *eps* value from the k-dist plot, which decreases *eps* accuracy and clustering efficiency (Parvez 2012).

A few multi-level or hierarchical methods have been proposed to produce clusters with varying densities. To represent the density-based hierarchical clustering structure of a data set, OPTICS (Ankerst *et al*. 1999) constructs a plot of reachability distance, which in turn can be used to extract clusters of different density accordingly. However, OPTICS does not explicitly produce clusters. Wang *et al*. (2016) generated two different *minpts* values by analyzing an adjacency list graph and found more meaningful clusters of varied densities on the sample data set. However, this algorithm still needs to supply *eps* value. Karami and Johansson (2014) combined Binary Differential Evolution and DBSCAN algorithm to automatically tune the best combinations of *eps* and *minpts* for varying data distributions. Campello *et al*. (2013) pointed out the interrelationship between outliers and clusters, and proposed a hierarchical density estimation method for outlier score calculation, which unified the detection process of both local and global outliers. The hierarchical clustering method first computes a hierarchy of mutual reachability distances to represent the distance between a pair of objects, and detects connected components and noise objects at each hierarchical level. Next, an improved hierarchical

clustering method was further proposed to simplify the representation of clustering hierarchy for detecting the most significant clusters with different densities by maximizing their overall stability (Campello *et al*. 2015). These three studies were evaluated and achieved good classification results with general datasets. However, these algorithms are very complicated to generate density estimates and structure (e.g. hierarchy), and their performance and applicability on large-scale and complex datasets is unknown.

To address the limitations of existing work of using a density-based approach for digital footprints, and fully automate the clustering process, we develop an M-DBSCAN algorithm (Algorithm 1 and Algorithm 2 in Section 5). It can aggregate spatial points derived from geo-tagged social media data, into clusters of varying spatial distribution and densities with minimal user inputs.

3. Uncertainty of detecting activity zones using digital footprints

In this work, we define an activity zone as the activity region or area that an individual frequently appears or visits. In other words, activity zones should reflect actual human activity space and include frequent (or regular) activities. This section discusses the potential sources of uncertainty introduced in the process of detecting regular activities from four aspects, including 1) data sources, 2) methods for identifying activity zones, 3) representation of the activity zones, and 4) inference of activity zone type. While this paper focuses on addressing the uncertainty posed by methods for detecting activity zones, we elaborate each source as below.

3.1. *Uncertainty from data sources*

The digital footprints are sparse in nature and include a relatively small number of points. Footprint points are captured only when users post a message on the social media platform, and also enable the location-tagging service. As such, compared to spatial trajectory data collected through GPS devices or travel diary establishing the movement trajectories with great detail, using social media data needs to handle 'hit or miss' situations (Huang and Wong 2016). For example, 'check-in' data captured from social media (e.g. Foursquare) may not reveal individual's regular movements and trajectories of individuals. This is because many users are less likely to check in at the regular activity locations (e.g. work), but more likely to check in the places for random activities, such as dining and entertainment (Noulas *et al*. 2012b, Liu *et al*. 2015).

Other social media platforms for people sharing personal experiences and thoughts, such as Twitter and Facebook, can automatically include location to each message as a geo-tag once precise location service is enabled. Accordingly, digital footprints recorded by these platforms have potential to capture human daily movements more comprehensively (Liu *et al*. 2015). Still, people use such platforms more likely in certain places (e.g. metro/bus stations) and time periods (e.g. non-working time during lunch or in the evening), contributing to the spatial and temporal biases of the data (Huang and Wong 2016). In other words, digital footprints might not capture daily regular activity zones (e.g. work), and include irregular activities (e.g. dinning at certain places) instead, introducing noise or uncertainty.

Additionally, location position accuracy and uncertainty are always a part of the conversion when leveraging digital footprints for activity pattern studies. To attach additional location context to a message, users can add a general location (e.g. 'Madison, Wisconsin'), a specific business, landmark, or other POI, or precise location of the mobile device while posting the message. A general location (e.g. city's location), or a specific POI near to a user's activity can mask the true location of the message – as a proxy of user's activity location. Additionally, the precise locations of social media messages (e.g. geo-tagged tweets) are primarily acquired through three methods: GPS, cell tower triangulation, and WiFi hotspots. Different devices with varying positioning methods have its own levels of precision. For example, Zandbergen (2009)'s work reveals that the location of an iPhone 3 achieved an average accuracy of 8 m with GPS, 74 m with WiFi and 600 m with cell tower positioning. As such, location accuracy could be varying across messages, and the detected activity zones may not accurately reflect the areas the user frequently visits.

3.2. *Uncertainty from activity zone detection method*

Digital footprints are recorded as a series of ST points that represent the varying locations of an individual has visited. The locations are resulted from both the regular and random visits. To detect the regular activity zones of an individual, spatial clustering analysis is applied to detect the places that an individual's regular activities take place, and identify outliers (noise) associated with irregular activities. Three major uncertainties could be introduced in this process, whereas the first uncertainty is influenced by the choice of clustering algorithm. Using different clustering algorithms, the number and the shape of clusters are highly varying. Additionally, clustering algorithms usually need to provide one or more input parameters greatly impacting the clustering process and produced results (Moreira *et al*. 2013). Therefore, even using the same clustering algorithm, the results are most likely different depending on the parameterizations of the algorithm (e.g. different *eps* and *minpts* values for the DBSCAN; Figure 4).

The second uncertainty results from the determination of the membership of footprint points while grouping them into different clusters, which are far less than straightforward and objective. For example, points A-F (Figure 4(f)) would be normally and ideally classified as outliers since they are relatively far away from other points in the cluster. However, based on the geographic environment interpreted from the Google Earth map, these activities captured through the online platform are performed in neighborhood of an apartment: points B-D are located at the pool associated with the apartment community, F is recorded at the parking lot, and finally A and E probably are on the entry to the apartment. Therefore, we can reasonably argue that these points are part of the activity zone of the individual, and a spatial clustering algorithm should capture these points instead of treating them as noise. Especially, many users only have a small amount of ST points recorded, the portion of noisy points should be kept minimal to discover meaningful human mobility patterns. On the contrary, while sometimes it is ideal to aggregate 'outlier' (or noise) points into a cluster, we may also need to collapse distinctive clusters capturing different types of activities and assign different clustering memberships to each point. For example, activities performed within two spatially close buildings but with different functions (e.g. work and eating) should be

ideally separated into two zones to examine an individual's mobility patterns. As such, the evaluation of a spatial clustering method is rather complex, and it should depend on how we define the noise and how we define the membership of a point in clusters among the trajectory points.

The third uncertainty is introduced during the process of discriminating a regular activity cluster from a random one among the clusters with a wide range of point sizes detected in each cluster. A common approach defines the minimal number of points (*mpts*) as a threshold to identify regular activity places and remove random activity places (Huang and Wong 2016). Specially, a cluster with the number of points < *mpts* are treated as noise and thus discarded. It is worth noting that *mpts* is different from *minpts* in the DBSCAN algorithm. While *minpts* is used during the clustering process to differentiate core points, boarder points, and noise of clusters, *mpts* removes random activity zones after clustering. A lower *mpts* value captures more clusters as the regular activity zones, and therefore may provide more detailed information to describe the daily movements of an individual and to perform more advanced human mobility analysis (e.g. daily trajectory analysis). However, it could include noisy clusters indicating random activities. A more complex solution should consider both the number of points and the temporal information of the points in a cluster. While the total point size in a cluster is larger than *mpts*, they could be generated just in one day. Therefore, this cluster should not be considered as a regular activity place as well. As such, we can define another temporal threshold as the minimum number of days (*mday*) to further control the selection of regular activity zones. Similar to the cluster size threshold (*mpts*), the choice of this temporal threshold value impacts the activity zone identification, therefore introducing uncertainty.

3.3. *Uncertainty from representation of the activity zone*

3.3.1. *Activity zone boundary*

The next step of examining the activities with digital footprints typically moves to the delineation of the activity boundary of each place detected from spatial clustering as an activity zone, which identifies the potential region and measures the size and trend of the area an individual moves at the place. However, the detected regions are influenced by the choice of the boundary reconstruction algorithms, such as convex hull, non-convex (concave) polygon, minimum boundary box, circumscribed circle, and standard deviational ellipse, resulting in highly varying polygon shapes and sizes (Figure 2). For example, while only one convex hull can be detected for a set of points, varying non-convex boundaries can be produced depending on different algorithms and associated input parameterizations. The chi-shape algorithm, a widely used algorithm for generating the concave hull, requires to provide the maximum length of border edges (*l*) as the input parameter (Duckham *et al*. 2008), making the produced boundaries sensitive to the choice of *l* values. Smaller *l* values may generate a boundary better characterizing and generalizing the activity regions where most of the points are concentrated. However, it could underestimate the area of an activity zones which an individual visits comparing with the shapes delineated by other methods (e.g. convex hull; Figure 2).

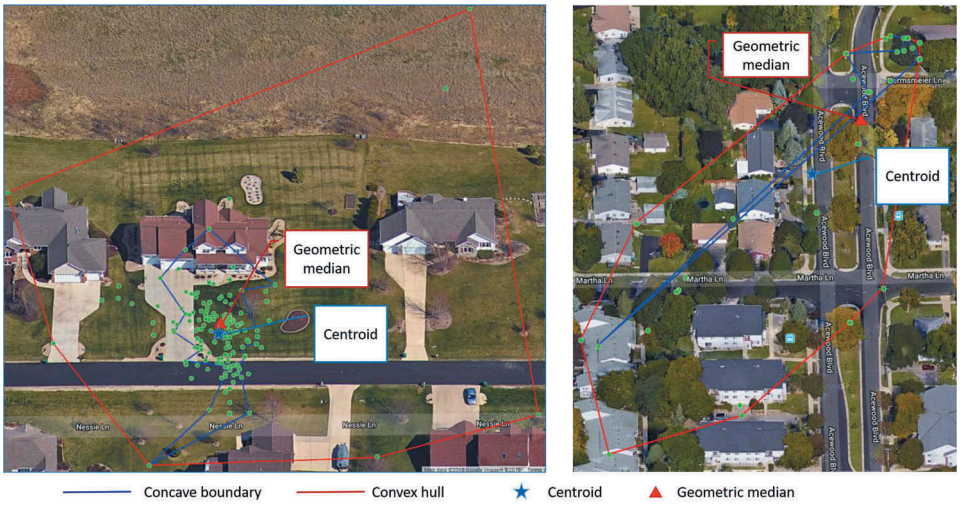

Figure 2. Boundaries and representative locations of two clusters using digital footprints.

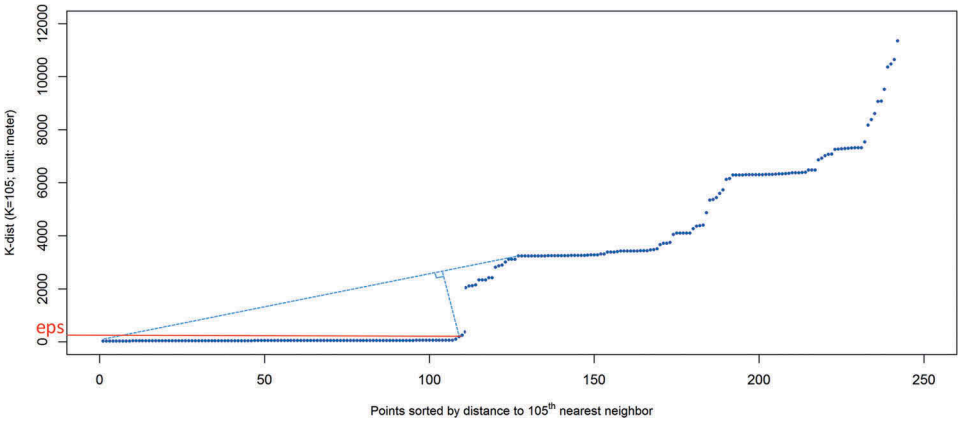

Figure 3. Generation of *eps* value based on k-dist plot.

3.3.2. *Representative location*

To construct a two-dimensional (2D) or three-dimensional (3D) space-time (s-t) path displaying movement trajectory, a representative location should be selected for each activity zone. By connecting consequent locations representing the activities occurred in different time period, s-t paths can be established. As such, the derived s-t paths depend on the selection of representative locations, and inappropriate representative locations could result in inaccurate depiction of regular daily movements. While the centroid (blue stars in Figure 2) of all points within a cluster is commonly used to as the representative point of the cluster (Huang and Wong 2015), it is not real trajectory point generated by an individual through online platforms. In some cases, centroids could be very close to a real point (green points in Figure 2(a)). However, they could be located far away from any real trajectory point (Figure 2(b)), and therefore resulting s-t paths may be less precise while

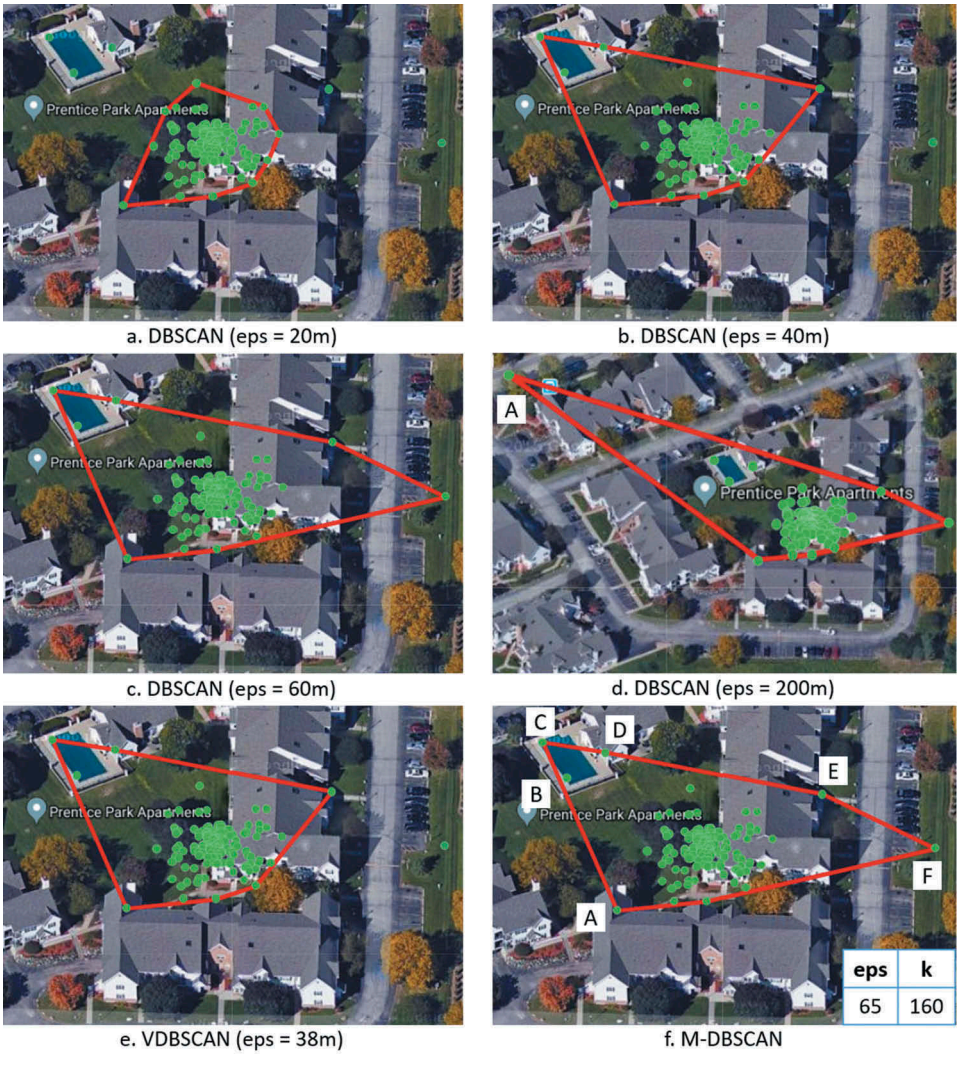

Figure 4. Detection of the first activity zone using DBSCAN with different *eps* values, VDBSCAN and M-DBSCAN for one selected Twitter user.

representing the individual's daily movement trajectory. To cope with this issue, geometric median, the point with minimal total distance to other points in a cluster, is used to represent the cluster (red triangles in Figure 2; Jurgens 2013). Geometric medians ensure that the selected representative locations are meaningful to the individual. Nevertheless, they still cannot guarantee that its location represents the exact spot where an individual stays most in an activity zone (e.g. the room where the individual lives in an apartment, works in an office or has lunch in a restaurant), resulting uncertainty while representing the movement trajectory of the individual using these locations.

3.4. *Activity zone type inference*

Through spatial clustering, a set of clusters are derived from an individual's daily footprints recorded by social media platforms with each cluster covering a region (i.e. activity zone) which this individual regularly visits. To identify the activity zone types, many scholars rely on land-use data as functionality of the activity zone (i.e. land-use type) which is correlated with people's activity there (Huang *et al*. 2014, Huang and Wong 2016). For example, people usually live in residential area and work in industrial or commercial area. Besides, the spatial distributions of POIs within or near the activity zone can indicate different types of people's activities or urban functional zones (Yuan *et al*. 2012, Jiang *et al*. 2015, Gao *et al*. 2017, Cai *et al*. 2018). For example, the inclusion of several restaurants within the activity zone might indicate an eating activity there.

However, the inference of the actual activities performed in activity zone is quite challenging and the accuracy of the inference based on land use and/or POIs cannot be guaranteed. For example, a building in a mix urban area may be used as both residential and commercial usage with upstairs associated with home activities, and downstairs primarily used for eating, shopping, entertainment or other activities. Given an activity zone (e.g. shopping mall), many different types of POIs could be returned. However, normally the primary POI type with the largest count (e.g. shopping store) is used to infer the activity type for an individual' activities on this zone, while other potential types (e.g. restaurant) are often ignored. As such, the results of activity zone type inference are highly uncertain and depend on the inference methods and underlying landscape and surrounding environment where the zones are located.

4. Datasets

This study uses Twitter data, publicly accessible digital footprints, to evaluate M-DBSCAN, and examine the sensitivity of the *eps* and *minpts* values to identify an individual's regular activity zones. Specifically, users in Madison, Wisconsin (Madison), and Washington DC (DC) were chosen, and Twitter's streaming API was used to archive geo-tagged tweets posted within Madison, and DC for a 3-month period. 13,922 unique users were identified within the boundary box of Madison. For DC, 14,066 unique users were identified with each user including 'Washington, DC' as their location information while signing up the Twitter account. For these users, we then re-harvested their historic tweets with the Twitter's search API, which allows to retrieve a maximum of 3,200 most recent tweets for each user.

We used the datasets generated by Madison users to demonstrate and evaluate M-DBSCAN. To make sure each user has sufficient trajectories for clustering discovery, we only selected the users who posted more than 200 geo-tagged tweets, resulting in only 49 eligible users. Since DC has a large number of social media users, we select DC users to examine the activity patterns based on digital footprints. We discard users with geo-tagged tweets less than 50 as these users may have inadequate trajectory points to unfold their activity patterns. Additionally, some users may have highly spare and diversified trajectories, and therefore no spatial cluster can be discovered from their data based on the DBSCAN algorithm. These users were removed from our future analysis. Finally, 3,351 DC users are used for our designed experiments in Section 6.

5. M-DBSCAN

In the following, we introduce the M-DBSCAN algorithm, and then validate how the proposed algorithm can effectively detect activity zones of varying densities subjectively and objectively.

5.1. *M-DBSCAN algorithm*

Given a set of digital footprint points, M-DBSCAN includes three steps to iteratively detect different local *k* and *eps* values for different clusters:

Step 1. Calculate the optimal *k* value with Algorithm 1.
Step 2. Calculate an eps value by partitioning and analyzing the k-dist plot based on the optimal k value (Figure 3).
Step 3. Generate clusters by executing DBSCAN using the *k* value generated in Step 1 as *minpts* and the *eps* value in Step 2 as *eps*.

Above three steps will be repeated until no effective *k* can be found in Step 1 to ensure that all the clusters are detected properly using corresponding local *eps* values. Algorithm 1 describes the detection of *k* value (Step 1; Algorithm 2, line 5). Similar as Chowdhury *et al*. (2010), *k* is determined based on the average mutual distance $d(P_i)$ of every two points to be clustered (line 1). In order to make sure that we compute a local *k* for points in each individual activity zone, rather than deriving a large global *k* for all points, we only calculate the average distance of each point to its near neighbors, instead of all other points. Specifically, to calculate $d(P_i)$ for each point (P_i), any point not within a radius (*R*) of P_i is considered as irrelevant and discarded. The larger *R*, the larger d(Pi) is calculated for each point (Pi), which will produce a larger avg(d) and thus a larger *k*. Similarly, the smaller *R*, the smaller *k* will be calculated. In other words, *R*'s value influences the detection of *k* value and clusters accordingly.

While there is no existing guideline to select an appropriate *R* value, previous study shows that distance from the typical American's house to the edge of his community is between 520 and 1060 m (Donaldson 2013). In other words, activities performed within one zone most likely are within a radius of 1060 m. Therefore, we use 1060 m as the value of *R* in our study and two points more than 1060 m away are considered as two different types of activities. Therefore, mutual distances larger than 1060 m are discarded when calculating $d(P_i)$ and avg(d).

Next, for each P_i, its neighborhood is obtained by collecting all the points within a radius of avg(d) centered on P_i. The amount of points within the neighborhood of P_i is recorded as k_i. A list $[k_i]$ for all points are collected, which can be used to calculate the frequency (or count) of each k_i value. These unique k_i values are recorded as $[k_j]$ and their frequencies are $[f_j]$ (line 4). Then, we delete k_j with small f_j value by selecting the top five k_j with the largest f_j values (line 5). Next, $[k_j]$ are sorted based on their frequency to find the largest k_j, which is selected as *k* value for the set of points (line 7). Based on our experiments (Section 5.2), each user has activity clusters of varying size (the number of points in a cluster) and density. This step iteratively identifies the optimal *k* to detect the current largest cluster(s).

Algorithm 1: Get k and corresponding *eps*

Input: The amounts of remaining points (RP) and rank
Output: k and corresponding *eps* based on k-dist plot
1: calculate avg(d) by averaging the average mutual distance of P_i to all other points within a radius R
2: calculate the amounts of points $[k_i]$ within avg(d) of P_i
3: while (t < rank)
4: generate map $[(k_j, f_j)]$ (f_j is the frequency of k_j value)
5: sort $[(k_j, f_j)]$ descending by f_j
6: maintain the anterior of $[(k_j, f_j)]$ where f_j-$f_1 < 5$
7: sort $[(k_j, f_j)]$ descending by k_j
8: delete (k_1, f_1) from $[(k_i, f_i)]$
9: t ++
10: end
11: if k cannot be found
12: return −1
13: sort distances ($[d_i]$) between P_i and its kth nearest neighbor (k-dist plot)
14: find start points (S_1, S_j) of the first two flat parts on k-dist plot
15: find the knee point S_{knee} with the maximum perpendicular distance to the connecting line of S_1 and S_j
16: return the *k* value and *eps* value, which is the distance from the S_{knee} to its kth nearest neighbor

Step 2 calculates corresponding *eps* value based on the local optimal *k* value. Instead of manually identifying the sharp change of k-dist plot for *eps* (Liu *et al*. 2007, Elbatta and Ashour 2013), we propose a novel method to automatically detect its value (Algorithm 1, lines 13–15). The distances from each P_i to its kth nearest neighbor are sorted ascendingly (an example with k = 105 shown in Figure 3). The first two parts of the point array with relatively flat ascending trend are then detected. The start points of the two parts are connected to find the farthest point between the two start points, known as S_{knee}, which has the maximum perpendicular distance to the connecting line (Line 15). S_{knee} indicates the sharpest change of plotted distances along the connecting line, and thus is selected to derive the *eps* value (Line 16).

Algorithm 2: M-DBSCAN

Input: a set of points representing geo-tagged tweets
Output: representative clusters signified as lists of tweets (RC)
1: while the amounts of remaining points (RP) are no less than 4
2: if RC remain the same
3: rank = 1
4: else rank ++
5: get K and corresponding eps (Algorithm 2)

```
6:      if eps < 0
7:              break
8:      if eps <20
9:              continue
10:     apply DBSCAN (minpts = K, eps = eps)
11:     add newly generated representative clusters to RC
12:     delete newly clustered tweet points from RP
13: end
```

Given a set of points representing an individual's daily trajectories, Algorithm 2 describes the entire process of M-DBSCAN for detecting his or her activity clusters. A loop is used to iteratively obtain the optimal *eps* value representing the radius of the current most prominent cluster, and *k* representing the minimum number of the points in the cluster (Line 5). They are taken as the values of *eps* and *minpts* respectively to conduct DBSCAN on the point set (Line 10). Then clustered points are deleted from the point list which will be further clustered during the next round (Line 12). The process is repeated until there are less than four points left since a cluster with three or less points are not representative enough to be considered as an activity zone, or no more cluster can be found using each possible *k*.

5.2. *Demonstration of the M-DBSCAN using twitter data*

5.2.1. *Detection of activity zones of varying densities*

A Twitter user is selected to demonstrate and evaluate the effectiveness of the M-DBSCAN on automatically detecting activity zones with varying *eps* and *minpts* values. Figure 4 displays the results of the first activity zone detection using DBSCAN with different *eps* values and M-DBSCAN for the selected Twitter user. Based on the surrounding environmental information indicated from the Google Earth map, we can infer that the activity zone is located within an apartment community and ideally footprint points capturing the activities within this region should be all considered as one cluster. With a small *eps* value (e.g. 20 and 40 m), DBSCAN can detect most of the points distributed in the apartment area but not those points in the pool, lawn and parking area. By using the *eps* value of 60 m, all the activities are captured in the cluster. However, further increasing the *eps* value to 200 m, an outlier (point A in Figure 4(d)) located at the street far away from the apartment building is included in the cluster. Similarly, VDBSCAN detects an *eps* value as 38 m, which cannot capture all the activities as well (Figure 4(e)). M-DBSCAN automatically detects the *eps* value as 65 m and obtains the ideal clustering results (Figure 4(f)).

The second example of activity regions (Figure 5) is also a residential community but with single house families. Using DBSCAN with *eps* value equal to 40 m, four small clusters (A, B, C and D) are detected (Figure 5(a)). Increasing the *eps* value, small clusters are merged as a big one (cluster A) and an additional small cluster (cluster B) across the transmeridional main avenue is detected (Figure 5(b)). Next, with 200 m as the *eps* value (Figure 5(c)), all the activity points are captured in two detected clusters. Finally, further increasing the *eps* value to 320 m merges the two clusters (Figure 5(d)), which ideally should be separated as there is a wide avenue between them. As the DBSCAN with an *eps* value of 40 m, VDBSCAN detects

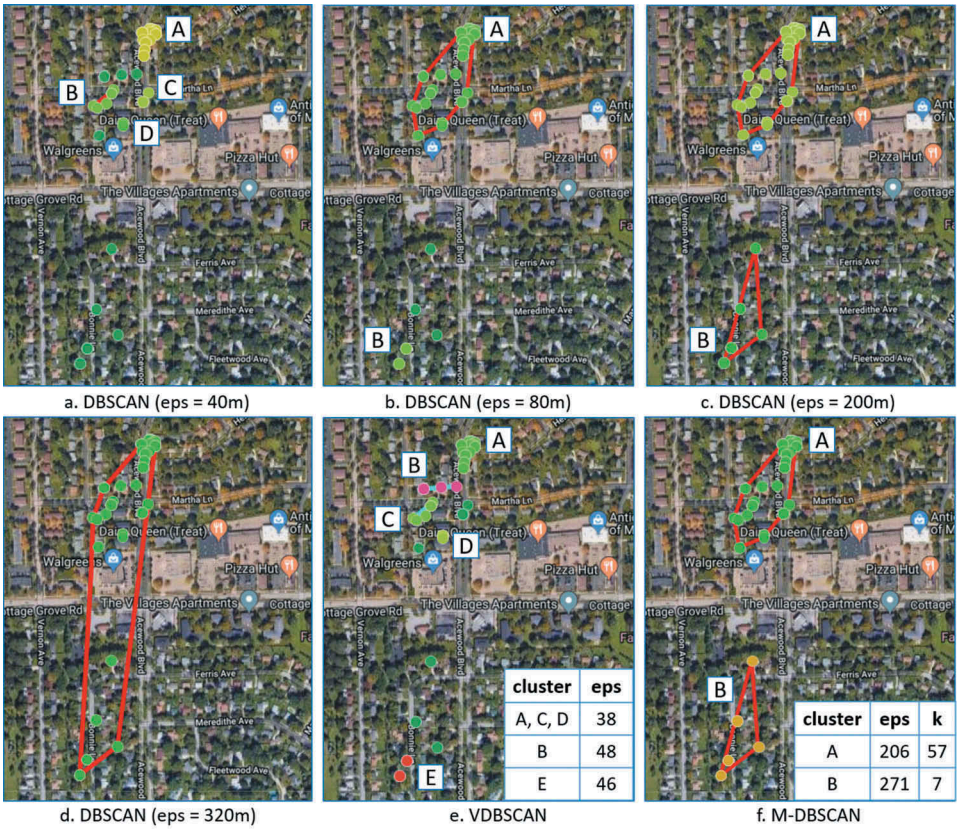

Figure 5. Detection of the second activity zone using DBSCAN with different eps values, VDBSCAN and M-DBSCAN for the selected Twitter user.

fragmentary activity zones (Figure 5(e)) using multiple *eps* values from 38 to 48 m, within the northern area. For the southern area, VDBSCAN is able to detect an additional activity zone (cluster E). Contrastively, M-DBSCAN identifies two optimal clusters using a local eps values of 206 and 271 m, respectively. The detected clusters are similar as DBSCAN using an *eps* value of 200 m (Figure 5(f)).

5.2.2. *Detection of activity zones of low densities*

Besides enabling inclusion of more eligible footprint points, M-DBSCAN can detect clusters of largely diverse densities (Figure 6), some of which are missed by either DBSCAN or VDBSCAN. When using 60 m as *eps*, the residential area (blue boundary in the inset map of Figure 6) demonstrated in Figure 4, can be detected while a prominent shopping area is missing (Figure 6(a)). When using 500 m as eps, the shopping area can be detected (cyan boundary) while the residential area is much enlarged that two noise points are also included (Figure 6(b)). VDBSCAN cannot detect the shopping area either (Figure 6(c)). However, M-DBSCAN can detect both the residential area and the shopping area without evident noise (Figure 6(d)).

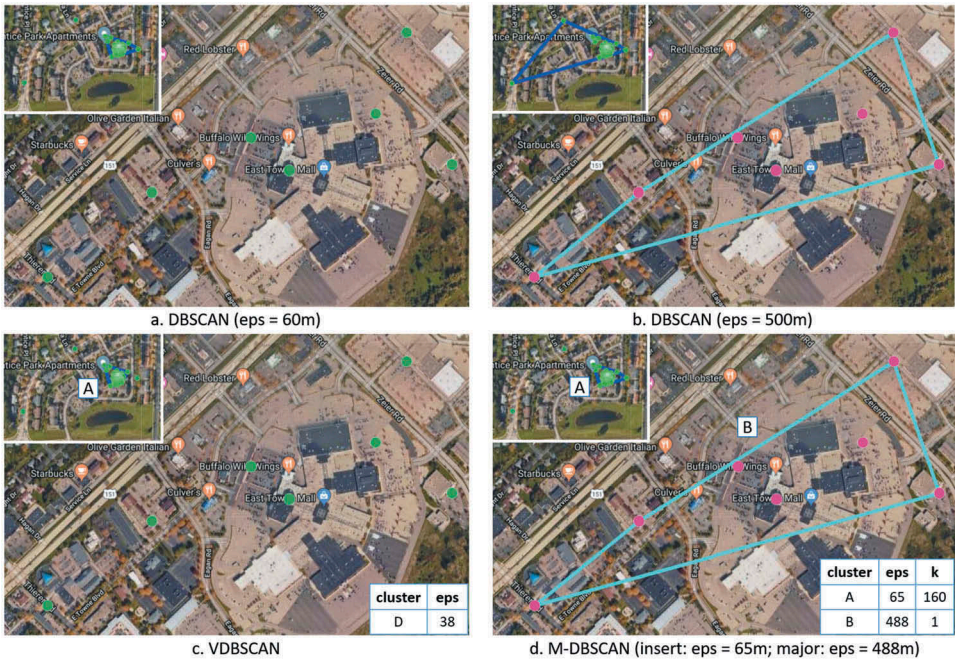

Figure 6. Detecting activity zones of extremely diverse footprint densities using DBSCAN with different eps values, VDBSCAN and M-DBSCAN for one selected Twitter user; insert maps at the upper-left show the detection of the residential activity zones for the selected user as Figure 4; main maps show the activities around the shopping mall area.

The three examples demonstrate that different activity zones include highly diversified activities. While some activities may occur in a compact space (e.g. apartment) which can be detected with smaller *eps* values (Figure 5), an activity zone (e.g. shopping) may include scattered activities in a large space (e.g. mall), and therefore a large *eps* value should be used to capture all of them (Figure 6). The developed M-DBSCAN can automatically detect varying *eps* values for digital footprints of varying density and capture the maximal scope of activities.

5.3. *Clustering evaluation with rand index and adjusted rand index*

Rand index (RI; Rand 1971) is an evaluation measure for a clustering problem in terms of agreement or disagreement between object pairs in two partitions (Walde 2006). In this measurement, if a pair of objects are assigned to the same class or they are assigned to different classes using two different partition methods, the assignment is considered as an agreement (A), otherwise it is considered as a disagreement (D). The similarity of two partitions can thus be evaluated by measuring the overlap of A versus D.

Given a set of objects $O = \{O_1, O_2, \ldots, O_n\}$, the $\mathrm{RI}(C, M)$ between a clustering result ($C = \{C_1, C_2, \ldots, C_x\}$) and the manual classification ($M = \{M_1, M_2, \ldots, M_y\}$) is defined as:

$$RI(C,M) = \frac{\sum_{i<j}^{n} \gamma(o_i, o_j)}{\binom{n}{2}},$$

where n is the number of objects, $\binom{n}{2}$ is computed as n(n-1)/2, and

$$\gamma(o_i, o_j) = \begin{cases} 1 \text{ if there exist } C_A \in C, M_B \in M \text{ such that objects } o_i \text{ and } o_j \text{ are in } C_A \text{ and } M_B, \\ 1 \text{ if there exist } C_A \in C, M_B \in M \text{ such that } o_i \text{ is in both } C_A \text{ and } M_B \\ \quad \text{while } o_j \text{ is in neither } C_A \text{ or } M_B, \\ 0 \text{ otherwise}. \end{cases}$$

Based on above definition, $RI(C,M)$ ranges from 0 to 1. A larger $RI(C,M)$ indicates that partition C is in higher agreement with partition M, indicating the clustering result is more similar to the ground truth (manual classification) and thus is better. If C perfectly agrees with M, $RI(C,M)$ achieves 1. However, the problem with the RI is that its expected value can vary when C and M are both partitioned randomly, which is expected to be constant when both partitions are generated with a random classification model. To solve this problem, the adjusted rand index (ARI; Hubert and Arabie 1985) was therefore developed by introducing the expected similarity of all pair-wise comparisons between C and M partitioned by a random model to the calculation. While the ARI has the maximum value 1, it produces a value of zero for all random partitions. Similarly, a larger ARI means that a clustering result has more agreement with the manual partitions (i.e. ground truth). Hereby, an ARI between C and M ($Rand_{adj}(C,M)$) is defined as (Hubert and Arabie 1985):

$$Rand_{adj}(C,M) = \frac{\sum_{i,j}\binom{t_{ij}}{2} - \frac{\sum_i \binom{t_i}{2} \sum_j \binom{t_j}{2}}{\binom{n}{2}}}{\frac{1}{2}\left(\sum_i \binom{t_i}{2} + \sum_j \binom{t_j}{2}\right) - \frac{\sum_i \binom{t_i}{2} \sum_j \binom{t_j}{2}}{\binom{n}{2}}},$$

Where t_{ij} denotes the number of objects in common between $C_i (\in C)$, *and* M_j $(\in M)$, t_i and t_j indicates the number of objects in common between C_i and M, and between M_j and C respectively.

ARI is widely used to validate clustering results by measuring the agreement between two partitions: one is generated by the clustering algorithm, and the other is produced by external criteria (e.g. manual classification), especially when the two partitions contain different numbers of clusters (Yeung and Ruzzo 2001, Walde 2006, Santos and Embrechts 2009). We thus utilize it to evaluate clustering results of different spatial clustering methods and to validate the effectiveness of the proposed M-DBSCAN model.

Firstly, we sort Twitter users in Madison based on the number of distinct dates when they posted online messages, and select the top 10 users to manually classify their geo-tagged tweets (i.e. partition M). Based on the uncertainties of activity zone detection discussed in Section 3, we follow three principles while manually classifying the geo-tagged tweets (Figure 7) as an activity zone: 1) tweets intensively (≥4 points; i.e. frequent tweets) located at a specific place (e.g. mall, restaurant, school, apartment)

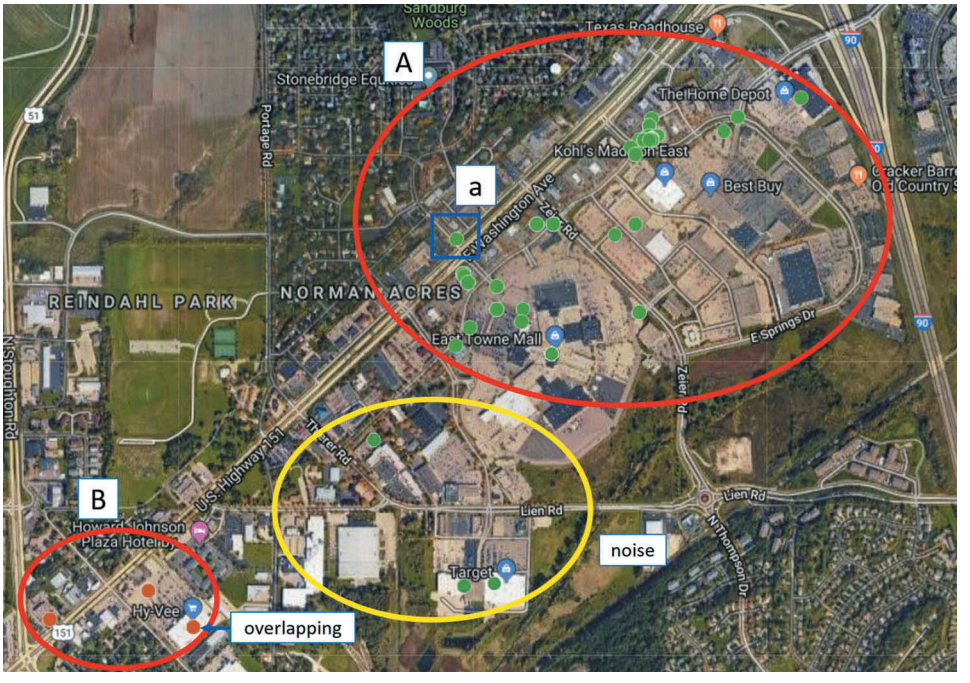

Figure 7. Demonstration of manual classification criterions.

or neighboring places for the same type of activities (e.g. shopping, eating) are considered as one cluster (i.e. cluster A and B; Note while it looks like only three points are displayed in the cluster B, there are repeated points overlapping at the same location as labeled in Figure 7); 2) tweets located outside but near a cluster, which represent the same activity as these tweets within the cluster (e.g. tweets located at the parking lot around a shopping mall; i.e. point a in Figure 7), should also be included in the cluster; 3) infrequent tweets (<4 points) located at a specific place or spread across multiple neighboring places for the same activity type (i.e. noise points in Figure 7) and frequent tweets located at distinct places for different activities (e.g. two tweets located at a restaurant and another two at a post office) are not clustered.

Next, we calculate $RI(C, M)$ and $ARI(C, M)$ for each selected user and the classification results (partition C) generated by different clustering methods including M-DBSCAN, VDBSCAN, and DBSCAN using different eps values. The averaged values of RI and ARI (i.e. $\overline{RI}(C, M)$ and $\overline{ARI}(C, M)$) for the 10 users are then derived (Table 1). During the calculation, all noise points are considered as one cluster and assigned to the same cluster

Table 1. Rand index and adjusted Rand index evaluation results using different clustering methods.

Cluster method	M-DBSCAN	VDBSCAN	DBSCAN				
eps (m)	N/A	N/A	40	80	100	200	300
$\overline{RI}(C, M)$	0.966	0.927	0.930	0.953	0.945	0.951	0.949
$\overline{ARI}(C, M)$	0.964	0.915	0.858	0.940	0.905	0.928	0.923

number (i.e. 0) as the classification consistency of each pair of points should be considered to measure the agreement between *C* and *M*.

The evaluation results show that M-DBSCAN is able to detect the activity zone clusters with the highest agreement with the manually identified activity zones (i.e. ground truth). Specifically, the evaluation results show that M-DBSCAN generates a higher $\overline{RI}$ (C, M) value than any other clustering method, which indicates its effectiveness of separating regular and random activity points. However, the differences of $\overline{RI}$ (C, M) between M-DBSCAN and other methods, especially DBSCAN with eps ≥80m, are rather small. After the adjustment by a random model, a $\overline{ARI}$ (C, M) is generated for each clustering method. These $\overline{ARI}$ (C, M) values are smaller than their $\overline{Rand}(C, M)$ in general. While there is only small decrease over the $\overline{RI}$ (C, M) for M-DBSCAN, the $\overline{ARI}$ (C, M) value decreases more for other methods, such as DBSCAN with eps equal to 40 m and over 100 m. This is because these methods detect a highly varying number of clusters (i.e. either much larger or smaller) as the number of clusters manually labeled.

6. Explore the uncertainty with M-DBSCAN

6.1. *Analysis of activity zones*

Given a set of 2D points capturing a user's historic trajectories, the proposed M-DBSCAN can automatically detect *minpts* and *eps* for grouping all 2D points into different clusters. Using the data of 3,351 selected DC users, 42,962 clusters are detected in total with each user having one or more clusters (Table 2). A majority of the users (75%) have a cluster number less than 16 and half of the users have a cluster number less than 8 with each cluster having a minimum number of 4 points per cluster (Figure 8). While examining these users with a large number of clusters detected, many users are found out to be also users of other social network platforms (such as Foursquare for checking-in places, and Instagram for sharing photos and videos). These users link Twitter account with other different platforms and the messages posted from these platforms can be automatically published on the Twitter. In fact, 2,192 out of 3,351 users are Instagram users, 977 users are Foursquare users, and finally 759 users are both Instagram and Foursquare users. Through these two platforms, users may generate repeated points of the same location (e.g. check-in the same place repeatedly through Foursquare), resulting a cluster to be formed at the locations. Besides, users of these platforms more likely check in at random places (e.g. restaurants) not visiting frequently in a daily basis, therefore, much diversified movement patterns can be observed.

Table 2. Quantile statistics of the number of activity clusters with different *mpts* value.

	Percentage of users			
mpts value	25%	50%	75%	100%
≥4	4	8	16	133
> 4	3	6	13	88
≥6	3	5	9	63
≥8	2	4	7	52
≥10	2	3	6	41
≥12	2	3	5	30

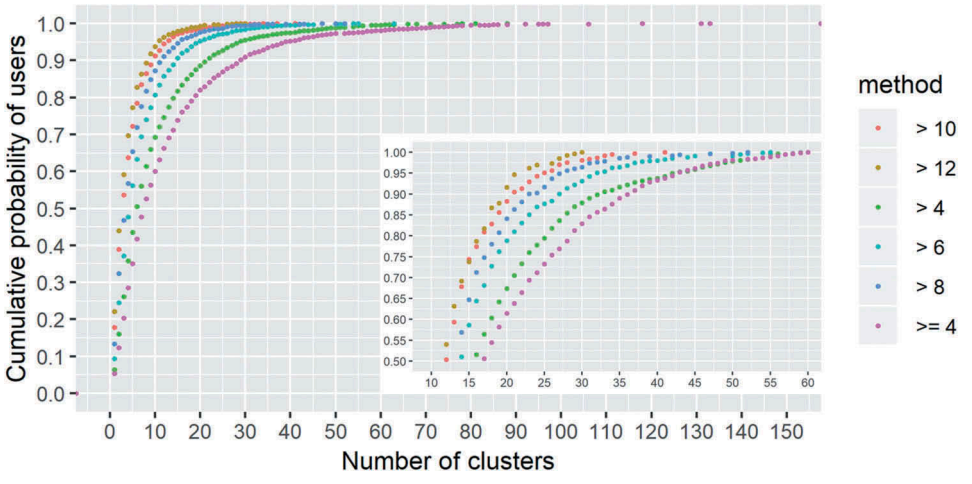

Figure 8. Density of activity zone (cluster) numbers of all users with different value of the minimum number of points (*mpts*) per cluster as the threshold to remove random activity clusters.

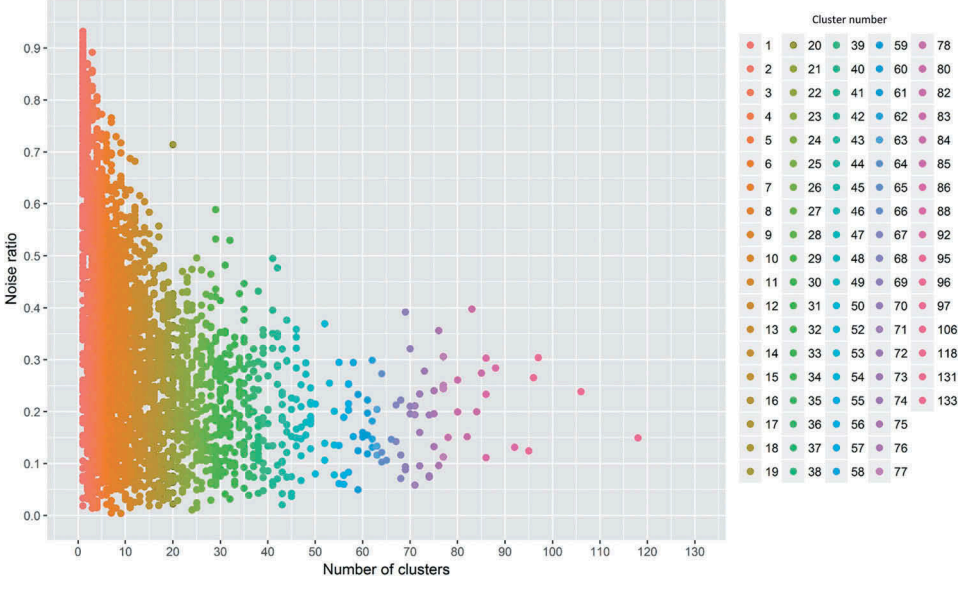

Figure 9. The relationship between the noise ratio and the cluster number of all users with *mpts* = 4.

Figure 9 shows the relationship between the noise ratio and the cluster number generated by all users. Given a set of trajectory points of a user, the noise ratio is the proportion of points not within any cluster, and therefore are considered as noise. For the users having a small number of activity zones (e.g. less than 5) detected, the distribution of noise ratio is relatively random ranging from 0 to approximate 1. With an increasing of activity zone number detected, the noise ratio drops gradually. Especially, for the users with a number of activity zones larger than 40 detected, the noise ratio has a clear distribution pattern with a ratio value mostly less than 40%.

During our experiment, we notice that most of the users have one activity zone, known as the primary activity zone, including the largest number of points (cluster size) and many clusters with relatively smaller point numbers, which most likely capture random activities. This primary cluster captures 46.4% of points that are clustered on average, and the percentage increases to 63.7% combining with the secondary cluster (with the cluster size ranked as second). In the proposed M-DBSCAN algorithm, the minimum number of points (*mpts*) should be included in a cluster is 4. In fact, if we remove the clusters with the number of points as 4 for each user, most of the users (75%) only have less than 13 clusters (second row of Table 2). In other words, each user will have three small clusters with only 4 points on average. For users with a large number of trajectory points, those clusters most likely capture some random activities for users. However, if a user only has a small number of points, these clusters could also record regular activities. Nevertheless, we can use the value of *mpts* as a threshold to remove the potential noisy clusters. If using 12 as *mpts* value, most of the users (75%) have less than 5 clusters detected. This is reasonable since people's regular activity is performed in five regular zones, including home, work, entertainment, eating, and shopping. Therefore, the choice of *mpts* affects the uncertainty of detecting the number of regular activity zones.

Quite interestingly, the primary activity zones are typically detected with a relatively small *eps* value. With an *eps* value larger than 439 m, we capture dispersed clusters with low density of points (Figure 10 left). Most of the clusters (80%) are detected with an *eps* value smaller than 439 m (Figure 10 left). The value of *minpts* detected by M-DBSCAN has a strong positive linear relationship (Pearson's correlation coefficient 0.947) with the size of the cluster to be detected (Figure 10 right). This makes sense as points in a larger cluster typically are denser, which means that each point has more neighbors reachable, and therefore a larger value of *minpts* should be used.

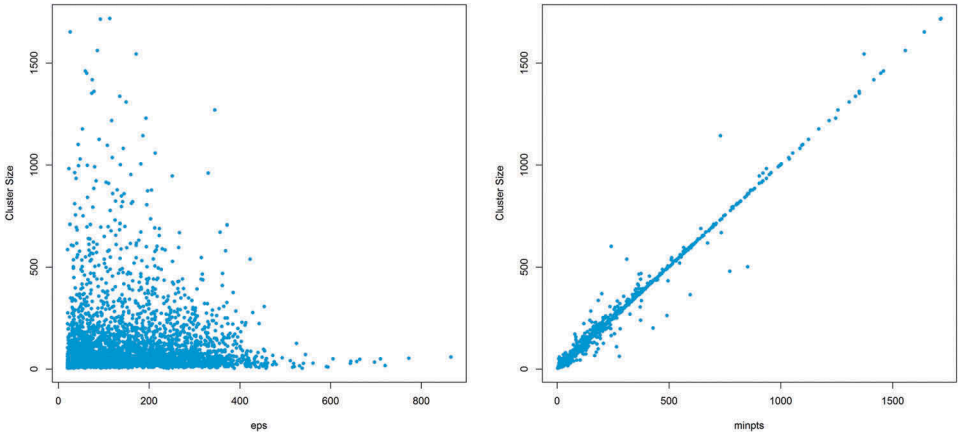

Figure 10. The correlation of *eps* and *minpts* with the number of points (cluster size) in each cluster.

6.2. *Selection of eps value for DBSCAN*

To explore the range and distribution patterns of appropriate parameter (*eps* and *minpts*) values for activity zone detection of individuals using DBSCAN algorithm, we change the *eps* value from 20 m approximating to the uncertainty in GPS readings, to 400 m. Using a value of 20 m, a large number of small clusters are detected (Table 3). The cumulative distribution of cluster numbers using DBSCAN with an *eps* of 20 m is very different from other methods (Figure 11) confirming that with such a small *eps* value, DBSCAN more likely groups the points into smaller clusters, running the risk of separating the activities of the same zone captured by the geo-tagged tweets into different zones. With the increasing of *eps* value from 20 to 40 m, smaller clusters are merged into larger clusters, resulting in the decrease of the number of clusters detected for all users in total. However, if we continue to increase the *eps* value from 40 to 400 m, an increasing number of clusters are detected, indicating that dispersed clusters with low density of points are identified. Herein, M-DBSCAN has the largest number of clusters detected except for using DBSCAN with an *eps* value of 20 m. This confirms that the M-DBSCAN can capture

Table 3. Results of M-DBSCAN and DBSCAN with different *eps* values and a *minpts* value of 4.

Cluster methods	Average number of clusters	Average noise ratio	Total clusters
DBSCAN (*eps* = 20)	9.75	69.0%	66,042
DBSCAN (*eps* = 40)	10.28	54.1%	35,294
DBSCAN (*eps* = 80)	11.12	49.0%	38,394
DBSCANE (*eps* = 100)	11.32	47.2%	39,187
DBSCANE (*eps* = 200)	11.97	43.1%	41,550
DBSCANE (*eps* = 300)	12.11	40.6%	42,076
DBSCANE (*eps* = 400)	7.82	38.7%	42,190
M-DBSCAN	12.50	32.7%	42,962

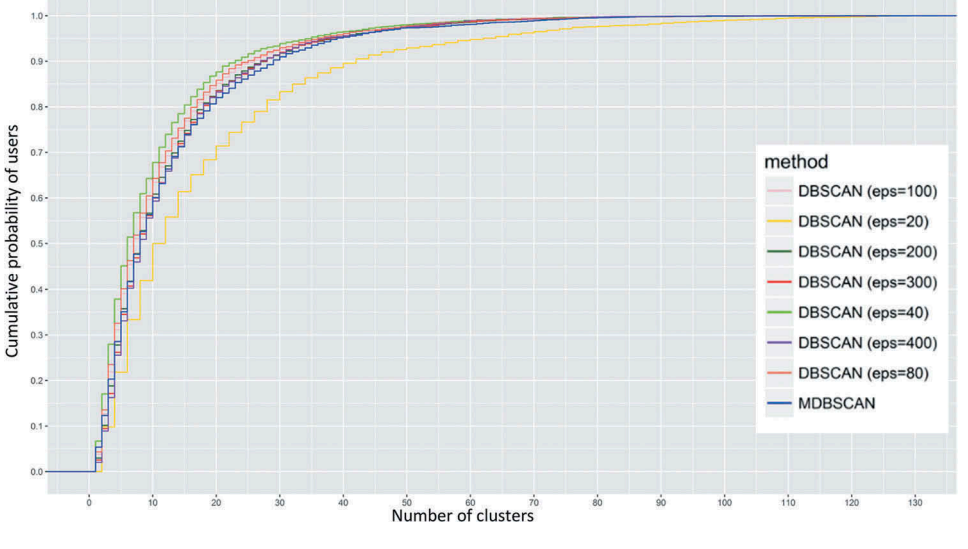

Figure 11. Density of activity zone (cluster) numbers of all users with M-DBSCAN and DBSCAN with different *eps* values and a *minpts* value of 4.

dispersed clusters with low density of points by automatically using larger *eps* values to detect these clusters (Figure 10 left).

With an *eps* value of 20 m used for DBSCAN, a major of points (69%) are detected as noise and therefore are not appropriate for mobility analysis for users on average (Table 3). While increasing *eps* value could reduce the noise ratio on average, the number of discarded points (noise) is still very high. With the increasing of *eps* values from 20 to 300 m, more clusters are detected, which is reasonable as a large *eps* value can help capture clusters of low density. However, if we continue to increase *eps* value from 300 to 400 m, only about eight clusters are detected on average indicating small clusters potentially with different activities being merged into large clusters. In contrast, the proposed M-DBSCAN has the smallest average noise ratio compared with the results detected by DBSCAN with different eps values. With an *eps* value as 300 m, DBSCAN can detect similar number of clusters as M-DSBCAN on average, though with a higher noise ratio.

To further identify appropriate *eps* values for DBSCAN, the two-sample Kolmogorov–Smirnov (KS) statistic (D value in Table 4) is used to test whether two underlying probability distributions of the number of clusters detected for the selected using DBSCAN using varying *eps* values and M-DBSCAN differ. The KS statistics provides a measure of the distance between the empirical distribution function of two samples (e.g. n and m). Given the empirical distribution functions of the first and the second sample $F_{1,\,n}$ and $F_{2,\,m}$, the KS statistic $D_{n,\,m}$ is calculated as:

$$D_{n,m} = sup_x \left| F_{1,\,n}(x) - F_{2,\,m}(x) \right|,$$

where sup_x is the supremum function. The smaller the *D* value, the better the distribution the two samples fit each other.

The test results in Table 4 show DBSCAN with *eps* values as 200 and 300 m have similar distribution of the number of clusters identified for each user as M-DBSCAN with a small D value. The significance indicated by p-value (0.2953) using 200 m as *eps* is greatly larger than a typical significance level (0.05) and thus we cannot reject the null hypothesis statement that two samples come from the same probability distribution. Besides, the p-value using 300 m as *eps* value is also greater than 0.05, which does not reject the null hypothesis. While increasing the *eps* value from 300 to 400 m, the KS test p-value dropped to 0.00869 and D value increased accordingly. In addition, the D value is larger than the critical value (0.033) at the significance level of 0.05, further confirming the dissimilarity of the distributions of the number of clusters detected by the two

Table 4. Two-sample Kolmogorov–Smirnov test of the probability distribution of the number of clusters using M-DBSCAN and DBSCAN with different *eps* values.

Cluster methods (*eps* in meters)	D value	p-value
DBSCAN (*eps* = 20)	0.148	< 2.2e-16
DBSCAN (*eps* = 40)	0.101	3.775e-15
DBSCAN (*eps* = 80)	0.050	0.0003976
DBSCAN (*eps* = 100)	0.040	0.0102
DBSCAN (*eps* = 200)	0.024	0.2953
DBSCAN (*eps* = 300)	0.032	0.06995
DBSCAN (*eps* = 400)	0.040	0.00869

methods. From the statistic results (Tables 3 and 4), a value between 200 and 300 m is recommended for *eps* while using DBSCAN for detecting activity zones in general.

7. Conclusion and future work

With the advancement of communication and information technologies, social media platforms emerge as new solutions to record people's movements in a daily basis. While exploring mobility patterns, spatial clustering over the digital footprint points is typically used to detect places where an individual regularly visits. Among all spatial clustering algorithms, DBSCAN is widely used as it is effective to detect clusters of arbitrary shape with noise, and only needs to supply *minpts* and *eps* as input, the values of which are relatively easy to determine compared to other algorithms. However, DBSCAN is sensitive to the two input parameters while detecting regular activity clusters from users' social media points, which are sparsely, irregularly distributed in space, and featured of varying densities. DBSCAN and existing improved DBSCAN methods (Ester *et al*. 1996, Ertöz *et al*. 2003, Liu *et al*. 2007, Wang *et al*. 2016) based on a k-dist plot, where k-dist is defined as the distance from each point to its k nearest point, often fall short to identify activity clusters.

This paper develops an improved density-based clustering algorithm based on DBSCAN, named as M-DBSCAN, which can automatically produce a set of activity zones (clusters), and select an appropriate *eps* and *minpts* value for each activity zones. In addition, M-DBSCAN can significantly reduce the noise ratio (the proportion of tweets not included in any cluster) by identifying all points capturing the activities performed in each zone. Using the historical online geo-tagged tweets of users in Madison and in DC, the results of M-DBSCAN and DBSCAN with varying *eps* value indicate that: 1) M-DBSCAN detects more clusters (activity zones) for each user, and results in a lower noise ratio (the proportion of tweets not included in any cluster); 2) A value of 40 m or higher should be used for *eps* in order to reduce the possibility of collapsing the activities of the same zone captured by the social media data points into different zones, and ensure an average noise ratio less than 54% during the clustering process; 3) A value between 200 and 300m is mostly recommended for *eps* while using DBSCAN for detecting activity zones; and finally 4) There is no optimal value for *minpts*. The value of *minpts* detected by M-DBSCAN displays a strong positive linear relationship (Pearson's correlation coefficient 0.947) with the size of the cluster to be detected.

The proposed M-DBSCAN was first evaluated subjectively by analyzing a selected user's activity zones detected by different clustering methods (Section 5.2). Next, we manually identify the daily activity zones of 10 users in Madison as ground truth data to evaluate the effectiveness of M-DBSCAN objectively. While determining the membership of a point (Section 3.2), it is challenging to evaluate whether it should be considered a part of an activity zone or as outlier (Figure 4(c) vs (d)). Similarly, it is quite difficult to identify if one cluster is better to describe an individual's activity zone than to separate it into two clusters or to keep some points as outliers (e.g. Figure 5(d) vs (f)). As such, contextual information derived from the social media contexts (e.g. text messages) and temporal information could be integrated to further improve the detection of activity zones. Finally, this paper explores the uncertainty of activity zone detection using

trajectory data of DC users. In the future, footprints of social media users in different cities can be explored to identify activity zones and patterns across regions.

The proposed M-DBSCAN can effectively detect user activity zones based on digital footprints, and has potential to facilitate a wide range of practical applications such as human mobility study, tourism recommender systems, and business site selection, where POIs should be discovered using digital footprints from massive users. Besides, the study results provide general guideline to choose optimal values for *eps* and *minpts* while running the DBSCAN algorithm to detect individual activity zones. In future, we will integrate multi-sourced data (e.g. OpenStreetMap land use, and Google place service data) to detect the activity zone types of each cluster, while allows us to further explore user tweeting behaviors at different types of activity zones and complex mobility patterns, such as trajectory patterns of different socioeconomic and demographic groups.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

Support for this research was provided by the University of Wisconsin - Madison Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation.

Notes on contributors

Xinyi Liu is a first year PhD student in the Department of Geography at University of Wisconsin-Madison. She earned a master's degree in GIScience/Cartography from the same department. Her doctoral research investigates spatial data mining and its applications in different study areas, especially human mobility.

Qunying Huang is an assistant professor in the Department of Geography at University of Wisconsin-Madison. Her fields of expertise include Spatial Computing, Spatiotemporal Big Data Analytics, Spatial Data Science, and Geographic Information Science (GIScience). She employs different computing models, such as Cluster, Grid, GPU, Citizen computing, and especially Cloud Computing to address big data and computing challenges in the GIScience, and leverages social media data for various applications, such as disaster management and human mobility.

Dr. Song Gao, is an Assistant Professor in GIScience at the Department of Geography, University of Wisconsin-Madison. He holds a Ph.D. in Geography at the University of California, Santa Barbara. His main research interests include Place-Based GIS, Geospatial Big Data Analytics, GeoAI, Geospatial Semantics, Human Mobility and Urban Computing.

ORCID

Xinyi Liu http://orcid.org/0000-0002-1431-8816
Qunying Huang http://orcid.org/0000-0003-3499-7294
Song Gao http://orcid.org/0000-0003-4359-6302

References

Ankerst, M., *et al*., 1999. OPTICS: ordering points to identify the clustering structure. *In*: ed. *Proceedings of the 1999 ACM SIGMOD international conference on management of data*, 31 May–03 June. Philadelphia, Pennsylvania, USA, 49–60. New York, NY: ACM.

Cai, L., *et al*., 2018. Integrating spatial and temporal contexts into a factorization model for POI recommendation. *International Journal of Geographical Information Science*, 32 (3), 524–546. doi:10.1080/13658816.2017.1400550

Campello, R.J., *et al*., 2015. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10 (1), 1–51. doi:10.1145/2733381

Campello, R.J., Moulavi, D., and Sander, J., 2013. Density-based clustering based on hierarchical density estimates. *In*: J. Pei, V. S. Tseng, L. Cao., H. Motoda, and G. Xu, eds. *PAdvances in Knowledge Discovery and Data Mining*. *PAKDD*. Lecture Notes in Computer Science, 7819. Berlin: Springer.

Chowdhury, A.R., Mollah, M.E., and Rahman, M.A., 2010. An efficient method for subjectively choosing parameter 'k'automatically in VDBSCAN (Varied Density Based Spatial Clustering of Applications with Noise) algorithm. *In*: ed. *The 2nd international conference on computer and automation engineering (ICCAE)*, 26–28 February. Singapore, 38–41. Piscataway: IEEE.

Donaldson, K., 2013. *How big is your neighborhood? Using the AHS and GIS to determine the extent of your community* [online]. Available from: https://www.census.gov/programs-surveys/ahs/research/working-papers/how_big_is_your_neighborhood.html [Accessed 30 April 2018].

Duckham, M., *et al*., 2008. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition*, 41 (10), 3224–3236. doi:10.1016/j.patcog.2008.03.023

Elbatta, M.T. and Ashour, W.M., 2013. A dynamic method for discovering density varied clusters. *International Journal of Signal Processing, Image Processing, and Pattern Recognition*, 6 (1), 123–134.

Ertöz, L., Steinbach, M., and Kumar, V., 2003. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. *In*: D. Barbara and C. Kamath, eds. *Proceedings of the 2003 SIAM international conference on data mining*, 1–3 May. San Francisco, CA, 47–58. Available from: https://epubs.siam.org/doi/book/10.1137/1.9781611972733?mobileUi=0

Ester, M., *et al*., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *In*: E. Simoudis, J. Han, and U. Fayyad, eds. *The second international conference on knowledge discovery and data mining (KDD)*, 2–4 August. Portland, Oregon, 226–231. Palo Alto: AAAI PRESS.

Gao, S., *et al*., 2014. Detecting origin-destination mobility flows from geotagged Tweets in greater Los Angeles area. *In*: M. Duckham, E. Pebesma, K. Stewart, and A. U. Frank, eds. *The eighth international conference on geographic information science (GIScience'14)*, 24–26 September. Vienna, Austria. New York, NY: Springer.

Gao, S., 2015. Spatio-temporal analytics for exploring human mobility patterns and urban dynamics in the mobile age. *Spatial Cognition & Computation*, 15 (2), 86–114. doi:10.1080/13875868.2014.984300

Gao, S., Janowicz, K., and Couclelis, H., 2017. Extracting urban functional regions from points of interest and human activities on location-based social networks. *Transactions in GIS*, 21 (3), 446–467. doi:10.1111/tgis.2017.21.issue-3

Hu, Y., *et al*., 2015. Extracting and understanding urban areas of interest using geotagged photos. *Computers, Environment and Urban Systems*, 54, 240–254. doi:10.1016/j.compenvurbsys.2015.09.001

Huang, Q., 2017. Mining online footprints to predict user's next location. *International Journal of Geographical Information Science*, 31 (3), 523–541. doi:10.1080/13658816.2016.1209506

Huang, Q., Cao, G., and Wang, C., 2014. From where do Tweets originate?-A GIS approach for user location inference. *In*: ed. *Proceedings of the 7th ACM SIGSPATIAL international workshop on location-based social networks*, 4 November. Dallas/Fort Worth, Texas. New York, NY: ACM.

Huang, Q. and Wong, D.W., 2015. Modeling and visualizing regular human mobility patterns with uncertainty: an example using Twitter data. *Annals of the Association of American Geographers*, 105 (6), 1179–1197. doi:10.1080/00045608.2015.1081120

Huang, Q. and Wong, D.W., 2016. Activity patterns, socioeconomic status and urban spatial structure: what can social media data tell us? *International Journal of Geographical Information Science*, 30, 1873–1898. doi:10.1080/13658816.2016.1145225

Hubert, L. and Arabie, P., 1985. Comparing partitions. *Journal of Classification*, 2 (1), 193–218. doi:10.1007/BF01908075

Jiang, S., *et al*., 2015. Mining point-of-interest data from social networks for urban land use classification and disaggregation. *Computers, Environment and Urban Systems*, 53, 36–46. doi:10.1016/j.compenvurbsys.2014.12.001

Jurgens, D., 2013. That's what friends are for: inferring location in online social media platforms based on social relationships. *ICWSM*, 13 (13), 273–282.

Kang, C., *et al*., 2012. Intra-urban human mobility patterns: an urban morphology perspective. *Physica A: Statistical Mechanics and Its Applications*, 391 (4), 1702–1717. doi:10.1016/j.physa.2011.11.005

Karami, A. and Johansson, R., 2014. Choosing DBSCAN parameters automatically using differential evolution. *International Journal of Computer Applications*, 91 (7), 1–11. doi:10.5120/15890-5059

Kwan, M.-P., 2013. Beyond space (as we knew it): toward temporally integrated geographies of segregation, health, and accessibility: space–time integration in geography and GIScience. *Annals of the Association of American Geographers*, 103 (5), 1078–1086. doi:10.1080/00045608.2013.792177

Lin, J. and Cromley, R.G., 2018. Inferring the home locations of Twitter users based on the spatiotemporal clustering of Twitter data. *Transactions in GIS*, 22 (1), 82–97. doi:10.1111/tgis.2018.22.issue-1

Liu, P., Zhou, D., and Wu, N., 2007. VDBSCAN: varied density based spatial clustering of applications with noise. *In*: ed. *2007 international conference on service systems and service management*. Chengdu, China, 1–4. Piscataway: IEEE.

Liu, Y., *et al*., 2015. Social sensing: A new approach to understanding our socioeconomic environments. *Annals of the Association of American Geographers*, 105 (3), 512–530. doi:10.1080/00045608.2015.1018773

Luo, F., *et al*., 2016. Explore spatiotemporal and demographic characteristics of human mobility via Twitter: a case study of Chicago. *Applied Geography*, 70, 11–25. doi:10.1016/j.apgeog.2016.03.001

Mahmud, J., Nichols, J., and Drews, C., 2012. Where is this Tweet from? Inferring home locations of Twitter users. *ICWSM*, 12, 511–514.

Mathew, W., Raposo, R., and Martins, B., 2012. Predicting future locations with hidden Markov models. *In*: *Proceedings of the 2012 ACM conference on ubiquitous computing*, 911–918. New York, NY: ACM.

Mennis, J. and Guo, D., 2009. Spatial data mining and geographic knowledge discovery—an introduction. *Computers, Environment and Urban Systems*, 33 (6), 403–408. doi:10.1016/j.compenvurbsys.2009.11.001

Moreira, G., Santos, M.Y., and Moura-Pires, J., 2013. SNN input parameters: how are they related? *In*: *Parallel and distributed systems (ICPADS), 2013 international conference on*, Seoul, South Korea, 492–497. Piscataway: IEEE.

Morstatter, F., *et al*., 2013. Is the sample good enough? Comparing data from twitter's streaming api with twitter's firehose. *In*: *Proceedings of the 7th International Conference on Weblogs and Social Media*, ICWSM 2013, Jul 8 -11, Cambridge, MA, United States. Palo Alto: AAAI press.

Noulas, A., *et al*., 2012a. A tale of many cities: universal patterns in human urban mobility. *PloS one*, 7 (5), e37027. doi:10.1371/journal.pone.0037027

Noulas, A., *et al*., 2012b. Mining user mobility features for next place prediction in location-based services. *In*: ed. *Data mining (ICDM), 2012 IEEE 12th international conference on*, 10–13 December. Brussels, Belgium, 1038–1043. Piscataway: IEEE.

Parvez, A.W.M.M., 2012. Data set property based 'K'in VDBSCAN clustering algorithm. *World of Computer Science and Information Technology Journal (WCSIT)*, 2 (3), 115–119.

Rand, W.M., 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66 (336), 846–850. doi:10.1080/01621459.1971.10482356

Richardson, D.B., *et al*., 2013. Spatial turn in health research. *Science*, 339 (6126), 1390–1392. doi:10.1126/science.1232257

Santos, J.M. and Embrechts, M., 2009. On the use of the adjusted rand index as a metric for evaluating supervised classification. *In*: ed. *International conference on artificial neural networks*, 14–17 September. Limassol, Cyprus, 175–184. Berlin: Springer.

Shaw, S.-L., Tsou, M.-H., and Ye, X., 2016. Human dynamics in the mobile and big data era. *International Journal of Geographical Information Science*, 30 (9), 1687–1693. doi:10.1080/13658816.2016.1164317

Steiger, E., Albuquerque, J.P., and Zipf, A., 2015. An advanced systematic literature review on spatio-temporal analyses of Twitter data. *Transactions in GIS*, 19 (6), 809–834. doi:10.1111/tgis.12132

Walde, S.S.I., 2006. Experiments on the automatic induction of German semantic verb classes. *Computational Linguistics*, 32 (2), 159–194. doi:10.1162/coli.2006.32.2.159

Wang, S., Liu, Y., and Shen, B., 2016. MDBSCAN: multi-level density based spatial clustering of applications with noise. *In*: ed. *Proceedings of the 11th international knowledge management in organizations conference on the changing face of knowledge management impacting society*, 25–28 July. Hagen, Germany, 21. New York, NY: ACM.

Xu, Y., *et al*., 2016. Another tale of two cities: understanding human activity space using actively tracked cellphone location data. *Annals of the American Association of Geographers*, 106 (2), 489–502.

Yeung, K.Y. and Ruzzo, W.L., 2001. Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics*, 17 (9), 763–774.

Yuan, J., Zheng, Y., and Xie, X., 2012. Discovering regions of different functions in a city using human mobility and POIs. *In*: ed. *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining*, 23–25 July. Edmonton, AB, Canada, 186–194. New York, NY: ACM.

Zandbergen, P.A., 2009. Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning. *Transactions in GIS*, 13, 5–25. doi:10.1111/tgis.2009.13.issue-s1

Zhou, C., *et al*., 2004. Discovering personal gazetteers: an interactive clustering approach. *In*: ed. *Proceedings of the 12th annual ACM international workshop on geographic information systems*, 12–13 November. Washington, DC, USA, 266–273. New York, NY: ACM.