International Journal of Engineering & Technology, *7 (4.39) (2018) 594-598*

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Clustering Method of Moving Points Based on Density and Directionality

Jinman Kim[1], Hyeonsang Hwang[2], EuiChul Lee[*3]

[1]Research Institute for Intelligent Engineering Informatics, Sangmyung University20, Hongjimun 2-gil, Jongno-gu, Seoul,03016, Republic of Korea
[2]Department of Computer Science, Sangmyung University20, Hongjimun 2-gil, Jongno-gu, Seoul, 03016, Republic of Korea
*[*3]Department of Intelligent Engineering Informatics for Human, Sangmyung University20, Hongjimun 2-gil, Jongno-gu, Seoul, 03016, Republic of Korea*
[*]Corresponding author E-mail: eclee@smu.ac.kr

Abstract

Background/Objectives: Density-based spatial clustering of applications with noise (DBSCAN) is a data-clustering algorithm that applies density-based clustering methods. Because it considers the density only at a single instance, this method is problematic when clusters of points change with time.
Methods/Statistical analysis: Our method analyzes the "staying time" and "directionality" of the GPS trajectory. As it incorporates directionality and has improvements over a conventional DBSCAN method, it is termed as DBSCAN-D. The "staying time" is the interval between two locations where the GPS data are obtained. The "directionality" is the direction toward the upcoming position relative to the previous location. This is obtained by analyzing the GPS data that are generated sequentially.
Findings: Because the time series data such as GPS data can effectively utilize the directionality information according to the recorded movements, the proposed DBSCAN-D method is found to be particularly suitable for clustering applications.
Improvements/Applications: In our work, we have applied open GPS data (Geolife), and confirmed that the proposed method exhibits superior performance compared to the existing DBSCAN method.

Keywords: *Clustering, density-based spatial clustering of applications with noise (DBSCAN), trajectory, point of interest (POI), global positioning system (GPS).*

1. Introduction

In recent times, the spread of mobile terminals, such as smart phones and the development of positioning techniques such as global positioning systems (GPSs) have enabled the potential for acquiring the trajectory data of a moving object. The moving object can continuously produce a spatiotemporal dataset consisting of positions and locational shapes. Moreover, with increasing use of location-based services (LBSs) in various applications, interests have been growing to extract meaningful information from these logs (i.e., GPS trajectory).

An LBS is an essential part of today's mobile technologies. Either used for entertainment purposes, or for accessing information through a mobile network, such devices register their geographical positions while in use[1]. Common usages of such technologies are wireless positioning gadgets for locating mobile terminals, LBS servers for providing core infrastructure technology services, and numerous other LBS applications[2].

Currently, many smartphone applications are providing services based on the user's location either directly or indirectly. In particular, an LBS uses the point of interest (POI) information as well as the location information through the user devices. A POI is a specific point location that someone may find useful or interesting[3]. More specifically, it is a combination of two factors. First, a physical location a person is interested on, which may also be a specific place on a map or drawing. Second, the location information of other adjacent amenities such as roads and buildings. There are three methods of expressing a POI. The most common method is to select the locations that are either well known, or where congregation of people is identified. Another method is to extract the POI directly from the location data from a device such as a GPS. The latter method expresses the POI by combining these two methods[4].

Other methods include clustering based on distance using mean shift[5], and clustering using K-means[6] to identify meaningful locations from the trajectory data of moving objects. The most common method is to perform clustering based on data density. The density-based clustering method assumes that the clusters are regions of high density that are separated from each other in space. In other words, the clusters can be identified by observing their density and if these can be separated from each other. In this way, the clusters of various shapes and sizes can be extracted effectively. However, the density-based clustering method has two problems. First, this method produces varying shapes depending on the selection of the parameters used for the cluster determination scheme. Second, when a group with density-difference exists in a dataset, the probability that a group with a relatively higher density will be recognized as a cluster, is lowered[7].

Density-based spatial clustering of applications with noise (DBSCAN)[8] is a representative technique of density-based clustering. Since the introduction of this technique, many studies have attempted to solve the recognized problems of density-based clustering. However, most studies focused on cluster extraction, that is, to create clusters in a dataset. In our previous study, we proposed the DBSCAN-D method [9]. This is a clustering technique for extracting POIs from the location datasets generated from moving objects such as GPS data. This method performs clustering using directionality information as well as the location of the moving objects. In this study, clustering is performed through the location and directionality information of the moving objects as found from the GPS trajectory data. It applies the method proposed in our previous study.

2. Related Work

2.1. DBSCAN

DBSCAN was first proposed in [8] as a density-based clustering technique. It was assumed that the cluster would consist of a high-density region separated by the object's density in the space. It was also assumed that the cluster would be created by using the position information of each object, and the density of the surrounding data. This method has been used in various fields such as education, medicine, finance and marketing to analyze the consumer behaviors, similarity in consumer characteristics, or purchasing patterns related to financial fraud[10].

DBSCAN expresses a given dataset as a vector. It uses the density between the vectors to identify the clusters and the data points that are not in any given cluster allowing their proper separation. The terms used in this technique are as follows [Figure 1].

- *Eps* (epsilon): This parameter specifies the radius around a point to measure the density of any point. It is expressed using the Greek letter *ε*.
- *Eps*-neighborhood: This represents a set of neighboring objects within an *Eps* radius centered around an arbitrary point.
- *MinPts* (minimum number of points): This denotes the minimum number of points that must exist within the *Eps* radius around a point to measure the density of any point. That is, this represents the minimum number of neighboring objects for which one point is a central object.
- Core point: If there are a number of neighboring objects more than or equal to *MinPts* within the *Eps* radius around an arbitrary point *p*, then point *p* is called a center object.
- Border point: This is a point that is less than *MinPts* in the *Eps* radius around an arbitrary point, but falls within the *Eps* radius of another core point of those objects.
- Noise point: This is the point excluding the core point and the border point, which implies all the points not included in the cluster.

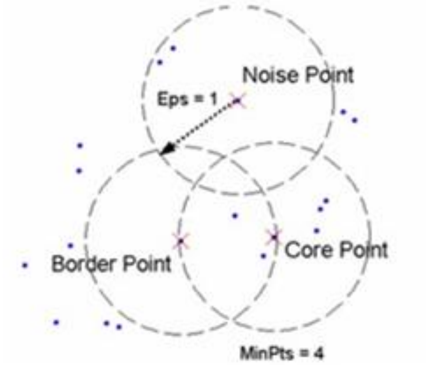

Figure 1: Classification of certain objects (points) in DBSCAN

Six definitions and two lemmas for generating clusters in the density domain are given in [8]. These are as follows.

Definition 1: (The *Eps*-neighborhood of a point) An *Eps*-neighborhood at any point is a set of neighbors within the *Eps* radius from that point.

$$N_{Eps}(p) = q \in D \mid dist(p,q) \leq Eps \quad (1)$$

If point q belongs to the dataset *D*, and the distance between points *p* and *q* is less than or equal to the *Eps* radius when the *Eps*-neighborhood of a point is represented by Eq. (1), then it can be defined as "point *q* is the *Eps*-neighborhood of point *p*."

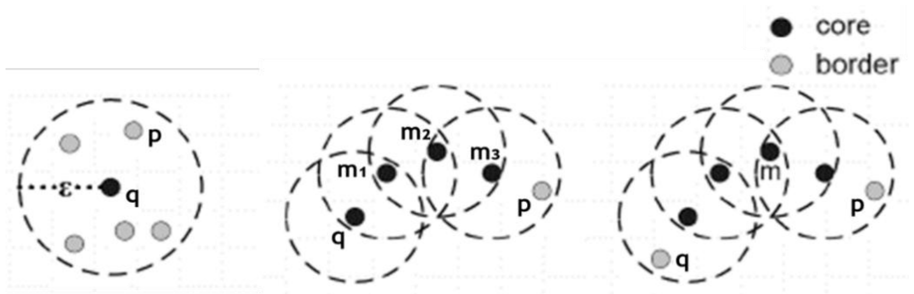

Figure 2: Concepts of (a) directly density-reachability, (b) density-reachability and (c) density-connectedness to determine whether the objects are density-connected [11]

Definition 2: (directly density-reachable) When point *p* belongs to a set of neighboring objects of point q (Eq. (2)), point q can be considered as the center object if the *Eps*-neighborhood of point q is equal to or greater than *MinPts* as given by Eq. (3). Therefore, in this case, the relationship between the entities can be defined as "point p is directly density-reachable from point q"(see Figure 2 (a)) [Figure 2].

$p \in N_{Eps}(q)$(2)

$|N_{Eps}(q)| \geq MinPts$(Core point condition) (3)

Definition 3: (density-reachable) "One point p is density-reachable from another point q" means that there is a direct density-reachable connection between the two points. For example, as shown in Figure 2 (b), if point$m_1$is directly-reachable from point q, and$m_2$is from m_1,$m_3$is fromm_2, and p is fromm_3, point p can be defined as the density-reachable from point q[Figure 2]. It should be noted that even if the density of q can reach *p*, the inverse cannot be guaranteed.

Definition 4: (density-connected) "One point *p* and the other point *q* are density-connected" means that points *p* and *q* can reach the density based on a certain point. For example, as shown in Figure 2 (c), point *p* can reach the density from *m*. Similarly, point *q* can also reach the density from point *m*[Figure 2]. Thus, point *p* is density-connected from point *q*, and its inverse is established.

Definition 5: (cluster) When there are arbitrary points *p*, *q*, point *q* is also included in the cluster if point *q* can reach the density from *p*. If points *p*, *q* belong to the cluster, points *p* and *q* are density-connectable. Therefore, a cluster is a set of density-connected points.

Definition 6: (noise) This indicates an exclusion from a cluster. If there are one or more clusters(C_i) in dataset *D*, the noise point belongs to set *D* but does not belong to any cluster. This can be expressed as follows.

$$noise = \{p \in D \mid \forall i : p \notin C_i \quad (4)$$

Lemma 1: Let O be a single cluster if$O = \{o|o \in O\}$, and the points *o* of the set *O* are able to reach the density from point *p*, belonging to the dataset *D* and satisfying Eq.(3).

Lemma 2: If there is a core point in cluster *C*, the set *O* consisting of points *o* reaching the density from point *p* can be said to be the same as cluster *C*.

As described above, DBSCAN can easily distinguish the clusters of various shapes and sizes by creating the clusters and excluding the core points and border points from a given dataset.

2.2. Extended Study of DBSCAN

The DBSCAN method mentioned above has great advantages in terms of cluster creation. However, it exhibits limitations with the shape of the clusters, which can change widely based on how the parameters such as *Eps* and *MinPts* are selected. As a result, the rate of cluster recognition is relatively slow due to the density difference between various groups [7]. In order to solve these problems, several studies have attempted to reflect both the position values and the unique attribute values of the points at the time of creating the clusters.

DBSCAN-W[12] creates clusters based on the weight of the data whereas DBSCAN focuses on the location of the data. DBSCAN-W considers other attribute values of the data in addition to the location attribute. Several concepts related to DBSCAN are redefined hereinafter. Every object in a set of DBSCAN-W has an area represented by circles of different sizes according to the importance of the object. In other words, when expressing an object in space, the difference of the property value around the position of the object is expressed by the radius of the circle. Therefore, objects are represented by circles of different sizes depending on the property values. When there is an object *p*, the *Eps*-neighborhood is defined as the set of neighbors whose object regions overlap within the *Eps* radius from *p*.

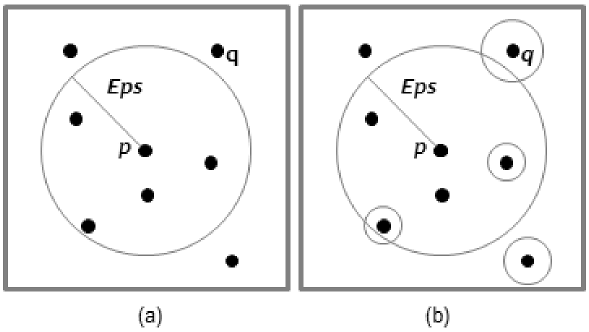

Figure 3: Eps-neighborhood of object pin DBSCAN and DBSCAN-W. (a) Eps-neighborhood of object p in DBSCAN, and (b) Eps-neighborhood of object p in DBSCAN-W

[Figure 3] compares the process of determining the *Eps*-neighborhood of DBSCAN and DBSCAN-W. This shows the process of finding the *Eps*-neighborhood of point *p* in DBSCAN. This neighborhood is identified by the four points contained within the *Eps* radius around point *p*. Figure 3 (b) shows the process of determining the *Eps*-neighborhood in DBSCAN-W. Here, each point is represented by circles of different sizes after being preprocessed. In this figure, point *q* is more than the *Eps* distance from point *p*. However, as the area of *q* overlaps the area of the *Eps* radius around point *p*, it is included in the *Eps*-neighborhood of point *p*. By using this method of expressing different regions of the data in accordance with the difference of their attribute values, the probability of categorizing important data as noise is reduced. As a result, this enhances the possibility of including credible neighbors.

DBSCAN-SI[13] is an algorithm that creates the clusters through two methods. First, DBSCAN-SI(1) extends the *Eps* radius, and increases the probability that the key values become neighbors of neighboring objects. Second, DBSCAN-SI(2) determines the central object of the cluster as the sum of the influence of neighbors. In DBSCAN-SI(1), the two concepts are redefined to determine the neighboring objects. The first concept conveys that the length of *Eps'* defining the neighborhood of a point *p* is the sum of *Eps*, and the radius (the influence of object *p*) of point *p*. The second concept outlines that the *Eps'*-neighborhood of a point *p* is a set of points where the circles represented by the area of radius *Eps'* from *p*, and the influences of each point overlap.

Unlike DBSCAN where a central object is determined by the number of neighboring objects, DBSCAN-SI(2) represents the influence of many properties of the object as values, and uses the sum to determine the center object. *MinPts* implies that the minimum number of neighbors, and *MinInf* indicates the sum of the influences of the minimum neighbors. In this case, if the number of *Eps'*-neighborhoods is larger than *MinPts*, or if the sum of the influences of the minimum neighbors is higher than *MinInf*, this point is called the center object. It also includes its own influence value. In other words, when there is an arbitrary point, if the influence sum of the neighbors is greater than the set reference value, the point is determined as the central object. Therefore, if the number of neighbors is small, the sum of the influences of the neighbors is greater than the set threshold.

[Figure 4] shows the influence values of the center object and the neighboring objects included in the *Eps* radius. Figure 4(a) contains four objects in the neighborhood of the *Eps* radius based on pointp_1. The sum of the influences of the neighboring objects and the center point is 18. In Figure 4(b), the number of the neighboring objects in the radius is three, and the sum of the influences of the neighboring objects including the center object is 32. If the condition of the central object is that the number of neighboring objects is four or more, or the sum of the influence values of the neighboring objects is 30 or more, point p_2 in Figure 4(b) becomes the central object. This is effective in a sense that even if the number of neighboring objects is smaller than *MinPts*, if the objects are heavy (the influence value is large), they become a central object, and are included in the cluster. In addition, even if *MinPts* is maximized and the number of neighboring objects is not limited, the center object can be selected through the sum of influences of the neighboring objects.

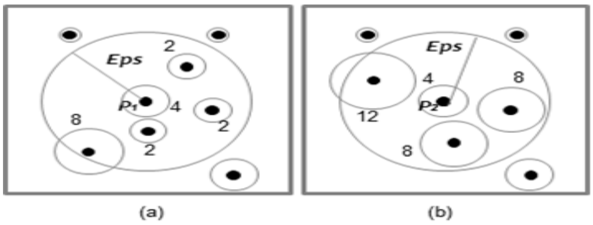

Figure 4: DBSCAN-SI(2)[13]. (a) Core point in DBSCAN-W (*MinPts*=4), and (b) Core point in DBSCAN-SI (*MinPts*=3)

3. Density-Based Clustering Method Considering Directionality

3.1. DBSCAN-D

In our previous study, we proposed the DBSCAN-D method, which is based on variations in the conventional DBSCAN methods such as DNSCAN-W and DNSCAN-SI. However, unlike these, DBSCAN-D considers clustering as a set of data that can express directionality as an attribute. Therefore, this method creates clusters by targeting GPS trajectory data.

DBSCAN-D preferentially analyzes the attributes of the objects in order to determine the area of each object in the GPS dataset. In a set of data that occurs sequentially with a time attribute (as in GPS data), the object can be expressed by the influence or weight differently by using the time difference between the objects. The area of each object extracts the property of a set of location data (such as GPS data), and expresses its size with the importance or value of each object. The GPS data format generally contains information such as longitude/latitude, UTC Time, N/S–E/W indicator, and altitude. In the proposed algorithm, we used the time difference between the GPS reception data to determine the area of the object. The time of staying at a specific point on one space is used as the size of the object. Because GPS data are accumulated sequentially, the time of reception fortwo consecutive points can be used to identify the time difference and the staying time at one point. Consequently, the time of staying at all the locations can be determined excluding the last location received from the GPS dataset.

Another attribute of clustering using DBSCAN-D is directionality. This is different from orientation, which is a unique attribute of GPS data. This directionality can be found through the relationship between two points as well as by determining the area of the object described above. In a set of data that occurs sequentially with a time attribute, if point *p* occurs before point *q*, the topology of the influence (area) of point *q* moves as close as possible to the position of point *p*. The range of movement is up to the maximum section where the influence topology of *q* does not deviate from the coordinate value of point *q*.

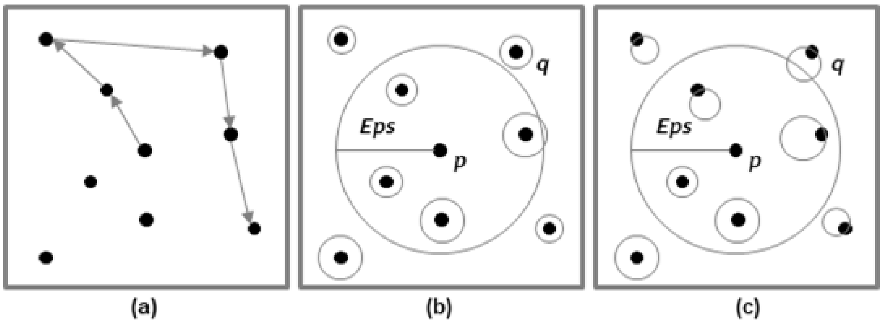

Figure 5: *Eps*-neighborhood of object *p* in DBSCAN-D. (a) Distribution of objects, (b) Weight area according to the staying time of objects in DBSCAN-D, and (c) Directional representation of objects in DBSCAN-D

Figure 5 (a) shows the time sequence of the data with arrows as the data points were received. Figure 5(b) shows the area of the data in proportion to their weights. In this case, the time of staying at that point is shown. Figure 5(c) shows the movement of the object region by applying directionality. If the condition of the central object is *MinPts* five or more within the *Eps* radius, point *p* in Figure 5(b) cannot be the center object because there are four neighboring objects in the radius. In contrast, Figure 5(c) shows that the influential topology of the data generated after point *p* is inclined toward point *p*. Therefore, unlike Figure 5(b), point *q* comes within the *Eps* radius. As a result, the number of neighboring objects is five, and point *p* becomes the central object.

3.2. Clustering with DBSCAN-D

DBSCAN-D uses the weight or influence of objects in the dataset as well as in DBSCAN-W and DBSCAN-SI when performing clustering. The difference is that directionality is taken into consideration during clustering. Thus, it is possible to find feature points such as POIs in a set of location data (as in GPS trajectory data). In this study, we have confirmed the cluster generation process of DBSCAN-D using actual trajectory data—as proposed in our previous study.

The GPS trajectory dataset used in this study was collected from the (Microsoft Research Asia) Geolife project[14-16]. This was gathered from 182 different users over a period of three years (from 2007 to 2012). This dataset contains 17,621 trajectories with a total distance of about 1.2 million km, and a total duration of 48,000+ h. This dataset recoded a broad range of users' outdoor movements, including daily routines such as going to and from home/work, entertainments and sports activities including shopping, sightseeing, dining, hiking and cycling. This trajectory dataset can be used in many fields of research such as mobility pattern mining, user activity recognition, location-based social networking, location privacy, and location recommendations[17].

A GPS trajectory of this dataset is represented by a sequence of time-stamped points, each of which contains the information of latitude, longitude, and altitude as follows.

39.984621, 116.313941, 0, 121, 39744.1219328704, 2008-10-23, 02:55:35

Detailed field information is shown in Table 1 below [Table 1].

Table 1: Geolife GPS trajectory data format [14-16]

Field 1	Latitude in decimal degrees
Field 2	Longitude in decimal degrees
Field 3	All set to 0 for this dataset
Field 4	Altitude in feet (-777 if not valid)
Field 5	Date – number of days (with fractional part) that have passed since 12/30/1899
Field 6	Date as a string
Field 7	Time as a string

The decimal degrees express latitude and longitude of geographic coordinates. Such decimal fractions are used in many geographic information systems, web mapping applications, and GPS devices [18]. Geolife data have resolution of 0.000001 (decimal degree having 6 decimal places). This decimal degree has an accuracy of 111.32 mm (N/S or E/W at equator), 102.47 mm (E/W at 23 N/S), 78.71 mm (E/W at 45/S), and 43.496 mm (E/W at 67 N/S). In other words, the difference of 0.000001 is only about 100 mm in N/S or E/W at equator. Therefore, if clustering is performed using such datasets, the quality of clustering cannot be guaranteed as it is likely that it will have data with excessive density. In addition, Geolife data have a constant GPS reception time interval. There is a problem of representing the weight or influence of the object by using the time difference of the objects. Therefore, we reduced the decimal degree of Geolife data from 0.000001 to 0.00001. By doing so, we were able to mitigate the density problem and express the area of the objects through time information.

We used DBSCAN-D to perform clustering with 300 sequentially received data from the GPS trajectory dataset. As a result, Figure 6 (a) shows three clusters created using the weights that can be obtained by the time difference of the dataset. Figure 6(b) shows that 15 clusters are added as a result of applying the directionality between objects.

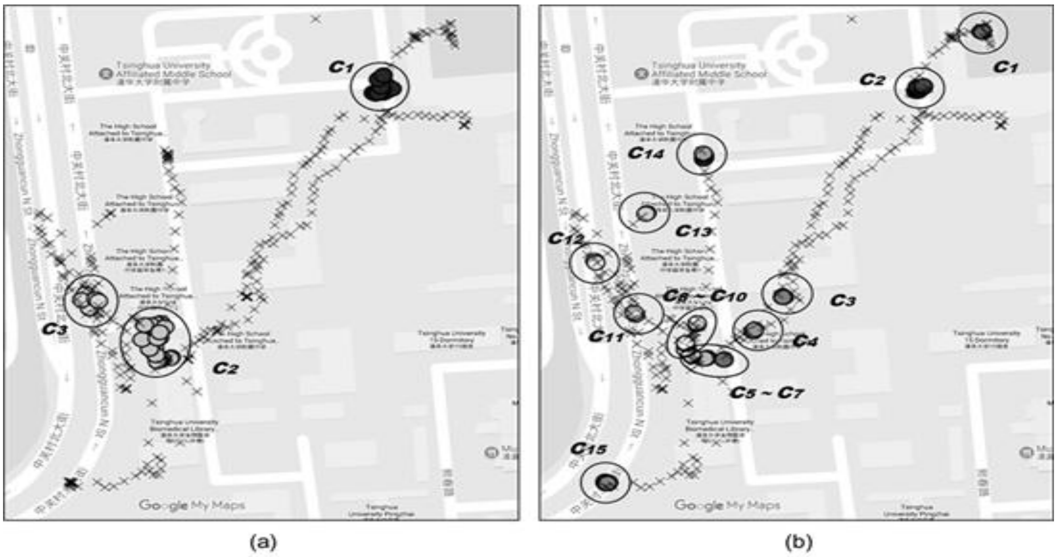

Figure 6: GPS trajectory data-clustering using DBSCAN-D. (a) Apply objects area only to cluster creation (3 clusters), and (b) Apply objects area and directionality to cluster creation (15 clusters)

4. Conclusion

In our previous study, we proposed the DBSCAN-D method to identify suitable POIs by analyzing the GPS data associated with the moving objects. The proposed method was developed by using the directionality and the staying time by mining the GPS data patterns. The staying time was found from the difference in intervals between consecutive GPS data captures. The directionality was mined from the moving patterns in sequentially generated GPS data. These two datasets were good indicators of finding POIs of the moving objects. We applied this method to perform clustering on trajectory data of moving objects. As a result, we confirmed that the object directionality is a useful factor while extracting POIs of moving objects. Unlike the previous studies, this study applied the proposed method to actual location data and examined its usability. We have thus far analyzed a set of real-world observations. Generalizations of such observations are expected in future studies.

References

[1] Mudgil S, Nambula H, Bharathi B. Location Based Services with Location Centric Profiles. International Journal of Electrical and Computer Engineering. 2016 Dec;6(6):3001-5.

[2] Adams PM, Ashwell GWB, Baxter R. Location-Based Services-an overview of the standards. BT Technology Journal 2003 Jan;21(1):34-43.

[3] Wikipedia. Point of interest [Internet]. 2018 [updated 2018 Mar 7; cited 2018 Aug 2]. Available from: https://en.wikipedia.org/wiki/Point_of_interest

[4] Heo YK, Oh JS, Paudel P, Thapa P, Jeon HJ, Jeong MA, et al. Density Based system for Recommendation of Hybrid POI. Proceeding of the Conference of the Institute of Electronics Engineers of Korea. 2015 June:1318-1322. Available from: http://www.dbpia.co.kr/Journal/ArticleDetail/NODE06385314

[5] Khetarpaul S, Chauhan R, Gupta SK, Subramaniam LV, Nambiar U. Mining GPS data to determine interesting locations. In Proceedings of the 8th International Workshop on Information Integration on the Web: in conjunction with WWW 2011. 2011 Mar:8.

[6] Dou AJ, Kalogeraki V, Gunopulos D, Mielikinen T, Tuulos V, Foley S, et al. Data clustering on a network of mobile smartphones. 2011 IEEE/IPSJ International Symposium on Applications and the Internet. 2011 Jul:118-127.

[7] Kirmse A, Udeshi T, Bellver P, Shuma J. Extracting patterns from location history. Proceeding of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. 2011 Nov:397-400.

[8] Ester M, Kriegel H, Sander J, Xu X. A density-based algorithm for discovering clusters in large geospatial databases with noise. Proceeding of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). 1996:226-231.

[9] Lim J, Kook J, Kim J. DBSCAN-D: A Density-Based Clustering Method of Directionality. International Journal of Applied Engineering Research. 2017;12(13):3927-3932.

[10] Santhisree K, Damodaram A, Appaji SV, NagarjunaDevi D. Web usage data clustering using DBSCAN algorithm and set similarities. 2010 International Conference on Data Storage and Data Engineering. 2010 Feb:220-224. DOI: 10.1109/DSDE.2010.14.

[11] Schlitter N, Falkowski T, Lässig J. DenGraph-HO: Density-based Hierarchical Community Detection for Explorative Visual Network Analysis. Research and Development in Intelligent Systems XXVIII. 2011 Oct:283-296.

[12] Kim HS, Lim HS, Yong HS. Design and development of the clustering algorithm considering weight in spatial data mining. Journal of Intelligence and Information Systems. 2002 Dec;8(2):177-187. Available from: http://www.jiisonline.org/

[13] Kim B. Design and Development of Clustering Algorithm Considering Influences of Spatial Objects. The Journal of the Korea Contents Association. 2006 Dec;6(12):113-120.

[14] Zheng Y, Xie X, Ma WY. Geolife: A collaborative social networking service among user, location and trajectory. IEEE Data Engineering Bulletin. 2010 June;33(2):32-39.

[15] Zheng Y, Zhang L, Xie X, Ma WY. Mining interesting locations and travel sequences from GPS trajectories. Proceedings of the 18th international conference on World Wild Web. 2009 Apr:791-800. DOI: 10.1145/1526709.1526816.

[16] Zheng Y, Li Q, Chen Y, Xie X, Ma WY. Understanding Mobility Based on GPS Data. Proceedings of ACM conference on Ubiquitous Computing. 2008 Sep:312-321. DOI: 10.1145/1409635.1409677.

[17] Microsoft. GeoLife GPS Trajectories [Internet]. 2012 [updated 2012 Aug 9; cited 2018 Aug 2]. Available from: https://www.microsoft.com/en-us/download/details.aspx?id=52367&from=https%3A%2F%2Fresearch.microsoft.com%2Fen-us%2Fdownloads%2Fb16d359d-d164-469e-9fd4-daa38f2b2e13%2F&751be11f-ede8-5a0c-058c-2ee190a24fa6=True

[18] Jiang H, Yu Q, Liu C, Zhu Q, Guo L. The Analysis of CRM Customer Information Based on Data Mining. Proceeding of the Ninth International Conference on Natural Computation. 2013 Jul:978-983. DOI: 10.1109/ICNC.2013.6818118.