International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 13 (2017) pp. 3927-3932
© Research India Publications. http://www.ripublication.com

DBSCAN-D: A Density-Based Clustering Method of Directionality

Joasang Lim[1], Joongjin Kook[2] and Jinman Kim[3]*

[1]Intelligent Engineering Informatics for Human, Sangmyung University, Seoul, Korea.

[2]Korea Electronics Technology Institute, Seoul, Korea.

[3]Intelligent Information Technology Research Institute, Sangmyung University, Seoul, Korea.

*(*Corresponding Author)*

Abstract

This research proposed DBSCAN-D, which is a clustering technique for locating POI based on the prior density-based clustering studies. Our method analyzed 'staying time' and 'directionality' of the GPS track logs. The staying time was the intervals between two locations where GPS was obtained. Directionality was the direction toward the coming position from the previous one, obtaining by analyzing the sequentially generated GPS data. The proposed DBSCAN-D used these two attributes to cluster the POIs from the routes of the moving objects.

Keyword: Clustering, DBSCAN, Density, POI, GPS

INTRODUCTION

Recently, the spread of mobile terminals such as smart phones and the development of communication techniques such as GPS (Global Positioning Systems) and WSN (Wireless Sensor Networks) has enabled many different ways of acquiring the tracking information of a moving object. The moving object may easily produce the spatio-temporal data set of positions and locational shapes continuously over time. Moreover, with the increasing use of location-based services (LBS) in various applications, interests have been growing to extract meaningful information from these logs.

The location-based service tracks the up-to-date positions of a mobile terminal using a mobile communication network or a satellite signal, and provides various contextual services. So, normally included in the system nowadays are wireless positioning gadgets for locating mobile terminals, LBS server technology for providing core infrastructure technology for services, and various LBS applications [1].

Currently, many smartphone applications provide services based on user`s location information directly or indirectly. These services mostly use POI information. POI (Point Of Interest) is a combination of two things: (1) a real world in which a person is interested, or a specific location on a map or a drawing, (2) the location information of a nearby road building [2]. The most common way of expressing a POI is to select locations either well-known or where many people are gathered. The next step is to extract the POI directly from location data such as GPS. Also, the method is existed for expressing the POI by combination these two methods [3].

In previous works, the method for finding meaningful places in trajectory data of moving objects has performed such as the distance-based clustering using mean shift [4] and clustering using K-means [5]. The previous clustering has been most commonly performed using data density. The density-based technique assumes that clusters are regions of high density separated each other in space. In other words, the clusters can be identified by looking at their density and if they can be separable from each other. Through the above mentioned method, clusters of various shapes and sizes can be easily extracted. However, the density-based cluster method has two problems. Firstly, the density-based cluster method varies in shape depending on the selection of parameters used for cluster determination. Moreover, when a group with density difference exists in a data set, the probability that a group with a relatively high density is recognized as a cluster can be lowered [6].

DBSCAN [7] is a representative technique of density-based clustering. Since the introduction of this technique, many studies have attempted to solve the problems of density-based clustering. However, most studies focused on cluster extraction and how to create clusters in a data set. We attempted to overcome the cluster extraction problems and suggested a newly designed clustering method of DBSCAN-D to find POIs from the location data sets generated from moving objects. The proposed method used the spatio-tempral data sets of GPS routes with directionality.

RELATED WORK

A. DBSCAN

DBSCAN was first proposed by Ester, Kriegel, Sander, and Xu [7] as a density-based cluster technique. It was assumed that the cluster is a high-density region separated by the object density region in the space, and the cluster was 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 and medical as well as financial and marketing fields such as analyzing characteristics of customers with similar characteristics or purchasing pattern related to financial fraud [8].

DBSCAN expresses the given data as a vector, and uses the density between the vectors to identify clusters and data points that are not in any clusters, thus properly separating the clusters. The terms used in this technique are as follows.

- *Eps* (epsilon): This parameter specifies the radius around a point to measure the density of any point. It can be expressed in Greek letters like 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 the point *p* is called the 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 means all points not included in the cluster.

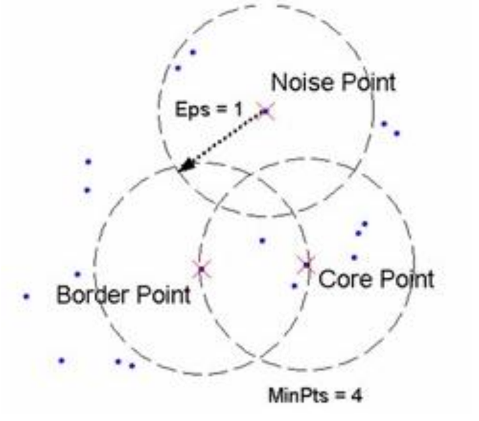

Figure 1. The classification of certain objects(points) in DBSCAN.

The paper [7] describes six definitions and two lemmas for generating clusters in the density domain. This is 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.

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

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

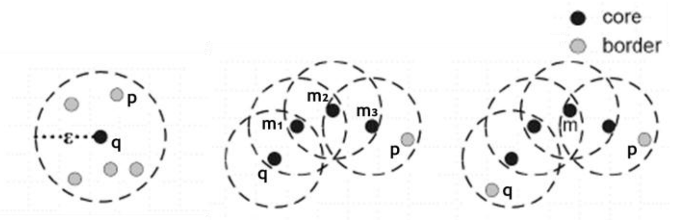

Figure 2. The concepts (a) directly density reachability, (b) density reachability and (c) density connectedness to determine whether objects are density connected[9].

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

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

$$|N_{Eps}(q)| \geq MinPts \text{ (core point condition)} \qquad (3)$$

Definition 3: (density-reachable) ‘One point p is density-reachable from one point q’ means that there is a direct density reachable connection between 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 from m_2, and p is each able to reach a density from m_3, the point p can be defined as the density reachable from the point q. 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 are points that can reach the density based on a certain point. For example, as shown in Figure 2 (c), the point *p* can reach the density from *m*. Similarly, the point *q* can also reach the density from the point *m*. Thus, the point *p* is densely connected from the point *q* and its inverse is established.

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

Definition 6: (noise) This means that it is not included in the 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 clusters. This can be expressed as follows.

$$\text{noise} = \{p \in D | \forall i : p \notin C_i\} \qquad (4)$$

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

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

As described above, DBSCAN is easy to distinguish clusters of various shapes and sizes by creating clusters by excluding core points and border points from a given data set.

B. Extended study of DBSCAN

The DBSCAN mentioned above has great advantages in terms of cluster creation. However, there is a problem that the shape of clusters is rapidly changed according to the selection of parameters such as *Eps* and *MinPts*, and the cluster recognition rate is relatively decreased due to the density difference between groups [6].

DBSCAN-W

In the paper [10], they proposed DBSCAN-W(a DBSCAN algorithm using region expressed as Weight) which takes into account the weight of data to create clusters. In DBSCAN, all the points or objects represented in space are expressed as having only the location attribute. That is, other attribute values of the respective data are not considered. DBSCAN-W considers other attribute values of data in addition to the location attribute of data in clustering. Based on the definition of DBSCAN, several concepts such as the following are redefined.

Definition 1: Every object in the set has an area represented by circles of different sizes according to the importance of the object in the application system. 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.

Definition 2: When there is an object *p*, the *Eps*-neighborhood is a set of neighbors that overlap each other in the *Eps* radius from the center of *p*.

Definition 3: Clusters are represented by the maximum set of dense connected regions.

DBSCAN-W performs three pre-processing steps as follows to represent the attributes of the entity. First, it determines the property of the entity, that is, the non-spatial property *A* to be weighted. Second, a radius value (r_i) to be used as an attribute size of each entity is determined through an appropriate deformation function $F(A_i) = r_i$. Third, the position of the object is represented on the space, and each object is represented by a circle through the radius r_i determined in the second step.

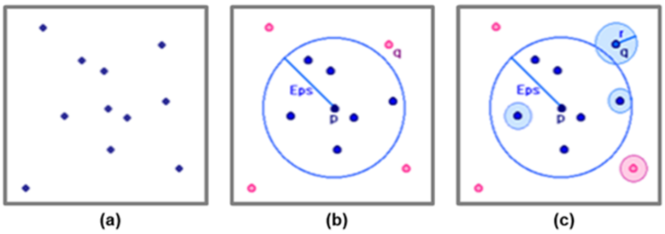

Figure 3. Eps-neighborhood of object *p* in DBSCAN and DBSCAN-W[11]. (a) Distribution of objects, (b) Eps-neighborhood of object *p* in DBSCAN, and (c) Eps-neighborhood of object *p* in DBSCAN-W.

DBSCAN-W cluster creation process is similar to DBSCAN. However, unlike DBSCAN, DBSCAN-W finds all density-reachable points and determines them as one cluster using the same method as defined in 2 above. Figure 3 compares the process of determining the *Eps*-neighborhood of DBSCAN and DBSCAN-W when individuals with the same distribution as 3(a) are located in space. Figure 3(b) shows the process of finding *Eps*-neighborhood of point *p* in DBSCAN. The *Eps*-neighborhood of point *p* is the five points contained within the

Eps radius around the point *p*. Figure 3(c) is the process of determining the *Eps*-neighborhood in DBSCAN-W, where each point is represented by circles of different sizes after preprocessing. In this figure, the point *q* is more than *Eps* the distance from the center point of the point *p*. However, since the area of *q* overlaps the area of the *Eps* radius around the point *p*, it is included in the *Eps*-neighborhood of the point *p*. In this way, by using a method of expressing different regions of the data in accordance with the difference of the attribute values of the data, the probability of handling important data as noise is reduced, thereby enhancing the possibility of including neighbors.

DBSCAN-SI

DBSCAN-SI [12] is an algorithm that creates clusters through two methods. One of the two methods, DBSCAN-SI(1), is to extend the Eps radius and increase the probability that the influential values become neighbor of neighboring objects. The other, DBSCAN-SI(2), is the method of determining the central object of the cluster as the sum of the influence of neighbors.

In DBSCAN-SI(1), the following two concepts are redefined to determine neighbor objects.

Definition 1: The length of Eps' defining the neighborhood of a point *p* is the sum of *Eps* defined by DBSCAN and DBSCAN-W and the radius (r_p) (the influence of object *p*) of the center point.

$$Eps' = Eps + r_p \tag{5}$$

Definition 2: The Eps'-neighborhood of a point *p* is a set of points in which circles represented by the area in radius Eps' from *p* and the influences of each point overlap. Here, the $dist(\,)$ is a function to obtain the Euclidean distance in space.

$$\begin{aligned} N_{Eps}(p) &= \{q \in D \mid dist(p,q) \leq (Eps + r_p + r_q)\} \\ &- Eps\colon dist(p,q) \leq Eps + r_p + r_q \qquad (6) \\ &- Eps'\colon dist(p,q) \leq Eps' + r_q \end{aligned}$$

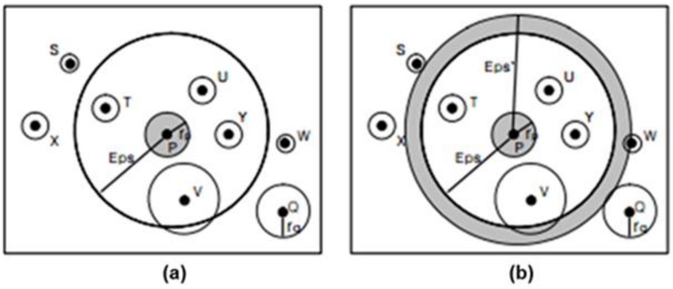

Figure 4. Eps extension[12]. (a) *Eps* in DBSCAN-W, (b) *Eps* in DBSCAN-SI(1).

Figure 4(a) shows *Eps* in DBSCAN-W and shows distance from object *p*. In this figure, it can be seen that T, U, Y, and V are neighboring objects within the *Eps* radius based on the center object *p*. Figure 4(b) shows that *Eps* is an extension of *Eps* proportional to the influence of object *p*. As a result, unlike Figure 4(a), W, Q, and S are included as neighbor objects in addition to T, U, Y, and V. That is, the objects W, Q, and S are outside the *Eps* radius, but can be included in the neighbor object because they extend *Eps*(*Eps'*) in proportion to the radius length(influence) of the central object *p*.

Unlike DBSCAN, in which 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. 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 neighbors is greater than the set threshold. In addition to the DBSCAN definition, the following definitions are given.

Definition 3: *MinPts* means the minimum number of neighbors, and *MinInf* means the sum of the influences of the minimum neighbors. In this case, if the number of *Eps'*-neighborhoods is more than *MinPts*, or if the sum of the influences of the minimum neighbors is more than *MinInf*, this point is called the center object. It also includes its own influence value.

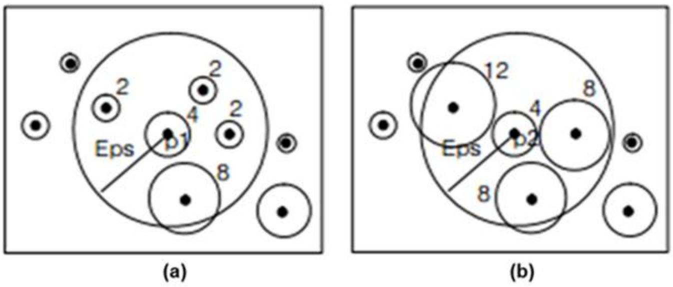

Figure 5. DBSCAN-SI[12]. (a) core point in DBSCAN-W(*MinPts* = 4), (b) core point in DBSCAN-SI(*MinPts* = 3).

Figure 5 shows the influence values of the center object and neighboring objects included in the *Eps* radius. In this figure, (a) contains four objects in the neighborhood of the *Eps* radius based on the point *p*. Here, the sum of the influences of the neighboring objects and the center point is 18. In (b), the number of neighboring objects in the radius is 3, and the sum of the influences of the neighboring objects including the center object is 32. Here, if the condition of the central object is that the number of neighboring objects is 4 or more, or the sum of the influence values of neighboring objects is 30 or more, the point *p* in FIG. 5 (b) becomes the central object. This is effective in 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 neighboring objects.

DENSITY-BASED CLUSTERING METHOD CONSIDERING DIRECTION

DBSCAN-D

The proposed DBSCAN-D method is based on DBSCAN such as DNSCAN-W and DNSCAN-SI. These three algorithms perform clustering by finding dense connected relation based on a central object according to their definition in a set when a specific data set is given regardless of data characteristics or domains. Although the proposed algorithm was based on DBSCAN, the location-based service domain that deals with spatial data is applied. Because the clustering considers a set of data that can be expressed directionality as attributes. Therefore, in this paper, we proposed the algorithm that can be performed clustering on GPS data.

DBSCAN-D was preceded by an analysis of the attributes of the data for determining the size of data before clustering similar to DBSCAN-extended algorithms that do not take into account the importance of data.

Definition 1: In a set of data that occurs sequentially with a time attribute, data can be expressed by the influence or weight differently using time difference between data.

The data size extracts the properties of the location data set such as GPS and expresses the importance or value of each data by their size. The GPS data is represented by additional information such as basic location information as longitude / latitude, UTC Time, N/S, E/W Indicator, Satellites Used, and Altitude. In the proposed algorithm, UTC time difference between GPS reception data is used to determine the size of data. That is, the time of staying at a specific point on one space is used as the size of the point. Because GPS data accumulates data sequentially, knowing the time at which one point and the next point are received can be used to fine the difference and stay at one point. Consequently, the time of staying at all locations can be known excluding the last location received from the GPS data set. The above mentioned can be represented in various sizes as shown in Figure 3(c).

Another of the data attributes used for clustering is directivity in DBSCAN-D. That is different from the orientation, which is a unique property of GPS data. The orientation can be found through the relationship between the two points as well as determining the size described above. The definitions are as follows.

Definition 2: In a set of data that occurs sequentially with a time attribute, if the point *p* was occurred before the point *q*, the phase of the influence of the point *p* moves as close as possible to the position of the point *p*. In here, the range of movement is up to the maximum interval in which the influence phase of *q* does not deviate from the coordinate value of point *q*.

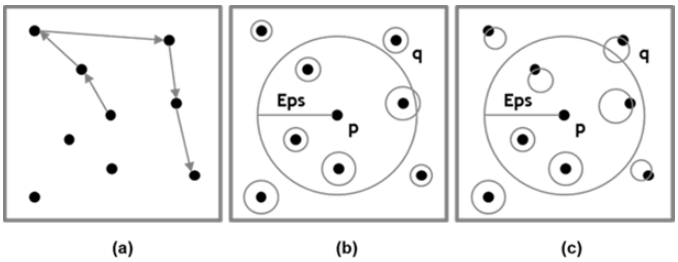

Figure 6. *Eps*-neighborhood of object *p* in DBSCAN-D. (a) Distribution of objects, (b) Weight area according to the staying time of objects in DBSCAN-D, and (c) Directional representation of objects in DBSCAN-D.

Figure 6(a) shows the order in which some data are generated through arrows as GPS data sets. Figure 6(b) shows the size of data in proportion to the weight of each data, in this case, the time of staying at that point by definition of DBSCAN-W. Figure 6(c) shows that the area represented by the size of the data is shifted by applying directionality. In here, the movement is performed until the position of the point lies around the circle in the reference point direction from the size of the points generated after that point with respect to one point according to definition 2. Therefore, if the condition of the central object is *MinPts* in *Eps* radius is 5 or more, Fig 6(b) cannot be the central object with 4 neighboring objects included in the radius.

CONCLUSION

In this paper, we proposed the DBSCAN-D method to find suitable POIs by analyzing GPS tracks of the moving objects. The proposed method was developed by using the direction and staying time by mining the patterns existed in the GPS data. The staying time was the difference in intervals when GPS data were captured. The directionality was mined from the moving patterns in sequentially generated GPS data, heading towards next stops. These two data sets were good indicators to find POIs of the moving objects.

The proposed scheme may be limited in that the sample we analyzed was neither sufficient enough nor based on real-world observations Thus caution should be taken to generalize the results of our study. In future works, we will examine the usefulness of the proposed method by applying it to actual location data. Also, we will elaborate more on the directionality of temporality with a more theoretical background.

ACKNOWLEDGMENT

This research was supported by the Research Program funded by the Sangmyung University (2014).

REFERENCES

[1] H. O. Choi, "LBS, Location-Based Services," TTA Journal, vol. 86, pp. 59-69, 2003.

[2] G. W. Lee and H. U. Son, Glossary of Geo Spatial Information System, 1th ed, Goomi Seokuan, Seoul, 2016.

[3] Y. K. Heo, J. S. Oh, P. Paudel, P. Thapa, H. J. Jeon, M. A. Jeong, and S. R. Lee, "Density Based system for Recommendation of Hybrid POI," Proceeding of the Conference of the Institute of Electronics Engineers of Korea, pp, 1318-1322, 2015.

[4] S. Khetarpaul, R. Chauhan, S. K. Gupta, L. V. Subramaniam, and U. Nambiar, "Mining GPS data to determine interesting locations," Proceeding of the 8th International Workshop on Information Integration on the Web: in conjunction with WWW 2011, ACM, p. 8, 2011.

[5] A. J. Dou, V. Kalogeraki, D. Gunopulos, T. Mielikinen, V. Tuulos, S. Foley, and C. Yu, "Data clustering on a network of mobile smartphones," Proceeding of the IEEE/IPSJ 11th International Symposium, IEEE, pp. 118-127, 2011.

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

[7] M. Ester, H. Kriegel, J. Sander, and X. Xu, "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), Portland, pp. 226–231, 1996.

[8] K. Santhisree, A. Damodaram, S. V. Appaji, and D. NagarjunaDevi, "Web usage data clustering using DBSCAN algorithm and set similarities," Proceeding of the Data Storage and Data Engineering (DSDE) 2010 International Conference, IEEE, pp. 220-224, 2010.

[9] N. Schlitter, T. Falkowski, and J. Lässig, "Dengraph-ho: Density-based hierarchical community detection for explorative visual network analysis." Research and Development in Intelligent Systems XXVIII. Springer London, pp. 283-296, 2011.

[10] H. S. Kim, H. S. Lim, and H. S. Yong, "Design and development of the clustering algorithm considering weight in spatial data mining," Journal of Intelligence and Information Systems, Vol. 8, No. 2, pp. 177-187, 2002.

[11] H. S. Lim, A Density-based Spatial Clustering Algorithm Considering Weight and Obstructed Distance, Master's Thesis of Ewha Institute of Science and Technology, 2002.

[12] B. C. Kim, "Design and Development of Clustering Algorithm Considering Influences of Spatial Objects," Journal of The Korea Contents Association, Vol. 6, No. 12, pp. 113-120, 2006