A Quantitative Analysis of Big Data Clustering Algorithms for Market Segmentation in Hospitality Industry

Avishek Bose[*], Arslan Munir[†], and Neda Shabani[‡]

[*][†]Department of Computer Science
[‡]Department of Hospitality Management
Kansas State University, Manhattan, KS, USA
Email: [*]abose@ksu.edu, [†]amunir@ksu.edu, [‡]nshabani@ksu.edu

Abstract—The hospitality industry is one of the data-rich industries that receives huge *Volumes* of data streaming at high *Velocity* with considerably *Variety*, *Veracity*, and *Variability*. These properties make the data analysis in the hospitality industry a *big data* problem. Meeting the customers' expectations is a key factor in the hospitality industry to grasp the customers' loyalty. To achieve this goal, marketing professionals in this industry actively look for ways to utilize their data in the best possible manner and advance their data analytic solutions, such as identifying a unique market segmentation clustering and developing a recommendation system. In this paper, we present a comprehensive literature review of existing big data clustering algorithms and their advantages and disadvantages for various use cases. We implement the existing big data clustering algorithms and provide a quantitative comparison of the performance of different clustering algorithms for different scenarios. We also present our insights and recommendations regarding the suitability of different big data clustering algorithms for different use cases. These recommendations will be helpful for hoteliers in selecting the appropriate market segmentation clustering algorithm for different clustering datasets to improve the customer experience and maximize the hotel revenue.

Index Terms—Hospitality, market segmentation, density-based clustering, embedded cluster, nested adjacent cluster

I. INTRODUCTION AND MOTIVATION

In recent years, the hospitality industry has emerged as one of the most profitable and vibrant businesses around the world. The hospitality Industry is perceived as the main source of revenue for many countries around the globe today. Many articles [1] [2] have shown that the growth of this industry will increase gradually. As the world steps into the Internet era with widespread utilization of Internet-connected appliances, the hospitality industry has transformed into a vastly data-rich industry [3] [4]. However, a structured way of utilizing available customer data for providing targeted recommendations [5] to customers is still missing. There are several other businesses like e-commerce websites and online stores that provide product recommendations to target potential customers [6] [7] [8] [9]. This trend of providing recommendations, such as customized offers and promotions, to customers via various mediums, such as websites, online social media, TV, and smart phones, is increasing day by day. However, it is infeasible to translate theses existing recommendation systems to the hospitality industry because of the vast scale of the hospitality network (i.e., customers, vendors, and proprietors) and its strict dependence on global economic trends. Furthermore, the hospitality industry requires an automated and dynamic recom-mendation system that renders many of the existing techniques focusing on offline recommendation systems ineffective.

An effective recommendation system for the hospitality industry can help hoteliers to better meet customer preferences thus resulting in increased customer satisfaction as well as overall increase in hotel revenue. [10] [11] suggest that identifying market segmentation could be the key criterion to driving the hospitality industry forward in this regard. As technologies such as online social media, websites, smartphone, etc., become increasingly prevalent, it is imperative that the hospitality industry also utilize these platforms for providing recommendations, customized offers, and promotions to their customers. Market analysts have identified many aspects, goals, and processes involved in market segmentation for customer recommendation [12]. One of these processes is data clustering [13] which makes market segmentation expedient for market professionals.

Market segmentation for large data volumes can be carried out using big data clustering algorithms. Data clustering is a concept where similar kinds of points or objects of a dataset are grouped to remain in the same class. Thus, the points in the dataset are classified by their proximity to each other based on parameters given to the clustering algorithm. Although several clustering algorithms have been proposed in the literature, there is little to no information available as to the suitability of one algorithm over another with regards to big data clustering in the hospitality industry. As hospitality datasets are significantly heavy featured, a quantitative comparison of different clustering algorithms is imperative for making an informed choice on the appropriate clustering algorithm.

There exist various types of clustering algorithms, namely: (i) centroid-based clustering, (ii) hierarchical clustering, (iii) distribution-based clustering, (iv) density-based clustering, and (v) grid-based clustering. In particular, several papers have discussed efficient density-based algorithms, such as DBSCAN [14], OPTICS [15], EnDBSCN [16], and few other variation of these algorithms [17] [18], however, each of these algorithms has its limitations and weaknesses. We restrict our analysis to density-based algorithms because market segmentation using these algorithms can be done efficiently. Furthermore, density-based algorithms incorporate various significant factors of clusterization, such as the number of actual noise points, number of actual clusters, etc., in the datasets. In general, the efficiency of a big data clustering algorithm is contingent on how many input parameters the algorithm depends on and its

clustering performance in different scenarios, such as varying densities, embedded clusters, and nested adjacent clusters. Fig. 1 shows an embedded cluster and Fig. 2 shows a cluster of varying density.
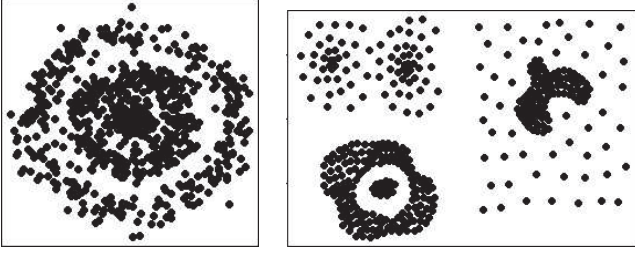

Fig. 1: Embedded cluster.   Fig. 2: Varying density cluster.

Our main contributions in this paper are:

- We have presented a detailed review of various clustering algorithms and classified them based on their usefulness for market segmentation in the hospitality industry for various use cases.
- We have characterized the limitations, performance, complexity, and usefulness of various clustering algorithms for different use cases.
- We have implemented various density-based clustering algorithms, such as DBSCAN, OPTICS, EnDBSCAN, and their variants, and have provided a comparative performance analysis of these clustering algorithms for different data sets.
- Based on our analysis and implementation, we have identified requirements of developing future clustering algorithms for market segmentation in the hospitality industry.

The rest of this paper is organized as follows: Section II discusses the motivation for this work. Section II presents the background study. The literature review is presented in Section III. Section IV presents simulation results and performance analysis. Finally, Section V concludes the work and identifies future research directions.

II. BACKGROUND

In this section, we present basic definitions and ideas related to density-based clustering algorithms.

Definition 3.1: Density-based clustering works by differentiating the density of points in a specific area. For example, the density of one area could be higher than the density of another area based on the number of points present in the specified or given area. Let p is the point in the dataset D, the density of a specified point p is measured by the number of data-points $|N_{\mu(p)}|$ present in p's neighborhood μ. $|N|$ denotes the number of points or objects in the neighborhood of any specific point or object.

Definition 3.2: Neighborhood μ of a point p is considered a circular area generated by a given parameter radius r as an input value, centering at the point p. If any point q from the dataset D is in the circular area of p and their shortest distance is $dist(p,q) \leq r$, it can be said that q

is in the neighborhood of p or in other words point q is point p's neighbor. So, neighborhood of p can be defined as $\mu_p \rightarrow \{q \in D | dist(p,q) \leq r\}$.

Definition 3.3: The number of points that must be present in the neighborhood of a point p to make it as a core-point to form a cluster is referred as $MinPts$. The number, size as well as shape of a cluster is heavily dependent upon this user given parameter. Additionally, except for the border point, a neighborhood of a particular point inside a higher density cluster has more data-points than $MinPts$, but the points inside lower density cluster may have at least the equal number of points as $Minpts$. As $MinPts$ can only be a natural number, therefore $MinPts \in \mathbb{N}$ where $\mathbb{N}$ denotes the set of natural numbers.

Definition 3.4: Core-point or core-object p of a cluster C_k (where $k = \{1,2,3,\ldots,n\}$) are those data-points in the cluster C_k which have equal or greater number of points as $MinPts$ in its neighborhood μ. Core-object $\theta \rightarrow \{p \in C_k \ \ s.t. \ |N_{\mu(p)}| \geq MinPts\}$ where θ refers to the set of all core-points.

Definition 3.5: Border-point or border-object s of a cluster C_k (where $k = \{1,2,3,\ldots,n\}$) is that data-point in the cluster C_k which do not have a sufficient number of points as $MinPts$ in its neighborhood μ, but still those are the member of that cluster. Border point $\lambda \rightarrow \{s \in C_k \ \ s.t. \ |N_{\mu(e)}| < MinPts\}$ where λ refers to the set of all Border points.

Definition 3.6: Noise points are those data-points which are not the member of any cluster. The area containing noise points has a low density of points than the other areas containing clusters. In other words, if any point except the border-point in the dataset doesn't have an equal number of points as $MinPts$ in its neighborhood, this point can be referred as noise. Let the dataset D has n number of clusters represented by the cluster set $Z_c = \{C_1, C_2, \ldots, C_k, \ldots, C_n\}$ where $k, n \in \mathbb{N}$. If ω is noise point, then noise $\omega \rightarrow \{\omega \in D \ \ s.t. \ \forall \ k = \omega \notin Z_c\}$, where ω refers to the set of all the noise points.

Definition 3.7: Core-distance γ of a point p is the minimum distance of neighborhood of the point which contains an equal number of points as $MinPts$ inside its neighborhood. Core distance $\gamma \rightarrow \{|\gamma| \leq r_\mu, |N_{\mu_{\gamma(p)}}| = MinPts\}$. Here r_μ is the given radius of the neighborhood, $\mu_{\gamma(p)}$ is the neighborhood covering p's core-distance, and $|N_{\mu_{\gamma(p)}}|$ is the number of points of that neighborhood.

Definition 3.8: Let p is a core point or object, and q is another point or object in the dataset. Reachability-distance ϕ of the object q is the shortest distance from p if q is directly density-reachable from p. Reachability distance $\phi(q,p) \rightarrow \{q \in |N_{\mu(p)}|, |r_{\mu(p)}| \geq \phi_q \geq \gamma_p\}$. Reachability-distance ϕ cannot be smaller than the core-distance γ. Fig. 3 illustrates core-point p, a point q in the neighborhood of p, core-distance γ_p of p, and reachability distance $\phi(q,p)$ of q from the point p to q.

Definition 3.9: A point q can be said directly density-

reachable from a point p if q is located inside the p's neighborhood μ_p and point p has at least $MinPts$ number of points in its neighborhood. **Definition 3.10:** A point q can be referred as density-reachable from a point p if those are connected via directly density-reachable points. Let a chain of points is p_1, p_2, p_3,...,p_n and any point p_{j+1} of this chain is directly density-reachable from p_j where $j \in \{1, 2, \ldots, n-1\}$. If $p_1 = p$ and $p_n = q$, q is density-reachable from p.
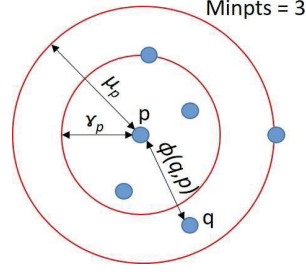

Fig. 3: Illustration of core-object p, core-distance γ_p, and reachability-distance $\phi(q, p)$.

Definition 3.11: A point q can be called density connected to a point p if both p and q are density-reachable from another point o.

III. LITERATURE REVIEW

A. DBSCAN

DBSCAN [14] is one of the significant early approaches in density-based methods to cluster points of a dataset. It works by traversing all the points of a dataset, and picks a point arbitrarily. If p is an arbitrary point selected from a dataset, this algorithm is able to access all the points within the point p's neighborhood . If p is a core object, it can access all neighborhood points and can let the process repeat for its neighbor points to expand a cluster, but this is not true for the border point . While any border point is chosen arbitrarily to access its neighborhood, this algorithm skips that point because it won't satisfy the condition to access the neighbor points due to less number of points than $MinPts$ of its neighborhood. The current cluster id is assigned at that border point and starts accessing next arbitrary point. If the chosen point is a noise, it will not satisfy density-connectivity feature of this algorithm.

DBSCAN faces some difficulties to identify a varying density space because it uses global neighborhood radius r_μ and $MinPts$. That's why it cannot perform well to detect varying density cluster and embedded cluster. If the two clusters are in close proximity or are adjacent to each other, the process may detect those as a single cluster. The same thing may happen if clusters of varying density are located one inside another such as a nested embedded cluster. In the case of detecting nested embedded cluster, outcomes go beyond the performance of this approach. If two adjacent clusters do not have more distance than given neighborhood radius r_μ, it is not possible to make any distinction between two sets of points to identify those as two separate clusters.

Algorithmic Analysis: The runtime complexity of DBSCAN algorithm for each point is the runtime needed for the query to process all the neighbor-points in the neighborhood μ. As this process would be performed for each object of the datasets,

the optimal runtime for DBSCAN algorithm is $O(n \log_k n)$ where n is the number of object in the datasets and k is the number of the core-object. The optimal runtime complexity is only applicable if tree based spatial index is used otherwise the complexity could be $O(n^2)$.

B. OPTICS

OPTICS [15] is another clustering algorithm based on density analysis which orders points by comparing points reachability-distance to the closest core-point that is directly density-reachable from those points to identify cluster. However, this algorithm does not directly identify cluster from the dataset because after ordering of the objects, any density-based clustering approach such as DBSCAN does rest of the task of clustering. According to the procedure of OPTICS algorithm after creating an augmented ordering of cluster points, this approach can be used with any other density-based approaches such as DBSCAN [11]. The ordering stores core-distance and suitable reachability distance for each point. After formulating reachability plot, an optimum neighborhood radius ropt might be selected to generate the proper result of clustering.

Algorithmic Analysis: OPTICS is evolved to encounter the limitation of DBSCAN [14] such as to detect the varying density of cluster objects. It provides a handy solution to meet the problems of global density parameter issue and varying density efficiently. As DBSCAN was vastly dependent upon input parameters like neighborhood radius r_μ and $Minpts$, ordering of objects has minimized the dependency for those parameters in OPTICS. Although this algorithm elaborately discusses visual techniques of cluster ordering, reachability plots, etc. to counter the dependency of input parameter, actually visual technique also need some parameter setting such as the threshold value of neighborhood or optimum neighborhood radius ropt to identify clusters. Afterwards, the process of selecting threshold value can be crucial to identify cluster. If an inappropriate threshold value is selected, some clusters are likely to be undetected, and the algorithm will not be able to detect embedded clusters. Moreover, this process experimented using the specific datasets to get range values, but whether these values are feasible or not for all datasets like hospitality datasets, is not specifically mentioned in this approach.

As OPTICS requires ordering of points as an extra calculation, its complexity is higher than other density based algorithm. If it uses any tree based spatial index, its runtime would be $O(n \log n)$ otherwise it would be $O(n^2)$. Only if the algorithm has direct access to the neighborhood μ or organized in a grid, the runtime requires to cluster from the ordered-dataset is O(n). So, the overall runtime complexity of OPTICS for extracting the clusters from the datasets is at least $O(n \log n) + O(n) = O(n \log n)$.

C. EnDBSCAN

The main idea of EnDBSCAN [16] algorithm is that if the difference of the core distance between two points is in the range of a pre-defined variance factor, both the points are

identified to be in the same cluster. EnDBSCAN also starts clustering by selecting an arbitrary point from a dataset and calculates its core-distance considering the given parameters $MinPts$ and neighborhood radius r_μ. If the point's core-distance is greater than given neighborhood radius r_μ, it is considered as noise point. When core-distance is smaller or equal to the given neighborhood radius r_μ, the point is considered as a core-point. Then the core-point is allowed to expand the cluster through its neighborhood points within the range of its core-distance γ. After assigning a new cluster id to the core point or object, all the core-neighbors of this point are assigned the same cluster id.

The process of expanding and clustering repeats until all the dataset's points have been assessed. To remain in the same cluster, the difference between the core-distance of an initially selected arbitrary point and the core-distances of core-neighbor points of that arbitrary point cannot be more than a predefined parameter β. If the difference does not satisfy this condition, it indicates a density variation between points and the points must be in different clusters. However, this situation occurs only in the boundary region of two different clusters, and requires repetition of this process for border points located in the border region of two different dense area.

Algorithmic Analysis: If a spatial index tree is used, the runtime complexity of EnDBSCAN will be $O(n \log n)$ like DBSCAN. If there are many clusters in a dataset such as those in hospitality datasets, processing runtime complexity of repetitive border-points need to be taken into consideration. However, if there are only a few number of clusters within a dataset, the runtime complexity of process repetition for border-points can be neglected.

D. A Variant of DBSCAN Algorithm to Find Embedded and Nested Adjacent Cluster

A variant of DBSCAN algorithm has been proposed in [17] to counter the limitation of previously presented density-based algorithms. To estimate the value of neighborhood radius r_μ as an input parameter, it uses the concept of k-distance plot and first-order derivative instead of selecting them by datasets observation. This approach allows the user to input the value of $MinPts$. To expand the cluster, firstly an arbitrary point has to be checked to verify the possibility of being a core-point. If the selected point is a core-point, only then the expansion process of clustering can be performed. Furthermore, this approach introduces a new term named neighborhood-difference. The term neighborhood-difference is defined as the difference between the numbers of neighborhood points of those two points. For example, one point belongs to a cluster as a core-point and another point is in the former's neighborhood with respect $MinPts$ and neighborhood radius r_μ, to determine those points are in the same cluster or not, the value of neighborhood-difference of those points must be within the range of tolerance factor α. The tolerance factor α is a value given as an input parameter by the user. If the neighborhood difference of that two points is greater than the tolerance factor α, those points might not be in the same cluster.

Instead of expanding cluster through neighborhood expansion like DBSCAN, this approach expands through core-neighborhood μ_γ of a cluster by satisfying the tolerance factor α issue as discussed earlier. Then a sorted k-distance graph is formulated in a plot to get the effective value of neighborhood radius μ_r. In this plot, a total number of points in the datasets take the independent (X) axis, and corresponding distances from each point to its kth-nearest neighbor take the dependent (Y) axis. After sorting and completing the k-distance vector and the first-order derivation respectively, we can get the effective value of neighborhood radius μ_r. If we see huge change of slope or sudden variation in the sorted k-distance graph, we can detect separation of cluster points from the noise points. Thus, we can also identify noise points by analyzing the threshold point from sorted k-distance graph. While sorting, if more than one data-points have equal k^{th} nearest distance, it is also possible that a neighborhood can contain more than (k+1) data-points.

Algorithmic Analysis: The manuscript where this approach has been presented does not clearly mention runtime complexity of this algorithm. Because the algorithm requires to implement a k-distance graph, the runtime complexity of this process will be $O(n)$. The graph-vector needs to be sorted, and the complexity of this process will be at least $O(n \log n)$ if an efficient sorting algorithm has been applied. Furthermore, optimal runtime complexity considering the expansion of cluster will be $O(n \log n)$ if spatial index tree used otherwise it will be $O(n^2)$. So the total optimal complexity of this approach is $O(n) + O(n \log n) + O(n \log n) = O(n \log n)$.

E. Effective Density-based Approach to Detect Complex Data Clusters

Nagaraju et al. [18] have proposed a density-based approach to identify clusters of varying densities and nested adjacent clusters. This approach recognizes that variation in the neighborhood data-point is useful to identify cluster rather than cluster density variation. To address their analysis this approach defined a new term named tolerance factor δ which is an input value given by the user. According to this approach, difference in the number of core-neighbors of a specific core-point and the number of core-neighbors of that core-point's core-neighbors might be less or equal to tolerance factor δ to remain in same class. If the difference is more than the tolerance factor, this algorithm may detect it as noise point or object.

Algorithmic Analysis: Although this approach is presented to minimize the dependency of global density parameter for clustering, this algorithm also requires efficient parameter setting such as the neighborhood radius r_μ and tolerance factor δ. This algorithm also requires continuous adjustment of tolerance factor δ to identify cluster's border-points properly. The significant problem with this approach is that it may identify many insignificant clusters. As this algorithm calculates neighborhood-difference and no predefined number of $MinPts$ is mentioned that may consist in a given neigh-

borhood μ, it might misleadingly identify too many clusters in the datasets. Whenever it finds the difference of neighborhood points, it may identify a new cluster.

As this approach has not specifically mentioned any use of spatial index tree, the runtime complexity of this algorithm will be $O(n^2)$.

IV. Comparison of Existing Clustering Algorithms Performance

Data related to human behavior and e-commerce is varied and complex. Thus providing recommendations by segmenting complex datasets such as hospitality industry datasets, requires efficient clustering algorithms which can identify varied density clusters and nested embedded cluster. Moreover, clustering algorithms should not be time-consuming if they are to be used for automated recommendation systems. The automated recommendation system is a kind of system which can generate a recommendation for the customer dynamically. Consequently, customer interaction with the system is also analyzed in real-time by the system to provide further effective recommendations. Therefore, runtime complexity is another criterion for measuring the performance of clustering algorithms.

As hospitality industry datasets are not coherent like medical imaging, animal genetic information datasets, and not predictable like e-commerce business market datasets, cluster analysis of these kinds of datasets is different. Scenarios such as varying density, nested adjacency, and nested embedded features of the cluster are very common in this industrys datasets. To address these scenarios properly in our study, we have used synthetic data. This approach produces significant results of clustering that help to evaluate the performances of algorithms mentioned in Section IV for hospitality big data.

In this section, we have first presented synthetic data relevant to the scenario mentioned above and then experimented with clustering algorithms over those datasets. We have performed algorithmic analysis of various density-based algorithms. Table I summarizes the runtime complexity of the implemented density-based algorithms.

Fig. 4 presents the comparative results of clustering for the algorithms mentioned above using different synthetic datasets. We have collected these synthetic datasets from [19], which are relevant in the hospitality industry context. We have further used R compiler to refine and generate new datasets. Fig. 4(a) represents a generic dataset without varying density cluster property.Fig. 4(e) represents a varying density and nested cluster feature dataset, and Fig. 4(i) represents a nested embedded cluster.

Firstly, Fig. 4(b), Fig. 4(c), Fig. 4(d) show the results of DBSCAN, OPTICS, and EnDBSCAN, respectively for the dataset shown in Fig. 4(a). Here, all the three density-based algorithms (i.e., DBSCAN, OPTICS, and EnDBSCAN) perform well to identify those clusters. Secondly, Fig. 4(f), Fig. 4(g), Fig. 4(h) show the results of DBSCAN, OPTICS, and EnDBSCAN, respectively, for the dataset shown in Fig. 4(e). For this dataset both OPTICS and EnDBSCAN perform well to identify those

TABLE I: Optimal runtime complexity of discussed density-based clustering algorithm

Algorithms	Optimal Runtime Complexity
DBSCAN [14]	$O(n \log n)$
OPTICS [15]	$O(n \log n)$
EnDBSCAN [16]	$O(n \log n)$
DBSCAN variant [17]	$O(n \log n)$
Complex Data Clustering [18]	$O(n^2)$

clusters whereas DBSCAN fails to identify some clusters because of varying density of points in the dataset. Finally, Fig. 4(j), Fig. 4(k), Fig. 4(l) show the results of DBSCAN, OPTICS, and EnDBSCAN, respectively, for the dataset shown in Fig. 4(i).

For this dataset, only EnDBSCAN performs well to identify the nested embedded clusters. On the other hand, DBSCAN and OPTICS both fail to identify embedded cluster. OPTICS indentifies many insignificant clusters instead of identifying these as a single cluster, and DBSCAN cannot identify that this dataset consist of different clusters. Clustering performance of another density-based algorithm, a variant of DBSCAN algorithm [17], may be better than the original DBSCAN algorithm because the variant detects neighborhood radius and threshold point of noise from first order derivative of the k-distance graph. However, comparable clustering performance can also be achieved by using the OPTICS algorithm [15] if an optimum neighborhood radius ropt is selected from the reachability plot. The density-based approach [18] mentioned in Section IV may identify many insignificant clusters instead of identifying the correct cluster. Furthermore, the approach [18] also has some dependency on its input parameter such as tolerance factor δ.

V. Conclusion

Although hospitality industry is one of the leading businesses in the world, very few research works have been conducted regarding the apt utilization of the huge volumes of available customer data. This paper provides insights into data clustering features of hospitality big data by analyzing existing density-based clustering algorithms. We have implemented popular density-based algorithms, such as DBSCAN, OPTICS, EnDBSCAN, and a few other variants of density-based algorithms, and have provided a comparative performance analysis of these algorithms. Results reveal that EnDBSCAN performs superior than DBSCAN and OPTICS in terms of identifying nested and embedded clusters. Similarly, OPTICS perform better than DBSCAN in identifying adjacent nested cluster for different datasets. However, all of the contemporary clustering algorithms have their limitations in identifying clusters from datasets because of their dependency on input parameters.

We can conclude that further research is needed to counter the limitations of existing clustering algorithms. Furthermore, novel clustering algorithms need to be developed for enabling automated recommendation systems for the hospitality industry to improve both the customers' experience and the revenue of the hospitality industry.

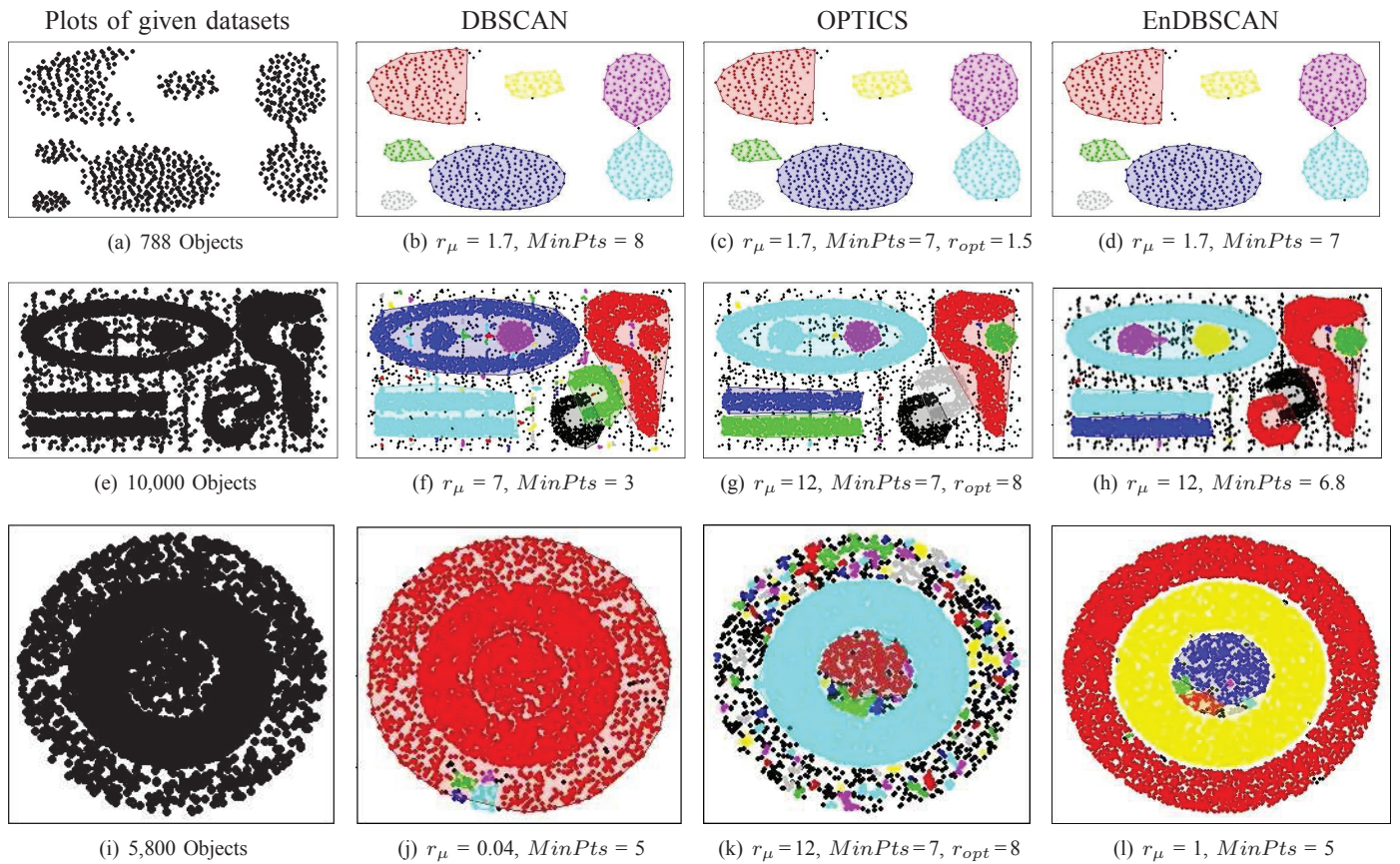| Plots of given datasets	DBSCAN	OPTICS	EnDBSCAN
(a) 788 Objects	(b) $r_\mu = 1.7$, $MinPts = 8$	(c) $r_\mu = 1.7$, $MinPts = 7$, $r_{opt} = 1.5$	(d) $r_\mu = 1.7$, $MinPts = 7$
(e) 10,000 Objects	(f) $r_\mu = 7$, $MinPts = 3$	(g) $r_\mu = 12$, $MinPts = 7$, $r_{opt} = 8$	(h) $r_\mu = 12$, $MinPts = 6.8$
(i) 5,800 Objects	(j) $r_\mu = 0.04$, $MinPts = 5$	(k) $r_\mu = 12$, $MinPts = 7$, $r_{opt} = 8$	(l) $r_\mu = 1$, $MinPts = 5$

Fig. 4: Performance result of different density-based clustering algorithms using synthetic datasets relevant to the datasets of hospitality Big Data.

REFERENCES

[1] M. Mowforth and I. Munt, *Tourism and Sustainability: Development, Globalisation and New Tourism in the Third World*. United Kingdom: Routledge, 2015.

[2] UNWTO. (2018, May) Tourism and poverty alleviation. [Online]. Available: http://step.unwto.org/content/tourism-and-poverty-alleviation-1

[3] Z. Xiang, Z. Schwartz, J. H. Gerdes, and M. Uysal, "What can big data and text analytics tell us about hotel guest experience and satisfaction?" *Elsevier International Journal of Hospitality Management*, vol. 44, pp. 120–130, January 2015.

[4] L. Zhou, S. Ye, P. L. Pearce, and M.-Y. Wu, "Refreshing hotel satisfaction studies by reconfiguring customer review data," *Elsevier International Journal of Hospitality Management*, vol. 38, pp. 1–10, April 2014.

[5] Z. Huang, D. Zeng, and H. Chen, "A comparison of collaborative-filtering recommendation algorithms for e-commerce," *IEEE Intelligent Systems*, vol. 22, no. 5, pp. 68–78, October 2007.

[6] Y. H. Cho, J. K. Kim, and S. H. Kim, "A personalized recommender system based on web usage mining and decision tree induction," *Elsevier Expert Systems with Applications*, vol. 23, no. 3, pp. 329–342, October 2002.

[7] M. Göksedef and c. Gündüz-Öğüdücü, "Combination of web page recommender systems," *Elsevier Expert Systems with Applications*, vol. 37, no. 4, pp. 2911–2922, April 2010.

[8] P.-H. Chou, P.-H. Li, K.-K. Chen, and M.-J. Wu, "Integrating web mining and neural network for personalized e-commerce automatic service," *Elsevier Expert Systems with Applications*, vol. 37, no. 4, pp. 2898–2910, April 2010.

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proc. of the 2nd ACM conference on Electronic commerce (EC)*, Minneapolis, Minnesota, October 2000, pp. 158–167.

[10] Y. Wind and D. R. Bell, "Market segmentation," in *The Marketing Book*, M. Baker and S. Hart, Eds. Taylor and Francis Group, 2008, pp. 222–244.

[11] G. Prayag, M. Disegna, S. A. Cohen, and H. Yan, "Segmenting markets by bagged clustering: Young chinese travelers to western europe," *Journal of Travel Research*, vol. 54, no. 2, pp. 234–250, December 2013.

[12] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, January 2003.

[13] G. Punj and D. W. Steward, "Cluster analysis in marketing research: Review and suggestions for application," *Journal of Marketing Research*, vol. 20, pp. 134–148, January 1983.

[14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, Portland, Oregon, August 1996, pp. 226–231.

[15] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," in *Proc. of the ACM SIGMOD International Conference on Management of Data*, Philadelphia, Pennsylvania, May-June 1999, pp. 49–60.

[16] S. Roy and D. K. Bhattacharyya, "An approach to find embedded clusters using density based techniques," in *Proc. of the Second International Conference on Distributed Computing and Internet Technology (ICD-CIT)*, Bhubaneswar, India, December 2005, pp. 523–535.

[17] S. Nagaraju, M. Kashyap, and M. Bhattacharya, "A variant of dbscan algorithm to find embedded and nested adjacent clusters," in *Proc. of the International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, February 2016.

[18] ——, "An effective density based approach to detect complex data clusters using notion of neighborhood difference," *Springer International Journal of Automation and Computing*, vol. 14, no. 1, pp. 57–67, February 2017.

[19] T. Barton and T. Brůna. (2018, May) clustering-benchmark. [Online]. Available: https://github.com/deric/clustering-benchmark