Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 125 (2018) 770–779

Procedia Computer Science

www.elsevier.com/locate/procedia

6th International Conference on Smart Computing and Communications, ICSCC 2017, 7-8 December 2017, Kurukshetra, India

Empirical Analysis of Data Clustering Algorithms

Pranav Nerurkar[a], Archana Shirke[b], Madhav Chandane[c], Sunil Bhirud[d]

[a]*Dept. of Computer Engineering & IT, VJTI, Mumbai - 400019, India*
[b]*Dept. of Computer Engineering & IT, VJTI, Mumbai - 400019, India*
[c]*Dept. of Computer Engineering & IT, VJTI, Mumbai - 400019, India*
[d]*Dept. of Computer Engineering & IT, VJTI, Mumbai - 400019, India*

Abstract

Clustering is performed to get insights into the data whose volume makes it problematic for analysis by humans. Due to this, clustering algorithms have emerged as meta learning tools for performing exploratory data analysis. A Cluster is defined as a set of objects which have a higher degree of similarity to each other compared to objects not in the same set. However there is ambiguity regarding a suitable similarity metric for clustering. Multiple measures have been proposed related to quantifying similarity such as euclidean distance, density in data space etc. making clustering a multi-objective optimization problem. In this paper, different clustering approaches are studied from the theoretical perspective to understand their relevance in context of massive data-sets and empirically these have been tested on artificial benchmarks to highlight their strengths and weaknesses.

Peer-review under responsibility of the scientific committee of the 6th International Conference on Smart Computing and Communications

Keywords: Clustering algorithms; Community structure; Unsupervised learning

1. Introduction

As the Digital transformation of the society gathers pace, there is an increase in proliferation of technologies that simplify the process of recording data efficiently. Low cost sensors, RF-IDs , Internet enabled Point of Sales terminals are an example of such data capturing devices that have invaded our lives. The easy availability of such devices and the resultant simplification of operations due to them has generated repositories of data that previously didn't exist. Today, there exist many areas where voluminous amount of data gets generated every second and is processed and stored such fields are social networks, sensor networks, cloud storages etc. This has boosted the fields of machine

* Corresponding author. Tel.: +91-961-999-7797.
E-mail address: pranavn91@gmail.com

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 6th International Conference on Smart Computing and Communications
10.1016/j.procs.2017.12.099

learning, pattern recognition, statistical data analysis and in general data science.

Even though such a volume provides huge opportunities to academia and industry it also represents problems for efficient analysis and retrieval [1]. To mitigate the exponential time and space needed for such operations data is compacted into meaningful summaries i.e. Exploratory Data Analysis [E.D.A.] which shall eliminate the need for storing data in unsupervised learning literature such summaries are equivalent to ”clusters”. E.D.A. helps in visualization and promotes better understanding of the data. It utilizes methods that are at the intersection of machine learning, pattern recognition and information retrieval. Cluster analysis is the main task performed in it.

A Cluster in a data is defined objectively using dissimilarity measures such as edit distance, density in a euclidean or non euclidean data space, distance calculated using Minkowski measures, proximity measures or probability distributions. All measures concur that a threshold value should be set for grouping of objects in a cluster and objects which exceed such a threshold are dissimilar and should be separated from the cluster. Clustering gives a better representation of the data since all objects within a cluster have less variability in their attributes and they can be summarized efficiently. Clustering has found applications in other fields like estimating the missing values in data or identifying outliers in data.

Clustering is thus a meta learning approach for getting insights into data and in diverse domains such as Market Research, E-Commerce, Social Network Analysis and Aggregation of Search Results among-st others. Multiple algorithms exist for organizing data into clusters however there is no universal solution to all problems. No consensus exists on the ”best” algorithm as each is designed with certain assumptions and has its own biases. These algorithms can be grouped into methodologies such as Partitioning based, hierarchical , density based, grid based, message passing based, neural network based, probabilistic and generative model based. However in terms of complexity it is a NP-hard grouping problem and so existing algorithms rely on approximation techniques or heuristics to reduce the search space in order to find the optimal solution. There is no universally agreed objective criteria for correctness or clustering validity and each of these algorithms has its own drawbacks and successes in solving the challenging problem of unsupervised clustering [3] [4] .

Motivated by these reasons, in this paper a review of the state of the art clustering algorithms is made to highlight their main strengths and weaknesses. Section II covers the theoretical aspects of these algorithms, Section III contains the Experiments performed using these algorithms and Section IV has the conclusion on the results.

2. Types of Clustering Algorithms

Various clustering algorithms are found in literature [1][3][4] and are broadly categorized into categories on the basis of an algorithm designer’s perspective with emphasis on the underlying clustering criteria:

2.1. Partition based clustering algorithms [5]

The general principle in these algorithms is that a cluster should contain atleast one object and that each object must belong to exactly one group i.e. hard clustering. The Number of clusters k is pre-specified by the user making this a semi supervised algorithm although many strategies have been suggested to estimate the ideal number of clusters like the empirical method where $k = \sqrt{n}$ where $n = \mid N \mid$ points and Elbow method where the k is fixed as the turning point on the graph of k v/s Avg. distance to centroid. The objective function to be minimized in these k-partitioning algorithms is SSE i.e. Sum of Squared Distance.

$$SSE = \sum_{k=1}^{k} \sum_{x_i \epsilon c_k} ||x_i - c_k||^2 \tag{1}$$

where c_k = centroid of the cluster,

Popular k-partitioning algorithms are K-Means which represents a centroid as the arithmetic mean of the objects in the cluster [6]. The algorithm was the most popular in this category even though it had drawbacks as it could not find non convex shaped clusters or handle non numerical attributes in higher dimensions. The time complexity was O(kN) making it suitable for large datasets, however the algorithm could theoretically take infinite iterations to converge and its mean based appraoch was sensitive to noisy data or data with outliers. Algorithms depending on Euclidean Distance measures suffer from the 'Curse of Dimensionality' due to distances being inflated in higher dimensions. K-Means++ by D. Arthur et. al 2007 [25] provides an improvement over K-Means by initializing the seed cluster centroids at maximum distance from each other. This technique has provided better results compared to random initializations with multiple repeats. Mini Batch K-Means by D. Sculley et. al [26] uses randomly sampled subsets of original data in each iteration of clustering. The approach improves computing time at the cost of slight reduction in accuracy compared to the Original K-Means algorithm.

To overcome the susceptibility to noise in mean based approaches, K-Mediods algorithm represented clusters by objects located near the centroids. Partitioning Around Mediods (PAM) [7] is the most popular approach for mediods based partitioning however it has a computational time complexity of O($k(n-k)^2$) and hence wouldn't scale well to large data-sets. Modified version of PAM such as CLARA (PAM with sampling) [8] and CLARANS [9] were proposed. CLARA has a computation time complexity of O($ks^2 + k(n-k)$) and CLARANS has O(n^2) and so both methods wouldn't be applicable to large sets of data.

Kernel K-Means [10] involves converting the points from euclidean space to high dimension kernel space. The kernel choices can be based on Mercer's criteria. R.B.F. or Gaussian is the common choice considered. The advantage of Kernel K-Means over K-Means is in finding non convex clusters albeit at the cost of computational time. BFR algorithm [11] is implemented for detecting clusters in large data-sets in a single pass over the data. The algorithm makes a strong assumption that clusters have objects that are normally distributed and due to this it can't find clusters at tilted angles to the axes or clusters of random shapes.

2.2. Fuzzy clustering

Fuzzy clustering algorithms assign a set of membership coefficients to each element which correspond to a "belongingness" or degree of membership to a cluster i.e. soft clustering. Fuzzy C-Means algorithm by Dunn in 1973 [17] and modified by Bezdek in 1981 [18] minimizes the objective function in Eqn. 4 for this purpose, Eqn. 5 defines the degree of belongingness u_{ij}^m and Eqn. 6 defines the centroid C_j of a cluster.

$$\sum_{j=1}^{k} \sum_{x_i \in C_j} u_{ij}^m (x_i - u_j)^2 \tag{2}$$

Where,

- u_{ij} is the degree to which an observation x_i belongs to a cluster C_j
- $_j$ is the center of the C_j
- m is the real number ($1 \leq m \leq \infty$) that defines the level of cluster fuzziness.

$$u_{ij}^m = \frac{1}{\sum_{l=1}^{k} (\frac{|x_i - c_j|}{|x_i - c_k|})^{\frac{2}{m-1}}} \tag{3}$$

$$C_j = \frac{\sum_{x \in C_j} u_{ij}^m x}{\sum_{x \in C_j} u_{ij}^m} \tag{4}$$

The algorithm minimizes intra -cluster variance but can converge to a local optimal solution. It depends on initialization of the seeds and different initializations may lead to different results. The number of clusters k have to be specified in advanced which is another drawback.

2.3. Model Based Clustering Algorithms [19]

The traditional clustering algorithms hierarchical and partition based clustering rely on heuristics whereas Model based algorithms assume that the data has been generated from a mixture of multiple probability distributions (Gaussian or multinomial) whose parameters mean, covariance matrix are to be estimated using the Expectation Maximization algorithm. The Bayesian information criteria or the Akaike information criteria can be used for selection of optimal number of clusters. The key drawback of this algorithms is that similar to k-means it also can converge to local optimal solution depending on the initial assignment of the k seeds. The objective function is not convex for these methods. Also, the optimization criteria can theoretically take infinite iterations to converge and a suitable threshold value has to be decided in advance. If the probabilities of the objects don't alter above this threshold then the algorithm can be stopped. Eqn. 6 is the prior probability that denotes the percentage of instances that came from source c. Eqn. 7 gives the mean i.e. expected value of attribute j from source c. Eqn. 8 gives the covariance matrix denoting the covariance of attributes j, k in source c.

$$P(c) = \frac{1}{n}\sum_{i=1}^{n} P(c|\vec{x_i}) \tag{5}$$

$$\mu_{c,j} = \sum_{i=1}^{n}\Big(\frac{P(c|\vec{x_i})}{nP(c)}\Big)x_{i,j} \tag{6}$$

$$\sum_{c}{}_{j,k} = \sum_{i=1}^{n}\Big(\frac{P(c|\vec{x_i})}{nP(c)}\Big)(x_{i,j} - \mu_{c,j})(x_{i,k} - \mu_{c,k}) \tag{7}$$

$$P(c|\vec{x_i}) = \frac{P(\vec{x_i}|c)P(c))}{\sum_{i=1}^{k} P(\vec{x_i}|c)P(c)} \tag{8}$$

$$P(\vec{x_i}|c) = \frac{1}{\sqrt{2\pi \sum_c}} \exp\Big(-\frac{1}{2}(\vec{x_i} - \vec{\mu_c})^T \sum_{c}^{-1}(\vec{x_i} - \vec{\mu_c})\Big) \tag{9}$$

2.4. Density based clustering algorithms

Cluster is defined as a connected dense component that can grow in any direction till the density continues to be above a threshold. This leads to automatic avoidance of outliers and detection of well separated clusters of arbitrary shapes. Popular methods based on this approach are DBSCAN by Kriegel et. al [23], OPTICS [24]. DBSCAN can find non linearly separable clusters and doesn't need the initial value of clusters to proceed. It uses euclidean distance measures to calculate distance between points in space and so is sensitive to curse of dimensionality. The parameters needed by DBSCAN are ϵ which defines the radius of neighborhood around a point and minimum neighbors $MinPts$ of a point in its ϵ - neighborhood.

In DBSCAN, pairwise distance is calculated between x_i and other points. For each point in the ϵ - neighborhood of x_i if the $N_{pts} \leq MinPts$ then mark it as *core* point. Then for each *core* point create a new cluster or assign it to a cluster if it is not assigned already. Find recursively all its density connected points and assign them to the same cluster as the core point. Iterate through the remaining un-visited points in the data-set. At the end of all iterations, the unassigned points are outliers.

OPTICS is an extension of DBSCAN to address the drawback of detecting clusters in varying densities. It accepts the parameters of DBSCAN ϵ and $MinPts$ in neighborhood $N_\epsilon(P)$. It additionally defines a new measure for every point known as $Core{-}Distance_{\epsilon,Minpts}(P) = C$ as in Eqn. 12 and $Reachability{-}Distance_{\epsilon,Minpts}(o, p) = R$ as in Eqn 13.

$$C = \begin{cases} UNDEFINED; & if|N_{\epsilon(p)}| < MinPts \\ smallestdistancetoN_{\epsilon}(p); & otherwise \end{cases} \tag{10}$$

$$R = \begin{cases} UNDEFINED; & if|N_{\epsilon(p)}| < MinPts \\ max(C, dist(o, p)); & otherwise \end{cases} \tag{11}$$

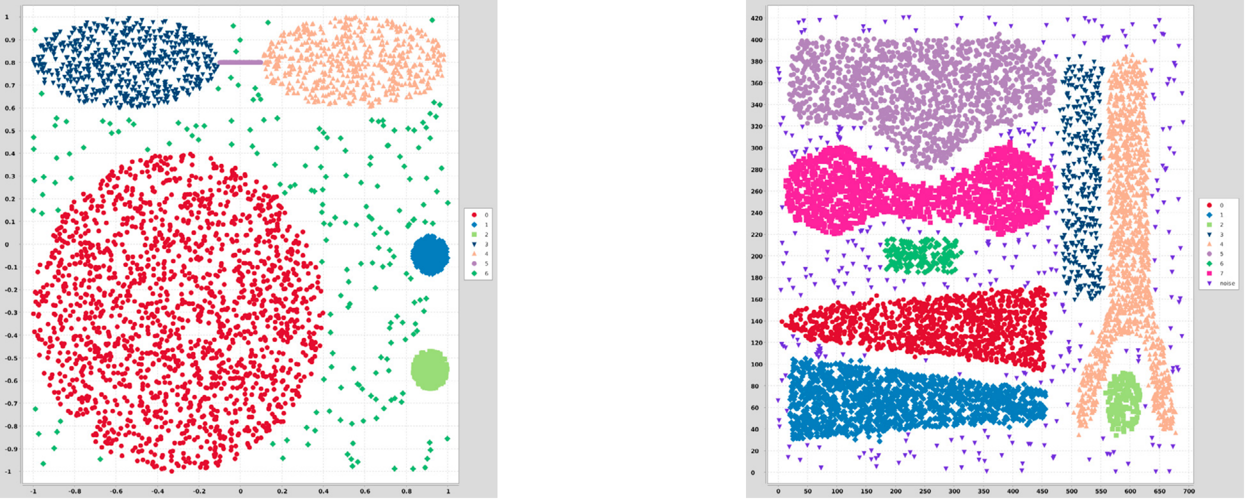
OPTICS produces a cluster ordering with respect to its density based clustering structure.

3. Experiments

Section II has examined clustering algorithms from a theoretical point of view and in this section their performance on a clustering benchmark data-set is provided for an empirical evaluation.

3.1. Data-set

The Data-sets selected are CURE-T2-4K and CLUTO-T8-8K which are publicly available artificially generated benchmarks by ClueMiner. The clusters can be identified by visualization but performance of clustering algorithms produces different results.



(a) CURE-T2-4K

(b) CLUTO-T8-8K

Fig. 1: Artificial Benchmark Data-sets for Clustering

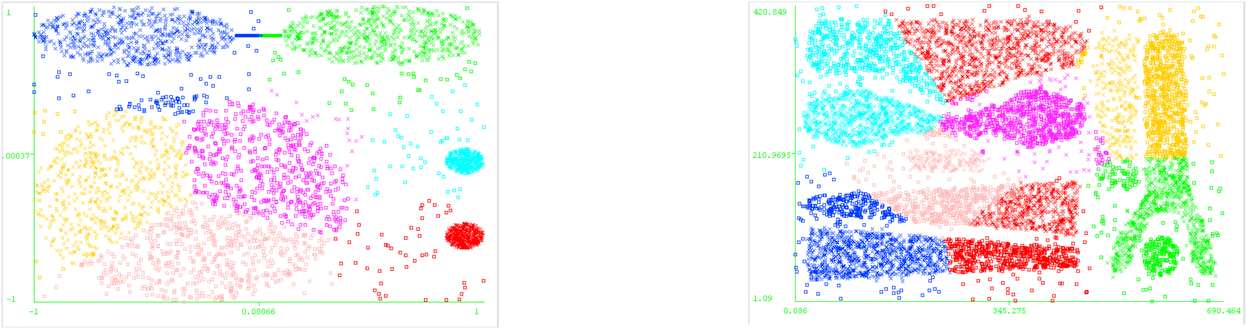
Table 1: Description of Data-set

Name	Instances	Attributes	Classes
CURE-T2-4K	4200	3	7
CLUTO-T8-8K	8000	3	9

3.2. Experimental Results

3.2.1. Partition based clustering Algorithm

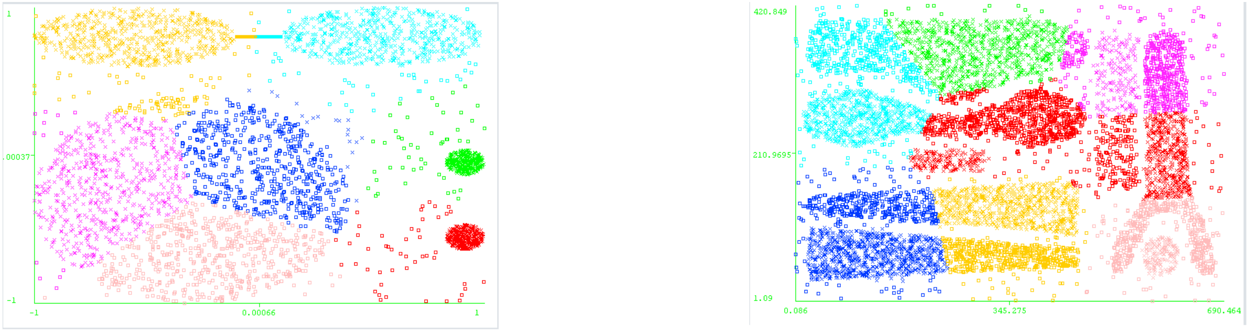
Results of K-Means in Fig 2, K-Means++ in Fig 3 and Kernel K-Means with RBF kernel in Fig 4 applied on the data-sets shows the detection of clusters symmetric along the axes is better than irregular shaped clusters.



(a) CURE-T2-4K

(b) CLUTO-T8-8K

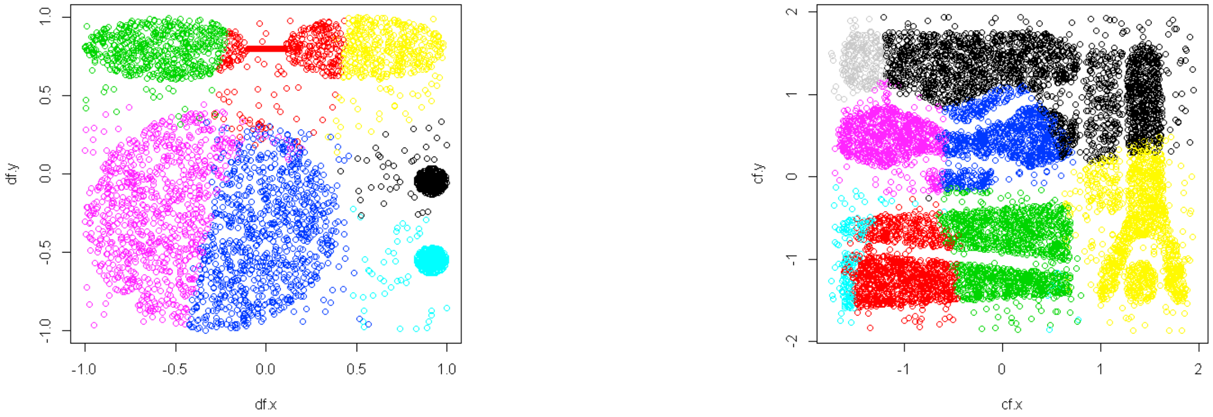
Fig. 2: Clustering based on K-Means



(a) CURE-T2-4K

(b) CLUTO-T8-8K

Fig. 3: Clustering based on K-Means++



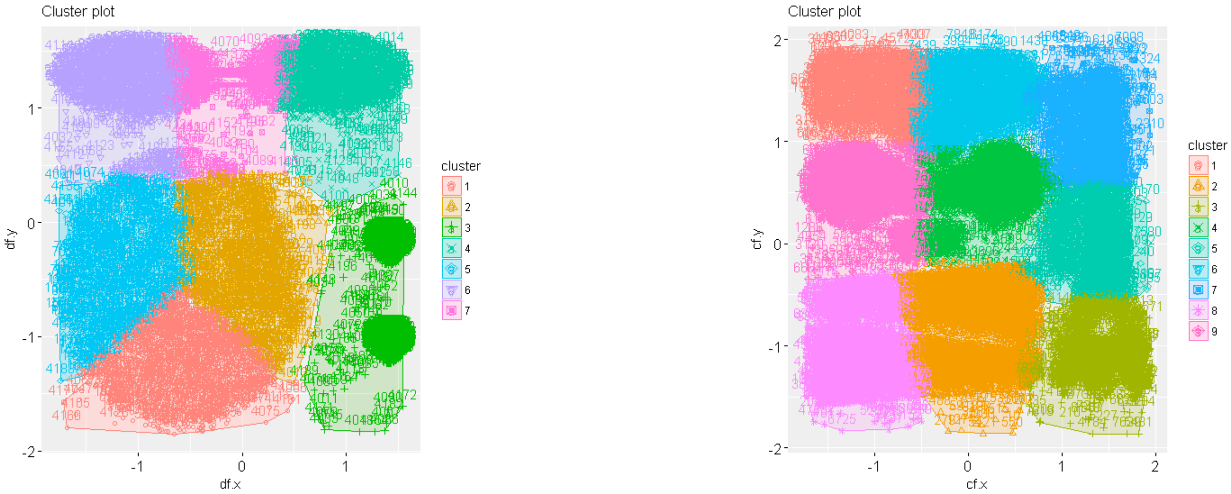
(a) CURE-T2-4K

(b) CLUTO-T8-8K

Fig. 4: Clustering based on Kernel K-Means

3.2.2. *Fuzzy clustering Algorithm*

The number of clusters have to be specified in advance and the clustering overlap is controlled by the fuzzifier parameter *m*. Random initialization was done with multiple repeats to avoid convergence to local minima. The approach was sensitive to noise and clusters were symmetric around the axes. Irregular shaped clusters were not detected.



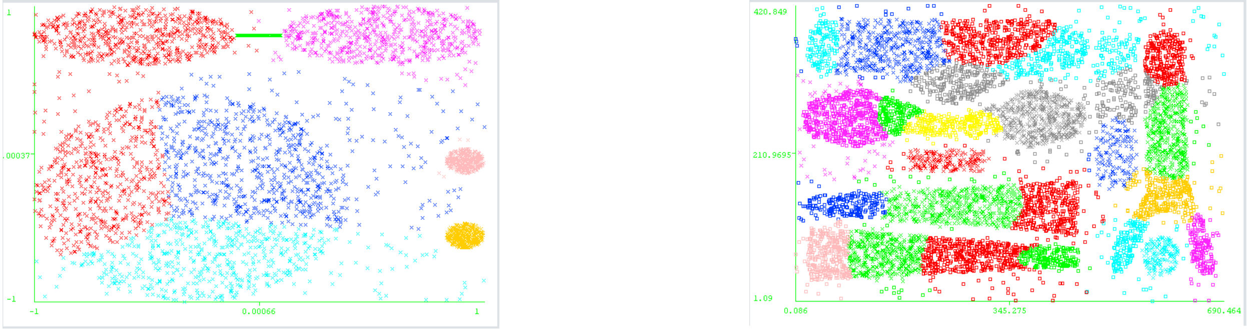
(a) CURE-T2-4K

(b) CLUTO-T8-8K

Fig. 5: Fuzzy C-Means Clustering

3.2.3. *Model based clustering Algorithm*

Performance of the Model based clustering algorithm was evaluated on the data-sets with the final model configuration selected by 10-cross fold cross validation. The initial seeds were randomly assigned 12 times to avoid local optima.



(a) CURE-T2-4K

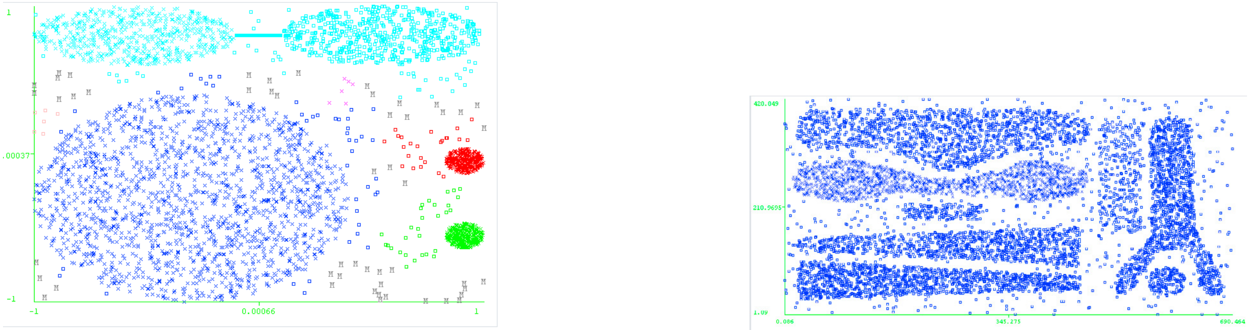
(b) CLUTO-T8-8K

Fig. 6: Clustering result of Model based clustering Algorithm

The mixture model based clustering has detected 8 clusters in CURE-T2-4K and 24 clusters in CLUTO-T8-8K as seen in Fig. 2 and incorrectly clustered instances were 50.23% and 60.66% respectively. A disadvantage of this approach was specification of the number of clusters initially.

3.2.4. *Density based Clustering Algorithms*

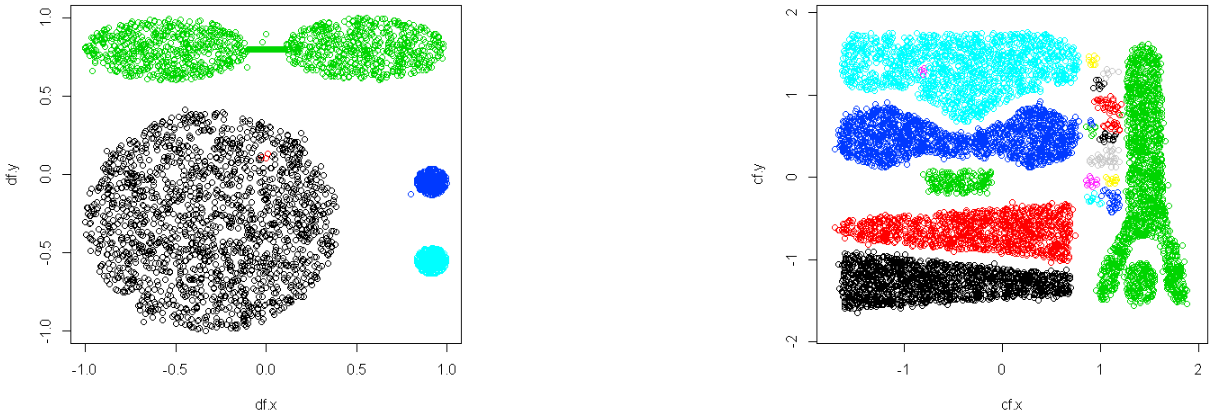
DBSCAN is parameter dependent and detected clusters of irregular shapes. It handled noise effectively. The key drawback is finding the right values of ε and $MinPts$ for a particular data-set. The clusters having irregular densities and not well separated were merged in both data-sets to give super clusters. OPTICS wasn't parameter sensitive and could identify clusters effectively.



(a) CURE-T2-4K

(b) CLUTO-T8-8K

Fig. 7: Clustering result of DBSCAN Algorithm



(a) CURE-T2-4K

(b) CLUTO-T8-8K

Fig. 8: Clustering result of OPTICS Algorithm

3.3. *Summary of Results*

Table 2: Evaluation of clustering algorithms

Name	Time Complexity	Parameters	Detect Asym Clusters
KMeans	O(nkd)	1	No
KMeans++	O(nkd)	1	No
Kernel KMeans	O(nkd)	1	No
Hierarchical clustering	O($n^2 logn$)	1	No
Fuzzy CMeans	O(n)	2	No
Model based	O(knp)	2	No
DBSCAN	O($nlogn$)	2	Yes
OPTICS	O($nlogn$)	2	Yes

4. Conclusion

Cluster detection poses a challenge to algorithms especially when underlying model for formation of community structure is not available. This is the case in most real world situations and hence there is ambiguity regarding defining the term ”cluster”. Ideally the approach to clustering should not require user interference, however all the current clustering algorithms require parameter tuning and this could result in models that over-fit the data and don’t generalize well. The algorithms could not identify clusters in the benchmark data-sets and had drawbacks like sensitivity to noise and outliers, high time and computational complexity and failure to detect clusters which were not well separated or of arbitrary shapes and densities.

References

[1] Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S. and Bouras, A., 2014. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. IEEE transactions on emerging topics in computing, 2(3), pp.267-279.

[2] Tukey, J.W., 1977. Exploratory data analysis, pp.68-197.

[3] Chapelle, O., Scholkopf, B. and Zien, A., 2009. Semi-supervised learning. IEEE Transactions on Neural Networks, 20(3), pp.542-542

[4] Estivill-Castro, V., 2002. Why so many clustering algorithms: a position paper. ACM SIGKDD explorations newsletter, 4(1), pp.65-75.

[5] Jain, A.K., 2010. Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), pp.651-666.

[6] MacQueen, J., 1967, June. Some methods for classification and analysis of multivariate observations. Fifth Berkeley symposium on mathematical statistics and probability, pp. 281-297.

[7] Kaufman, L. and Rousseeuw, P.J., 1990. Partitioning around medoids : Finding groups in data: an introduction to cluster analysis, pp.68-125.

[8] Jiawei, H., Pei, J. and Kamber, M., 2011. Data mining: concepts and techniques, pp.564-700.

[9] Ng, R.T. and Han, J., 2002. CLARANS: A method for clustering objects for spatial data mining. IEEE transactions on knowledge and data engineering, pp.1003-1016.

[10] Dhillon, I.S., Guan, Y. and Kulis, B., 2004. Kernel k-means: spectral clustering and normalized cuts. Tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 551-556.

[11] Bradley, P.S., Fayyad, U.M. and Reina, C., 1998. Scaling Clustering Algorithms to Large Databases. Symposium on Knowledge Discovery in Databases, pp. 9-15.

[12] Zhang, T., Ramakrishnan, R. and Livny, M., 1996. BIRCH: an efficient data clustering method for very large databases. ACM Sigmod Record, pp. 103-114.

[13] Guha, S., Rastogi, R. and Shim, K., 2000. ROCK: A robust clustering algorithm for categorical attributes. Information systems, pp.345-366.

[14] Guha, S., Rastogi, R. and Shim, K., 1998. CURE: an efficient clustering algorithm for large databases. ACM Sigmod Record, pp. 73-84.

[15] Karypis, G., Han, E.H. and Kumar, V., 1999. Chameleon: Hierarchical clustering using dynamic modeling. Computer, pp.68-75.

[16] Bezdek, J.C., Ehrlich, R. and Full, W., 1984. FCM: The fuzzy c-means clustering algorithm. Computers and Geosciences, pp.191-203.

[17] Dunn, J.C., 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Journal of Cybernetics, pp.32-57.

[18] Bezdek, J.C., Ehrlich, R. and Full, W., 1984. FCM: The fuzzy c-means clustering algorithm. Computers and Geosciences, pp.191-203.

[19] Fraley, C. and Raftery, A.E., 2002. Model-based clustering, discriminant analysis, and density estimation. Journal of the American statistical Association, pp.611-631.

[20] Ester, M., Kriegel, H.P., Sander, J. and Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. Second International Conference on Knowledge Discovery and Data Mining, pp.226-231.

[21] Ankerst, M., Breunig, M.M., Kriegel, H.P. and Sander, J., 1999. OPTICS: ordering points to identify the clustering structure. ACM Sigmod record, pp. 49-60.

[22] Arthur, D. and Vassilvitskii, S., 2007. k-means++: The advantages of careful seeding. Eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 1027-1035.

[23] Sculley, D., 2010. Web-scale k-means clustering. In Proceedings of the 19th international conference on World wide web, pp. 1177-1178.