IJICTR International Journal of Information & Communication Technology Research ITRC

Volume 9 - Number 2 - Spring 2017(27-34)

A Novel Density based Clustering Method using Nearest and Farthest Neighbor with PCA

Azadeh Faroughi
Computer Engineering and IT Department
Shiraz University of Technology
Shiraz, Iran
A.faroughi@sutech.ac.ir

Reza Javidan
Computer Engineering and IT Department
Shiraz University of Technology
Shiraz, Iran
javidan@sutech.ac.ir

Received: October 19, 2016 - Accepted: February 22, 2017

Abstract— **Common nearest-neighbor density estimators usually do not work well for high dimensional datasets. Moreover, they have high time complexity of $O(n^2)$ and require high memory usage especially when indexing is used. In order to overcome these limitations, we proposed a new method that calculates distances to nearest and farthest neighbor nodes to create dataset subgroups. Therefore computational time complexity becomes of $O(n \log n)$ and space complexity becomes constant. After subgroup formation, assembling technique is used to derive correct clusters. In order to overcome high dimensional datasets problem, Principal Component Analysis (PCA) in the clustering method is used, which preprocesses high-dimensional data. Many experiments on synthetic data sets are carried out to demonstrate the feasibility of the proposed method. Furthermore we compared this algorithm to the similar algorithm –DBSCAN- on real-world datasets and the results showed significantly higher accuracy of the proposed method.**

Keywords- nearest_neighbor density estimator; farthest neighbor; subgroups; principal component analysis(PCA);

I. INTRODUCTION

The main purpose of clustering [1] is finding the structure of unlabeled datasets. Data should be partitioned into clusters in a way that objects in the same cluster are more similar to each other than to those in other clusters. Unlike classification methods in which each data is assigned to precaution group, in clustering there is no prior information about the class membership among data and in fact clusters are extracted from data information. Data clustering can be used in some applications such as marketing, biology, analysis and classification of network traffic, image processing, time series forecasting, machine learning, pattern recognition and natural language processing [2]–[7] and it is potentially useful in other fields [8], [9].

The clustering has some challenges such as selecting suitable clustering algorithm that can handle huge number of dimensions and distributed data. Some of the density-based clustering methods are based on nearest neighbor density estimators. The time complexity of these methods is $O(n^2)$; because they need to find the nearest neighbor for every data in the dataset. Consequently, utilizing these methods are impractical when the dataset is large [10]–[12]. There are other methods that use *k*-nearest neighbor method to find the nearest neighbor in the dataset [13], [14]. Even these methods reduce the time complexity to $O(n \log n)$, but these indexing

algorithms have high memory requirement and this speedup only occurs in datasets with few dimensions [15].

In [16] we proposed an approach to density clustering based on finding nearest and farthest neighbors. This approach first creates some subgroups of dataset that each node becomes member of the nearest subgroup and then ensembles these subgroups to obtain the final clusters. Since the number of subgroups is smaller than the size of the dataset, the algorithm has an appropriate speed. However finding the clusters of high-dimensional data using this algorithm is a poor job. This may generate wrong number of clusters for real-world datasets. This is because many of the dimensions in high dimensional datasets are often irrelevant. These irrelevant dimensions can hide the clusters in noisy data which leads to confusion of clustering algorithm. In this paper on the basis of the former method, an improved method is proposed to overcome the previously mentioned drawbacks using Principal Component Analysis (PCA) [17]. In Some works like [18] PCA is used as a feature extraction mechanism to map the dataset to one with a lower feature space by removing less significant features.

The main contributions of this paper are:

1- PCA is used in proposed clustering estimator to reduce dimensions of dataset.
2- The proposed method is tested on synthetic datasets and its feasibility is demonstrated. In addition, to evaluate the performance of the proposed method, it is compared to DBSCAN algorithm on real datasets which are acquired from UCI repository [19]. The test results showed that the proposed clustering method is better than DBSCAN method and when dimensions of dataset are high, PCA-based method has a better performance than others.

The rest of this paper is organized as follows: In Section 2 some related works about clustering methods are reviewed. In Section 3, the proposed method is described in details. Section 4, the proposed algorithm based on PCA is explained. In Section 5, the experimental results on synthetic datasets and UCI datasets are presented. The final Section covers conclusion.

II. Related Works

Clustering is defined as unsupervised classification of data into groups or clusters. Various types of clustering algorithms have been proposed and developed in the literature (e.g., [20] and the references therein). Generally, clustering methods are divided into three main categories: partitioning approach, hierarchical approach, and density-based approach.

In *partitioning approach* various partitions are constructed and then evaluated by some criterion like minimum sum of square errors. Typical methods of this approach are *k*-means [21] and CLARANS [22]. These methods are simple and they converge to local optimum very fast. However the limitation of these methods is that the number of clusters must be predefined and they don't work well for clusters with different sizes and shapes.

In *Hierarchical approach*, a hierarchical decomposition of datasets is created using some criterion. CURE [23] and CHAMELEON [24] are examples of this approach. These methods are suitable for clusters with different size and shape, but their complexity is high and their convergence is slow.

Density-based algorithms such as DBSCAN [25], SSN [26], OPTICS [27] and MSC [28], [29] are based on connectivity and densities that exist among datasets. In this approach, clusters are zones with high density of data, which are separated by regions of lower density. In these methods clusters can be arbitrarily shaped and the number of clusters is automatically determined simultaneously during the operation of clustering. DBSCAN requires two parameters: $\varepsilon(eps)$ and the minimum number of points required to form a dense region $(\min Pts)$ .It starts with an arbitrary starting point that has not been visited. This point's $\varepsilon - neighborhood$ is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise.

Some density based clustering algorithms like *k*-nearest-neighbor density estimator and DBSCAN determine a local neighborhood based on a global parameter, i.e., k or ε ; and the density is calculated based on these variables. In addition, these algorithms consider the entire dataset to find nearest neighbors, which leads to time complexity of $O(n^2)$ for n nodes. To overcome this execution complexity, some researches are focused on reducing this cost by employing different indexing methods while these algorithms need high memory. Some works like [30] focused on the discretization of data to improve accuracy.

Our method computes the density, based on finding the nearest and farthest neighbors of subset centers with novel features as follows. The number of objects in each subgroup and its volume are adaptive according to data distribution; unlike the methods that create subgroups based on a global parameters and their subgroup's size are fix. This method only needs a small subsample to find nearest neighbors and it searches farthest neighbors inside each subgroup which form smaller search space than the whole dataset.

III. The Proposed Clustering Method

This method includes two steps. In the first step, subgroups are created and then at the next step, subgroups are merged to form clusters based on novel proposed assembling technique.

A. Creating subgroups of dataset

In this step, a subsample M of size $m = 2\log n$ from dataset is selected. The process of subsample selection uses uniform distribution which causes the selected samples to be distributed across the entire dataset uniformly. Sample nodes are representative nodes of M subgroups. Thus the dataset is divided into M subgroups according to the procedure described below.

First of all, for each node in M the nearest sample node is identified. Euclidean distance is used to compute nodes distances (Eq. (1) and (2)).

$$d(s,k) = \|s,k\|_2 = (\sum_{i=1}^{n}(s_i - k_i)^2)^{1/2}, \quad (1)$$

$$\underset{i\in M, i\neq k}{j} = \arg\min_{k\in M}(d(i,k)), \quad (2)$$

Where, $d(i,k)$ is the distance between node i and node k and $\arg\min$ denotes minimum arguments, i.e. the nearest sample node to node i. Then data at the local region with center of each sample node and its radius are assigned to subgroups. At the beginning of the algorithm, radius of each subgroup is defined by Eq. (3):

$$\underset{i\in M}{r_i} = \min_{j\in M}(\frac{d(i,j)}{2}) \quad (3)$$

In Eq. (3) r_i is the radius of local region sub_i with the center of sample node. When some nodes are joined to subgroups, new representative node for each subgroup is identified which is the farthest node from previous representative node (here that is the sample node). Each of these new representative nodes is called *First Farthest Neighbor (FFN)* and they form a new set called FFN- set. At the next iteration for each object in FFN-set nearest neighbor is identified, according to Eq. (1) and (2). At this point, new radius for each subgroup is calculated based on variance criterion (according to Eq. (4)) in a way that subgroups with higher variance and dispersion have larger radius.

$$\underset{i\in M}{r_i} = \frac{(Min_{j\in M} d(i,j) \times \text{var\_}sub_i)}{(\text{var\_}sub_i + \text{var\_}sub_j)}, \quad (4)$$

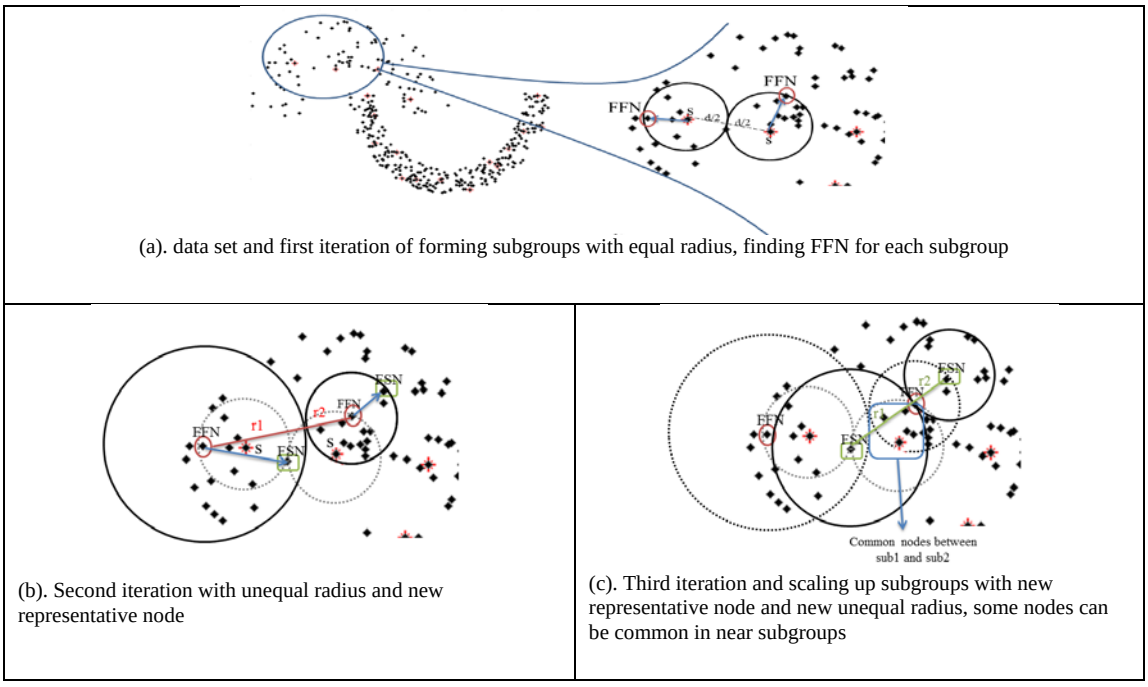
Where, $d(i,j)$ is the distance between FFN of subgroup i and it's nearest FFN in FFN-set. Variables of $\text{var\_}sub_i$ and $\text{var\_}sub_j$ are nodes distribution of subgroup i and its nearest subgroup (j) respectively; j is obtained based on Eq. (2). In this step variance of each subgroup is calculated according to minimum distance between members of each subgroup to its sample node (s) or FFN (Eq. (5)).

$$\underset{i\in M}{D_i} = \{dist_n = \min(d(n,c_i), d(n,FFN)) \forall n \in sub_i\}, \quad (5)$$

$$\underset{i\in M}{\text{var\_}sub_i} = E[(dist_n - E(dist_n))^2], \quad (6)$$

Where D_i is the distance set of each subgroup's member in Eq. (5). Variance of D_i is computed as the expected value $E[.]$ of the squared deviation from the expected value of $dist_n$ in Eq. (6). To obtain variance of each subgroup, instead of computing the distance between all objects in each subgroup, only the distance between each member and the center or FFN is used.

So according to new calculated radiuses nodes, which are in the range of subgroups and are not labeled by any subgroup yet, are assigned to these local regions. Due to new radius calculation, some nodes which were labeled with one subgroup might be on the range of other subgroups. So these nodes are considered as common nodes between two or more subgroups.



(a). data set and first iteration of forming subgroups with equal radius, finding FFN for each subgroup

(b). Second iteration with unequal radius and new representative node

(c). Third iteration and scaling up subgroups with new representative node and new unequal radius, some nodes can be common in near subgroups

Fig. 1. Example of process of subgroups forming

In the next step, in each subgroup the farthest node to *FFN* is selected as *Second Farthest Neighbor (SFN)*. This is performed to extend each subgroup in different directions. As these SFN nodes are the new representative nodes for the subgroups, like pre-representative nodes, nearest SFN node is identified according to Eq. (1) and (2) for each node in *SFN* set. Then new ranges of subgroups are computed according to Eq. (4). In this step, to compute the variance of each subgroup, the minimum distance between each node and 3 points of its subgroup including c, FFN and SFN is considered.

$$j = \arg\min_{k=c,FFN,FSN}(d(i,k)), i = 1,...,|sub_i| \quad (7)$$

Where $|Sub_i|$ denotes the cardinality of each subgroup in Eq. 7. Then variance and new r_i are calculated based on Eq. (4) and (6). This part of the algorithm is repeated until all of the nodes are assigned to one subgroup.

Algorithm 1. Forming subgroups
Input: *N*-Input Data
Output:{sub_i |*i*=1,…,*m*}

1. *m*=2log(*N*)
2. **for** *i*=1 to *m*
3. $C \leftarrow Sample(N)$
4. **end**
5. **for** *i*=1 to *m*
6. $j \leftarrow$ find nearest neighbor from other sample
7. $radius \leftarrow \frac{1}{2} \times \|c_i, c_j\|_2$
8. **end**
9. *assign*=0;
10. **for** *i*=1 to *m*
11. **for** *j*=1 to *N*
12. **if** x is unassigned and $\|rep, x_j\|_2 \le radius_i$
13. $Sub_i \leftarrow x_j$
14. *assign*=1;
15. **end**
16. **if** x is assigned to sub_j and $\|rep, x_j\|_2 \le radius_i$, $j \ne i$
17. $Com(i, j) \leftarrow Com(i, j) + 1$
18. **end**
19. **end**
20. **end**
21. **for** $i = 1$ to m
22. find farthest neighbor for rep node
23. **end**
24. **for** $i = 1$ to m
25. $j \leftarrow$ Find nearest neighbor from other rep nodes
26. $radius \leftarrow \frac{\text{var}_i}{\text{var}_i + \text{var}\, j} \times \|rep_i, rep_j\|_2$
27. **end**
28. **if** *assign*==1 and there are unassigned nodes **Go** to step 9
29. **return** {sub_i |*i*=1,…,*m*}

At these steps FFN and SFN are frequently changed but centers are constant. An example of subgroups' creating steps is shown in Fig. 1(a), Fig. 1(b) and Fig. 1(c). Since the size of M is $O(\log(n))$ and in each iteration for each subgroup all of unassigned nodes are considered, the time complexity of the algorithm is at most $O(\log(n))$. The first step of algorithm is implemented in algorithm 1.

B. *Assembling technique*

In the proposed method, two measures of common nodes and density are intended to merge subgroups. Subgroups with common nodes can be considered as a single cluster. Due to different dispersion of clusters, there might be some subgroups that have common nodes but do not belong to the same cluster. To handle this problem, other measure is also used called density. Therefore, two subgroups will belong to the same cluster, if two measures of common nodes and densities are satisfied.

The density criterion refers to ratio of middle point density between two subgroups to minimum density of these subgroups. If this ratio is more than threshold parameter β , density criterion is satisfied. The threshold of β is selected according to the problem context and in most β has a value between 0.25 and 1. Density of each subgroup can be expressed as Eq. (8):

$$\rho_{\substack{i \\ i \in M}} = \frac{|Sub_i|}{Max(d(FFN_i, SFN_i), d(FFN_i, c_i), d(SFN_i, c_i))} \quad (8)$$

Where $|Sub_i|$ is the cardinality of subgroup i, and maximum mutual distance between three parameters of c, FFN and SFN is the diameter of subgroup i.The density of middle point is calculated according to Eq. (9)

$$\rho_{\substack{middle(i,j) \\ i,j \in M}} = \frac{|middle|}{d(c_i, c_j)}, \quad (9)$$

Where $d(c_i, c_{j)}$ is the distance between sample nodes of subgroups i and j , $|middle|$ is the number of the nodes that are at the region with the center of middle point and radius $r_{middle} = \frac{d(c_i, c_j)}{2}$.

When both of the two conditions are established, suitable subgroups are merged to form a cluster. This process is repeated until there are no subgroups for merging under these conditions.. Assembling technique is implemented in algorithm 2.

IV. The Poposed Clustering Method with PCA

Due to existence of irrelevant features in datasets, proposed algorithm may generate wrong number of clusters especially on real-world datasets. In this paper to improve our algorithm on dealing with high dimensions datasets we bring forward our method based on principal component analysis (PCA).

Algorithm 2. Assembling subgroups
Input: set of subgroups, parameter β
Output: number of clusters, nodes tag

1. **for** i=1 to m
2. $\rho_i \leftarrow$ density of sub_i
3. **end**
4. k=0
5. **for** i=1 to m
6. **for** j=1 to m
7. **if** sub_i and sub_j aren't member of any cluster
8. $k=k+1;$
9. **end**
10. $\rho_{middle} \leftarrow$ density of middle point of sub_i, sub_j
11. **if** i and j have common nodes and $\frac{\rho_{middle}}{\min(\rho_i, \rho_j)} \geq \beta$
12. $cluster_k \leftarrow$ join sub_i, sub_j
13. **end**
14. **end**
15. **end**
16. **return** k and datasets with their tags

Principal components analysis (PCA) [17] is a widely used dimensionality reduction algorithm that can be used to significantly speed up unsupervised feature learning algorithms. The basic idea of PCA is to project the original data onto a lower-dimensional subspace, which highlights the principal directions of data's. The following steps describe this algorithm procedure.

1- Computing the average of dataset according to Eq. (10).

$$\Psi = \frac{1}{n}\sum_{i=1}^{N} X_i, \tag{10}$$

where $X = [X_1, X_2, ..., X_n]$ are the set of observation in which each of observation has a row vector of length m, so the dataset is represented by a matrix $X_{n\times m}$.

2- Calculating the sample covariance matrix of dataset according to Eq. (11).

$$C = \frac{1}{n}\sum_{i=1}^{n}(X_i - \Psi)(X_i - \Psi)_T. \tag{11}$$

3- Calculating the eigenvalues-eigenvectors pairs of sample covariance matrix C, in which for a square matrix C of order n, the number λ is an eigenvalue if and only if there exists a non-zero vector V such Eq. (12) will be satisfied.

$$CV = \lambda V, \tag{12}$$

Where,

$$(\lambda, v) = ((\lambda_1, v_1), (\lambda_2, v_2), ..., (\lambda_m, v_m)). \tag{13}$$

4- Sorting these eigenvalues in decreasing order and choosing k eigenvectors having the largest eigenvalues. The selection of k eigenvectors can be determined by Eq. (14).

$$\frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{m}\lambda_i} >> \alpha, \tag{14}$$

where α is the ratio of variation in the subspace to the total variation in the original space. Therefore k eigenvectors are selected for data reduction.

5- Obtaining the new representation of the data by projecting it onto the k-dimensional subspace according to $Y = V^T X$. In this paper we pick the smallest value of k by considering α=0.99 as shown in Eq. (15).

$$\frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{m}\lambda i} >> 0.99 \tag{15}$$

Then we use the new datasets in our method to be clustering. In Algorithm 3 the PCA-based clustering method is implemented.

Algorithm 3. The Proposed Clustering Method based on PCA
Input: N-Input Data: $X \in R_{n\times m}$
Output: reduction dimension of X: $Y \in R_{n\times k}$

1. **for** i=1 to N
2. $\Psi \leftarrow mean(X_i)$
3. **end**
4. **for** i=1 to N
5. $C \leftarrow Co\,\mathrm{var}\,iance(X_i)$
6. **end**
7. **for** i=1 to M
8. $(EigVec, EigVal) \leftarrow Eig(C_i)$
9. **end**
10. $(EigVal) \leftarrow sort(EigVal, 'descent')$
11. **While** $\frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{m}\lambda_i} < 0.99$
12. Continue
13. **Else**
14. **for** j=k+1 to m
15. $EigVec(j) \leftarrow zeros(EigVec(j))$
16. **end**
17. **end**
18. **for** i=1 to N
19. $Y_i \leftarrow EigVec^T \times X_i$
20. **end**
21. $S \leftarrow \{SubgroupSets) = For\,\min gSubgroups(Y)$
22. $return(k, Y, Labels) \leftarrow AssemblingSubgroups(S, \beta$

V. Experimental results

In this section, the performance of the proposed method is tested on two types of experiments. These experiments include tests on synthetic datasets and tests on real-world datasets. The results of tests on synthetic datasets are used to show the feasibility of the proposed method and the results of tests on real world datasets are compared with DBSCAN algorithm to compare the accuracy.

A. Results of synthetic data sets

Since there are only two feature dimensions in synthetic datasets, the results are represented visually for this type of experiments. Because the performance of clustering algorithm and PCA based algorithm on two dimensional datasets are similar, we only test the performance of proposed method.

The proposed method is tested on 6 synthetic datasets. The first synthesized dataset includes 5 clusters with 200 nodes in each cluster and totally 1000 nodes. Data is generated randomly with normal distribution on vertical and horizontal axes in each cluster. The spaces between clusters are selected in such a way that clusters are easily separable. The second dataset is aggregation [31], which consists of seven distinct clusters which some of them are not obviously separable. Data is generated with a non-Gaussian distribution. The third dataset named spiral [32], forms 3 similar spiral shaped clusters that each cluster has 106 nodes. The forth dataset named Jain [33], consists of 2 clusters with different densities and the fifth one –Flame [34]- has 2 clusters with different sizes and shapes. R15 [35] is the sixth dataset which contains 15 clusters that are positioned as rings. Fig. 2 shows that the proposed method works perfect for all datasets with different shapes and different densities and it can find the correct number of clusters and suitable tag label for each node.

B. Real-world dataset

The real-world datasets which are used in this paper are acquired from the *UCI Machine Learning Repository* [19]. These datasets include Animals, Pendigits, Segment, Yeast, Iris and WDBC. The details of these datasets are given in Table I.

TABLE I. Real world datasets

Datasets	# cluster	# dimension	Size n
Iris	3	4	150
Yeast	10	8	1484
Pendigits	10	16	10992
Animals	4	17	200,000
Segment	19	19	2310
WDBC	2	30	569

The clustering results by DBSCAN and our proposed method and PCA-based proposed method are shown in Tables II and III. In these tables the results are assessed in terms of number of unassigned nodes which don't join any cluster, number of clusters that method can identify, and F-measure which is calculated based on assigned instances only. Some papers like [36] use some assessment metrics such as accuracy, true positive and false negative to compare results but here we use F-measure which is equal to 1 when all assigned instances are in the correct clusters, i.e. perfect clustering and is equal to 0 if all instances are assigned to wrong clusters.

As it is obvious in Table II and III, for all of methods their parameters are set to values in a way that they get better result and detect true clusters. The proposed method shows better result in identification of number of clusters than DBSCAN and PCA-based for iris dataset. Experiment results on Pendigits and Yeast datasets show all methods haven't found the suitable number of clusters but our method and PCA-based have shown higher accuracy than DBSCAN. Our method resulted in 769 unassigned nodes in Pendigits dataset while this count is 4563 for DBSCAN that means proposed method could extend better in some directions and covered more nodes. All of methods obtain correct clustering on the Animals dataset and have good accuracy. All of methods obtain correct clustering on the WDBC dataset, however PCA-based has a better accuracy than others

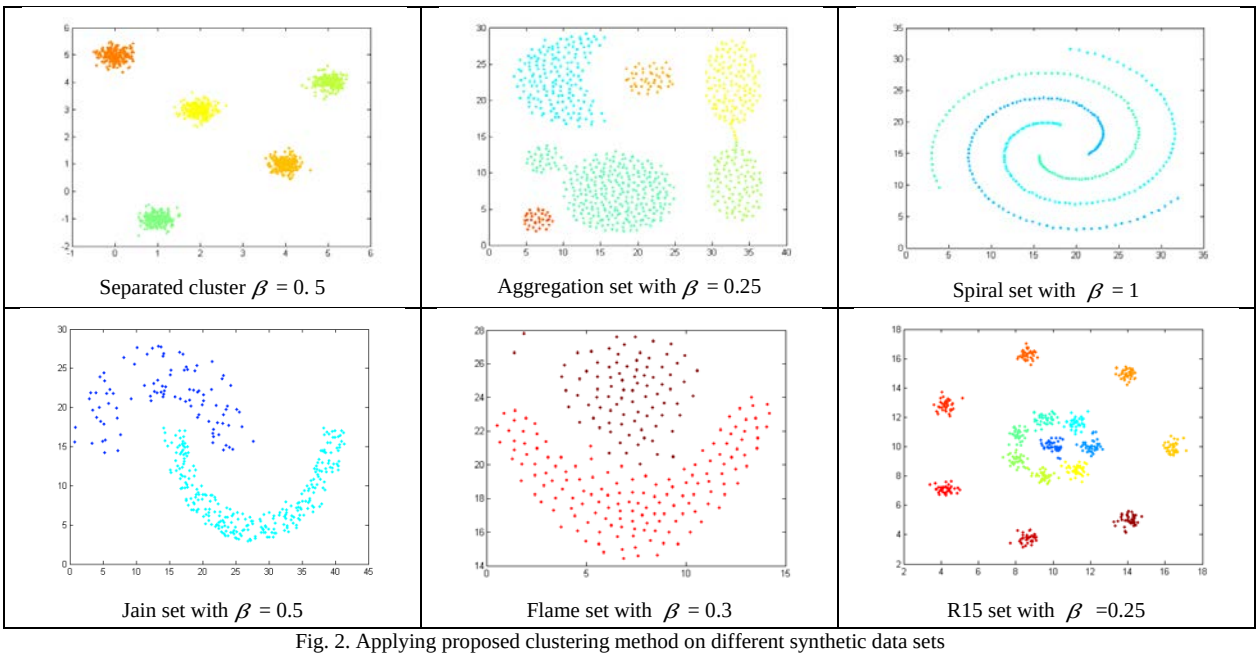


Fig. 2. Applying proposed clustering method on different synthetic data sets

Table II. Clustering results on the Iris, Yeast and Pendigits datasets

	Datasets	Iris-3	Yeast-10	Pendigits-10
DBSCAN	*Parameter*	$\varepsilon = 0.1$ MinPts =6	$\varepsilon = 0.07$ MinPts =6	$\varepsilon = 0.2$ MinPts =5
	# cluster	5	12	46
	# Unassigned	75	1097	4563
	F-measure	0.79	0.24	0.7
Our Method	*Parameter*	$\beta = 0.5$	$\beta = 0.45$	$\beta = 0.75$
	# cluster	3	12	12
	# Unassigned	3	400	769
	F-measure	0.91	0.3	0.78
Our PCA based Method	*Parameters*	$\beta = 0.45$	$\beta = 0.3$	$\beta = 0.55$
	# cluster	4	13	11
	# Unassigned	26	578	654
	F-measure	0.83	0.2	0.82

.

As shown in Table II and III, Our method outperforms others in low-dimensional datasets such as Iris and yeast, because it doesn't miss any information of features. However, our PCA-based method has a better performance compared to others in relatively high-dimensional data sets. PCA not only reduces the dimensionality of the data, but also maintains information as much as possible. When some datasets have relatively high dimensions, our method does poor job, so we need to employ PCA to reduce dimensions and remove irrelevant features. In consequence, PCA-based method outperforms others in high-dimensional datasets.

VI. Concolusion

In this paper a new clustering algorithm based on subsampling method with PCA was introduced. This algorithm first forms subgroups based on nearest and farthest neighbors and then aggregate the subgroups to obtain correct clusters.

TABLE III. Clustering results on the Animals, Segment and WDBC datasets

	Datasets	Animals-4	Segment-19	WDBC-2
DBSCAN	*Parameters*	$\varepsilon = 0.7$ MinPts =5	$\varepsilon = 0.1$ MinPts =6	$\varepsilon = 0.3$ MinPts =6
	# cluster	4	43	2
	# Unassigned	3342	1043	294
	F-measure	1	0.62	0.84
Our Method	*Parameter*	$\beta = 0.5$	$\beta = 0.25$	$\beta = 0.65$
	# cluster	4	36	2
	# Unassigned	3021	320	234
	F-measure	1	0.76	0.94
Our PCA based Method	*Parameters*	$\beta = 0.65$	$\beta = 0.5$	$\beta = 0.5$
	# cluster	4	28	2
	# Unassigned	2564	270	127
	F-measure	1	0.84	0.96

Computational complexity of this method is less than other similar methods because this method doesn't need to search nearest and farthest neighbors over the entire dataset. PCA is used to reduce the dimensions of high dimensional dataset to remove the irrelevant features of datasets. Experimental results revealed that proposed method is able to detect the correct number of clusters and assign nodes to correct cluster with high accuracy in synthesized and real-world datasets even with arbitrary shape. Moreover when the dimension of dataset is high, PCA-based method has a better performance than others.

References

[1] A. Nagpal, A. Jatain, and D. Gaur, "Review based on data clustering algorithms," in *2013 IEEE CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES*, 2013, pp. 298–303.

[2] Z. Nafar and A. Golshani, "Data Mining Methods for Protein-Protein Interactions," in *2006 Canadian Conference on Electrical and Computer Engineering*, 2006, pp. 991–994.

[3] W. Yu, G. Qiang, and L. Xiao-li, "A Kernel Aggregate Clustering Approach for Mixed Data Set and Its Application in Customer Segmentation," *2006 Int. Conf. Manag. Sci. Eng.*, pp. 121–124, 2006.

[4] Zicheng Liao, H. Hoppe, D. Forsyth, and Yizhou Yu, "A Subdivision-Based Representation for Vector Image Editing," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 11, pp. 1858–1867, Nov. 2012.

[5] P. K. Vemulapalli, V. Monga, and S. N. Brennan, "Robust extrema features for time-series data analysis," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 35, no. 6, pp. 1464–1479, 2013.

[6] Q. Song, J. Ni, and G. Wang, "A fast clustering-based feature subset selection algorithm for high-dimensional data," *Knowl. Data Eng. IEEE Trans.*, vol. 25, no. 1, pp. 1–14, 2013.

[7] Y. Chen, H. Sampathkumar, B. Luo, and X. Chen, "iLike: Bridging the Semantic Gap in Vertical Image Search by Integrating Text and Visual Features," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2257–2270, Oct. 2013.

[8] S. Yin and X. Zhu, "Intelligent Particle Filter and Its Application on Fault Detection of Nonlinear System," *IEEE Trans. Ind. Electron.*, pp. 1–1, 2015.

[9] S. Yin, X. Zhu, and O. Kaynak, "Improved PLS Focused on Key-Performance-Indicator-Related Fault Diagnosis," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1651–1658, Mar. 2015.

[10] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios, "BoostMap: An Embedding Method for Efficient Nearest Neighbor Retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 89–104, Jan. 2008.

[11] S. D. Bay and M. Schwabacher, "Mining distance-based outliers in near linear time with randomization and a simple pruning rule," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, 2003, p. 29.

[12] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF:identifyingdensity-basedlocal outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00*, 2000, pp. 93–104.

[13] D. O. Loftsgaarden and C. P. Quesenberry, "A nonparametric estimate of a multivariate density function," *Annals of Mathematical Statistics*, vol. 36, no. 3. pp. 1049–1051, 1965.

[14] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces," *23rd VLDB Conf.*, pp. 426–435, 1997.

[15] A. Beygelzimer, S. Kakade, and J. Langford, "Cover trees for nearest neighbor," *ICML '06 Proc. 23rd Int. Conf. Mach. Learn.*, pp. 97–104, 2006.

[16] A. Faroughi, R. Javidan, and M. Emami, "A new density estimator based on nearest and farthest neighbor," in *2016 8th International Symposium on Telecommunications (IST)*, 2016, pp. 185–190.

[17] I. Jolliffe, "Principal Component Analysis," in *Wiley StatsRef: Statistics Reference Online*, Chichester, UK: John Wiley & Sons, Ltd, 2014.

[18] H. Haddad Pajouh, R. Javidan, R. Khayami, D. Ali, and K.-K. R. Choo, "A Two-layer Dimension Reduction and Two-tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks," *IEEE Trans. Emerg. Top. Comput.*, pp. 1–1, 2016.

[19] A. F. and A. Asuncion, "UCI Machine Learning Repository," *Irvine, CA: Univ. California, Irvine*, 2010. [Online]. Available: http://archive.ics.

[20] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[21] J. Wang and X. Su, "An improved K-Means clustering algorithm," in *2011 IEEE 3rd International Conference on Communication Software and Networks*, 2011, pp. 44–46.

[22] R. T. Ng and Jiawei Han, "CLARANS: a method for clustering objects for spatial data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 5, pp. 1003–1016, Sep. 2002.

[23] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," *Inf. Syst.*, vol. 26, no. 1, pp. 35–58, Mar. 2001.

[24] G. Karypis, Eui-Hong Han, and V. Kumar, "Chameleon: hierarchical clustering using dynamic modeling," *Computer (Long. Beach. Calif).*, vol. 32, no. 8, pp. 68–75, 1999.

[25] X. X. Martin Ester, Hans-Peter Kriegel, Jörg Sander, "A Density Based Notion of Clusters in Large Spatial Databases with Noise," *Int. Conf. Knowl. Discov. Data Min.*, pp. 226–231, 1996.

[26] Zhong Wang, Yanling Hao, Zhilan Xiong, and Feng Sun, "SNN clustering kernel technique for content-based scene matching," in *2008 7th IEEE International Conference on Cybernetic Intelligent Systems*, 2008, pp. 1–6.

[27] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, I. Müller-Gorman, and A. Zimek, "Detection and Visualization of Subspace Cluster Hierarchies," in *Advances in Databases: Concepts, Systems and Applications*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 152–163.

[28] Yizong Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, 1995.

[29] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Inf. Theory*, vol. 21, no. 1, pp. 32–40, Jan. 1975.

[30] M. R. Parsaei, R. Taheri and R. Javidan "Perusing The Effect of Discretization of Data on Accuracy of Predicting Naïve Bayes Algorithm," *Curr. Res. Sci.*, no. January, pp. 457–462, 2016.

[31] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *Proc. - Int. Conf. Data Eng.*, vol. 1, no. 1, pp. 341–352, 2005.

[32] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognit.*, vol. 41, no. 1, pp. 191–203, Jan. 2008.

[33] A. K. Jain and M. H. C. Law, "Data Clustering: A User's Dilemma," in *Lecture Notes in Computer Science*, 2005, pp. 1–10.

[34] L. Fu and E. Medico, "No Title," *BMC Bioinformatics*, vol. 8, no. 1, p. 3, 2007.

[35] C. J. Veenman, M. J. T. Reinders, and E. Backer, "A maximum variance cluster algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1273–1280, Sep. 2002.

[36] M. R. Parsaei, S. M. Rostami, and R. Javidan, "A Hybrid Data Mining Approach for Intrusion Detection on Imbalanced NSL-KDD Dataset," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 6, pp. 20–25, 2016.

Azadeh Faroughi received her B.Sc. degree of Information technology engineering from Isfahan University and M.Sc. degree in computer networks from Sahand University of technology in 2010 and 2012, respectively. Now she is a Ph.D. candidate on Computer Networks in Shiraz University of Technology. Her main research interests include machine learning, data mining, computer network, network securities, wireless network, optical network and heuristic algorithms.

Reza Javidan born in 1970. He received his M.Sc. Degree in Computer Engineering (Machine Intelligence and Robotics) from Shiraz University in 1996 and his Ph.D. degree in Computer Engineering (Artificial Intelligence) from Shiraz University in 2007.Dr. Javidan has many publications in international conferences and journals regarding Image Processing, Underwater Wireless Sensor Networks (UWSNs) and Software Defined Networks (SDNs.His major fields of interest are Network security, Underwater Wireless Sensor Networks (UWSNs), Software Defined Networks (SDNs), artificial intelligence, image processing and SONAR systems. Dr. Javidan is now member of faculty and lecturer in Department of Computer Engineering and Information Technology in Shiraz University of Technology.