Initial Free Self-Organizing Map by Aggregation

Amin Allahyar[1] and Hadi Sadoghi Yazdi[1,2]

*[1]Department of Computer Engineering,
Ferdowsi University of Mashhad, Mashhad, Iran
[2]Center of Excellence on Soft Computing and Intelligent Information Processing,
Ferdowsi University of Mashhad
Amin.Allahyar@stu.um.ac.ir, h-sadoghi@um.ac.ir*

Abstract

Initialization parameters of Self-organizing map networks (SOM) have a great influence in their final result. This is the only known problem of SOM networks. The uncertainty of solution is resolved with the fusion methods in many machine learning algorithms. Motivated by this application, we aim to reduce the uncertainty in SOM final result and provide a more robust solution. This is done by aggregating the results obtained from several run of SOM network. The objective in uniting several SOM run is to reduce the effect of two above factors and reach a more appropriate solution. To achieve this, a number of methods for integration of SOM solutions have been proposed and their characteristics are discussed in detail. Finally, these approaches are experimentally compared with some recently proposed SOM networks and their effectiveness are investigated.

Keywords: *Self-Organizing Map; Clustering Fusion; Initial Network; Data Sequence; Robustness; Cluster Validity*

1. Introduction

Clustering and Classification are to main fields of research in machine learning [9]. Classification is a supervised learning approach and basically is a search to find the optimal hyper planes for discriminating two or more class of input data. On the other hand clustering is an unsupervised learning method which essentially groups the given data such that *homogeneity* and *heterogeneity* of clusters achieved simultaneously. In the other word, clustering aims to group the given data point such that data point in each cluster has maximum similarity (homogeneity) while has a considerable amount of differences to other clusters (heterogeneity) [9]. Cluster analysis organizes data by abstracting underlying structure either as a grouping of individuals or as hierarchy of groups. It has a wide variety of application in Data Mining [17], Natural Language Processing [16], Image Analysis [10], Bioinformatics [15] and *etc*.

Many clustering algorithms have been proposed in the literature and each has a set of advantage and disadvantages. Sequential clustering [15], divisive clustering [8], hierarchal clustering [3], agglomerative clustering [4], centroid based clustering [6] and adaptive clustering [13] are some of the most widely known clustering approaches. Among these algorithms, the adaptive clustering gained more popularity in large scale problems owing to low computational cost and memory requirement. In addition these kinds of clustering algorithms natively support online learning based on an intuitive procedure and simple implementation. One of the most known centroid based algorithm Self-Organizing Map (SOM) algorithm proposed by Kohonen [13]. The SOM algorithm is essentially a network of centers which adapt a group of centers when new data point is acquired. A schematic of this initial network is demonstrated in Figure 1. It starts with a randomly chosen centers and aim to find the best possible location for centroids with a *greedy* routine for attained data stream. In fact, random initialization and greedy routine for center adaptation are the only disadvantages of these algorithms which stop them to reach a global optimal clustering solution.

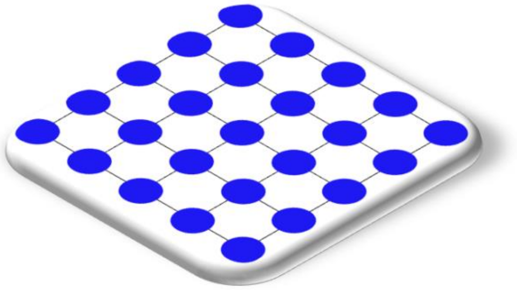


Figure 1. Demostration of starting network in SOM algorithm. The blue circles are centers (nodes) and the thin black lines are the neighborhood connections

As discussed, the limitations of adaptive based clustering is the dependence on the initial network of centers and sequence of data stream arrival [1]. More specifically, the SOM algorithm will produce a different result with different initial network of centroids or different sequence of input data and thus it is not guaranteed that the global minimum will be found. Alternatively, appropriate initialization of network will lead to fast convergence of the algorithm to a suitable result. Therefore, it is desirable to execute the algorithm several times in order to increase the reliability of the final results. However, as SOM is an unsupervised algorithm, measuring the quality of delivered results and determining the best one is a difficult task.

The diversity on result is also encountered in classification. It is met when the result obtained from a group of trained classifiers are aggregated to reach a more accurate prediction. This process is called *Ensemble Classifier* and has a wide range of application in real world problems including: handwritten recognition [19], web page classification [18], noise elimination [12] and etc. Ensemble classifier is also used to cope with uncertainty [5]. This is even more close to our problem. Since primary values for centers in the starting network are selected randomly, several executions of SOM algorithms may lead to different clustering results. Here we want to find suitable centers from results obtained from set of SOM clustering algorithms. Combining the results of several executions of SOM algorithms in a proper way can help us to reduce errors and obtain a better result.

Although it seems promising, but interpretation of results obtained from an unsupervised procedure and measuring the quality of each cluster in this environment is a challenging task. To handle this, we use *Cluster Validation* methods. Cluster validation refers to procedure of evaluating the result of cluster analysis in quantitative and objective manners. In this paper we present a group of intuitive procedures for quantitative verification of the clustering results obtained from SOMM cluster analysis. Then based on this quantization, we weight each result and produce the final cluster centers based on their weighted average. Finally to investigate the effectiveness of these procedures, comprehensive experimental results are performed on synthetic and UCI data set and their corresponding results are reported.

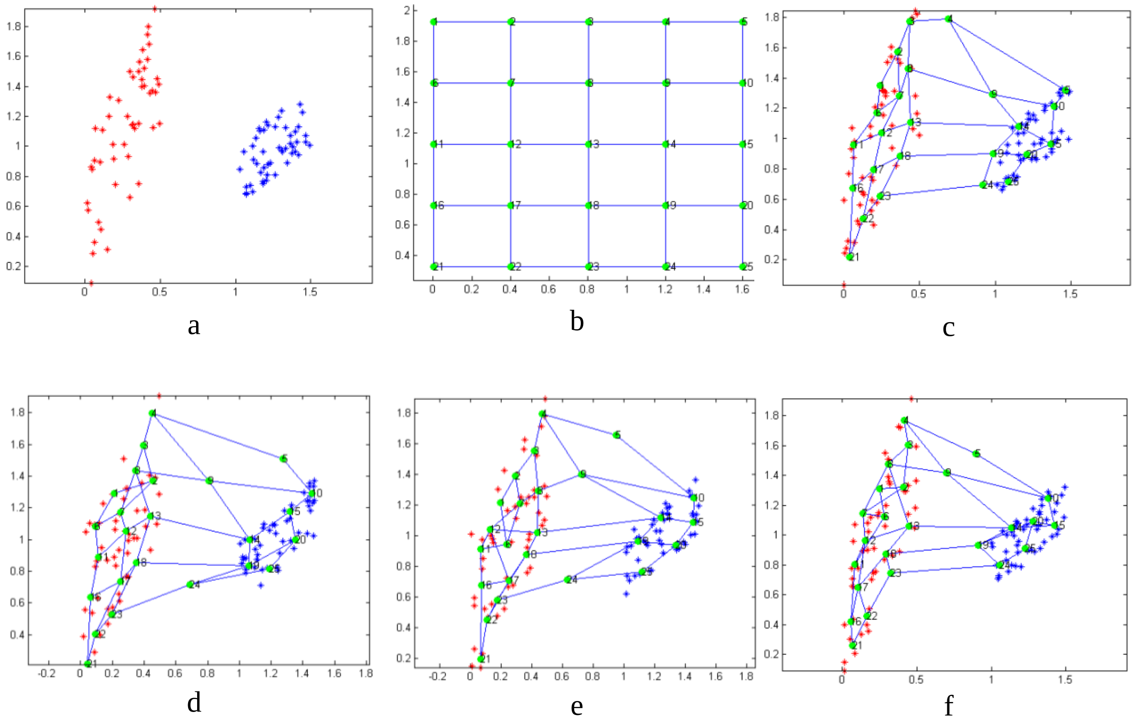


Figure 2. Demonstration of result diversity with each individual runs of SOM algorithm. a) input data, b) initial network, c-f) four different run of SOM algorithm

The rest of this paper is organized as follows. In the preliminary section, the self-organizing map algorithm is introduces and its limitation will be demonstrated. We present a review of previous ensemble clustering studies in Section 3. Section 4 is dedicated to our proposed cluster analysis schema and its corresponding discussion. The experimental results are given in Section 5 and finally Section 6 concludes the paper.

2. Preliminary

First we define our notation in this paper. Scripted letters such as $\mathcal{C}$ and $\mathcal{M}$ represent sets. Capital letters like X and W are matrixes while bold lower case letters show column vectors e.g. $\boldsymbol{x}$ and $\boldsymbol{u}$. Lower case letters indicate scalars, *e.g.*, n and d. Similar to popular notation we use subscripts to index elements in matrixes or vectors. For example x_i is i-th element of vector $\boldsymbol{x}$ and $\boldsymbol{w}_j$ is the j-th column vector in matrix W. The notation $|\mathcal{C}|$ is number of items in the set $\mathcal{C}$ and vector norm $\|.\|$ is the l_2 norm. So by definition we have $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$. In addition we show the value of $\boldsymbol{\mu}$ in t-th iteration with $\boldsymbol{\mu}^{(t)}$ notation. Let $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\} \in \mathbb{R}^{d\times n}$ be n given d dimensional samples. Also there are m clusters available $\mathrm{M} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_m\}$ so the whole dataset is divided into m set where $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_m\}$.

2.1. Self-Organizing Map Algorithm

Self-Organizing Map (SOM) algorithm was proposed by Kohonen [13] and since then has been proven to be beneficial in many application including natural language processing [11], process monitoring [2], cancer diagnosis [7] and *etc*. SOM belongs to a wide category of clustering algorithm called *competitive learning networks*. Because the *topology preservation* property, it can be used to detect the inherent structure of high dimensional data and represent this structure in two dimensional lattices. The topology preservation implicates that this representation preserve the relative distance between points in high dimensional space. In the other word, points that are close together in high dimensional space, will still kept close in low dimensional representation. This is the main reason that SOM can be effectively applied in data analysis and clustering (unsupervised learning).

The SOM algorithm is consists of several steps. At first step, the desired number of centers (nodes) are randomly initialized. Then their neighbors connection are assigned. These connections for sample SOM network in Figure 1 are demonstrated with black thin lines. Generally each node has connection to its direct neighbor in four main directions. Then by arrival of each data point, the distance to every center of the network is calculated and the closest one is considered as *Best Matching Unit* (BMU) as (1).

$$BMU = \underset{j}{\operatorname{argmin}} \quad \left\| \boldsymbol{x}^{(t)} - \boldsymbol{v}_j^{(t)} \right\| \tag{1}$$

Where $\boldsymbol{x}^{(t)}$ is the t-th input sample acquired, $\boldsymbol{v}_j^{(t)}$ is j-th center in time t. Then the BMU node is updated to get closer to input data $\boldsymbol{x}^{(t)}$ by (2).

$$\boldsymbol{v}_{BMU}^{(t+1)} = \boldsymbol{v}_{BMU}^{(t)} + \eta_{BMU}^{(t)} \times \left(\boldsymbol{x}^{(t)} - \boldsymbol{v}_{BMU}^{(t)} \right) \tag{2}$$

In this formulation, $\eta_{BMU}^{(t)}$ is the BMU learning rate in time t which decays over time. The weight update in (2) is visually described in Figure 3.

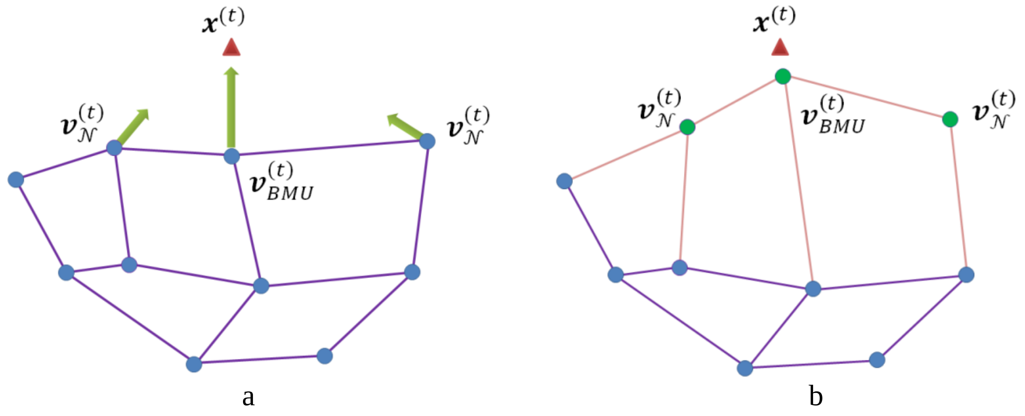

$\boldsymbol{x}^{(t)}$
$\boldsymbol{v}_{\mathcal{N}}^{(t)}$
$\boldsymbol{v}_{\mathcal{N}}^{(t)}$
$\boldsymbol{v}_{BMU}^{(t)}$
a
$\boldsymbol{x}^{(t)}$
$\boldsymbol{v}_{\mathcal{N}}^{(t)}$
$\boldsymbol{v}_{BMU}^{(t)}$
$\boldsymbol{v}_{\mathcal{N}}^{(t)}$
b

Figure 3. Representation of weight update process. a) finding the best matching unit and its neighborhood nodes; b) moving toward the input data location. The movement of best matching unit is larger than neighbor nodes toward input data location

The decay behavior of $\eta_{BMU}^{(t)}$ can be simulated by many functions. For example exponential decay function can be used which defined as (3).

(3)

$$\eta_{BMU}^{(t)} = e^{-(\frac{t}{\lambda_{BMU}})} \times \eta_{BMU}^{(0)}$$

Where e is the neperian number and λ_{BMU} is the BMU decay parameter that determine how fast the learning rate should drop. The effect of this parameter is represented in Figure 4. $\eta_{BMU}^{(0)}$ is the initial learning rate and need to be set according to data structure scheme.

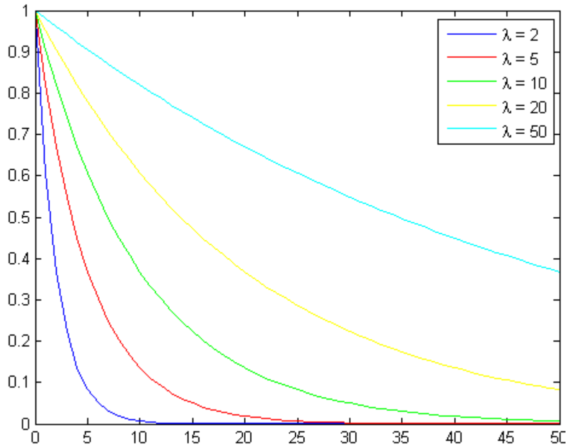


Figure 4. Effect of decay parameter λ. The smaller λ result in quicker drop of value

Next we need to update the *topology neighbors* of BMU unit. The topology neighbors are defined as the units that can be reached from BMU unit with at most θ step where θ is a user defined parameter. Assuming that the index of these nodes are stored in $\mathcal{N}$, the updating process can be defined as (4)

(4)

$$\boldsymbol{v}_{\mathcal{N}}^{(t+1)} = \boldsymbol{v}_{\mathcal{N}}^{(t)} + \eta_{\mathcal{N}}^{(t)} \times \left(\boldsymbol{x}^{(t)} - \boldsymbol{v}_{\mathcal{N}}^{(t)}\right)$$

Where $\eta_{\mathcal{N}}^{(t)}$ is the learning neighborhood learning rate. Where regularly $\eta_{\mathcal{N}}^{(t)} < \eta_{BMU}^{(t)}$. This will cause the neighbor nodes moves with a smaller step compared to BMU node which is demonstrated in Figure 3. In order to achieve a smoother network, a decay parameter π is also utilized in the literatures that relate the neighbor movement to the distance between BMU and that specific neighbor nodes. π can be defined as (5).

(5)

$$\pi_{\mathcal{N}}^{(t)} = e^{-(\frac{\left\|\boldsymbol{v}_{BMU}^{(t)} - \boldsymbol{v}_{\mathcal{N}}^{(t)}\right\|}{\lambda_{\mathcal{N}}})}$$

Where $\lambda_{\mathcal{N}}$ is the decay parameter similar to (3). The decay effect of $\pi_{\mathcal{N}}^{(t)}$ is visually demonstrated in Figure 5.

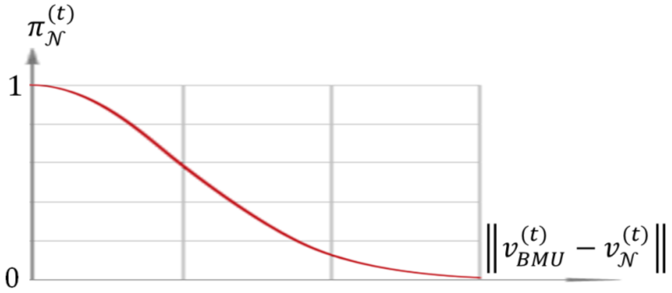


Figure 5. The decay effect of $\pi_{\mathcal{N}}^{(t)}$ with increase of distance between best matching unit and neighbor units

Using (5) measure, (4) can be extended to (6).

$$\boldsymbol{v}_{\mathcal{N}}^{(t+1)} = \boldsymbol{v}_{\mathcal{N}}^{(t)} + \pi_{\mathcal{N}}^{(t)} \times \eta_{\mathcal{N}}^{(t)} \times \left(\boldsymbol{x}^{(t)} - \boldsymbol{v}_{\mathcal{N}}^{(t)} \right) \tag{6}$$

This way, the movement effect would gradually decrease as distance between BMU and neighbor node increase. Thus a smoother network will produces after the training phase finished. The pseudo code of SOM algorithm is given in Algorithm 1.

Algorithm 1: Self Organizing Map

Input: Matrix of data points $X \in \mathbb{R}^{n\times d}$, number of desired centers m or network size $z \times z = m$, number of neighbors θ, BMU and neighbor initial decay $\eta_{BMU}^{(t)}, \eta_{\mathcal{N}}^{(t)}$, decay ratio for BMU and neighbor nodes $\lambda_{BMU}, \lambda_{\mathcal{N}}$.

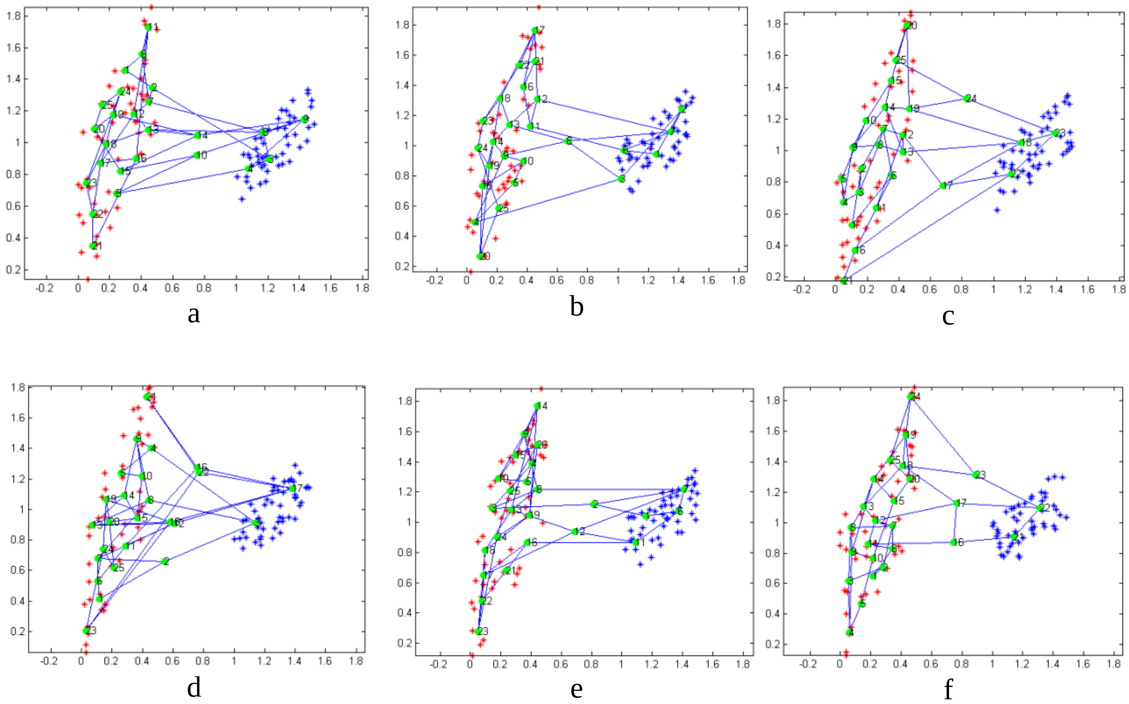
```
1. Initialize the network matrix $V \in \mathbb{R}^{m\times d}$ randomly.
2. Initialize the neighborhood connections. Each node is connected to its four closest neighbors.
3. $t = 1$.
4. Acquire a data point $\boldsymbol{x}^{(t)}$ from input device.
5. Calculate the distance between $\boldsymbol{x}^{(t)}$ and every node on the network.
6. Determine the BMU unit by (1).
7. Calculate the distance between $\boldsymbol{v}_{BMU}^{(t)}$ and every other node on the network and select indexes corresponding to $\theta$ of the closest one as BMU neighbors $\mathcal{N}$.
8. Update the weights of BMU by (2)
9. Update the weights of $\mathcal{N}$ by (4) or (6).
10. $t = t + 1$.
11. If the change in node weights are considerable or the maximum iteration is not reached continue to step 3.
```

Output: The network matrix $V \in \mathbb{R}^{m\times d}$

2.2. Limitation of SOM Algorithm

By analyzing Algorithm 1, it can be realized that SOM algorithm requires several parameter to be defined appropriately in order to reach a promising result. In addition, the learning process starts from a random initialization point and the weight update is actually a greedy method that tries to reduce the distance between acquired data point and its nearest nodes. Therefore, SOM algorithm delivers a different result with every individual run.

This issue is represented in Figure 1 when the network weights kept constant while the sequence of data points changes randomly. Figure 6 shows the diversity of solutions when the network weights randomly chosen while the sequence kept unchanged. Same data has been used in this experiment with parameters set to following values: $m = 25$, $\theta = 4$, $\eta_{BMU}^{(t)} = 0.4$, $\eta_{\mathcal{N}}^{(t)} = 0.01$, $\lambda_{BMU} = 10^4$, $\lambda_{\mathcal{N}} = 10^4$ 10^4 iteration. The diversity of the result is clearly observable in these results. For example, for Blue class, different number of centers ranging from 2 to 5 is allocated.



a b c

d e f

Figure 6. Demonstration of result diversity with each individual runs of SOM algorithm. a-f) show these results with a same sequence of data but with different network initialization

In the next section, we first propose a group of measures to weight the quality of result provided by each individual SOM algorithm. Then by proposing several aggregation methods we aim to reach a more appropriate final result.

3. Related Work

The problem of combining results of classification methods is a well-studied subject. On the other hand, a few number of authors investigated the possibility of clustering result aggregation. Dimitrios proposed a multi-clustering fusion method [7]. The algorithm consists of two sequential procedures. The *Partitioning* procedure which partitions whole data points into clusters, and the *Fusion* procedure which specifies the true structure of the data. The initial number of clusters and the number of iterations are defined for the partitioning procedure, in which a clustering algorithm and a voting scheme are implemented in order to produce a distinct partition of the data set. The partitioning procedure applies the same clustering algorithm for a number of iterations in order to distinctly partition data points to a predefined number of cluster $\widehat{m}$. This number is regularly selected large compared to final desired number of cluster m. The Fusion procedure is developed based on the neighborhood relation among clusters. This procedure starts with $\widehat{m}$ clusters detected by partitioning procedure. Then after removing the clusters with zero data points, it merges the ones which are closest to each other. The result will be an optimal number of arbitrary shape clusters for the given data set based on some specified criteria.

Yang proposed a new combination method for fuzzy clustering algorithms [20]. The majority voting rule is applied to combine different clustering algorithms, rather than similar ones. Three Fuzzy C-Means clustering algorithms namely FCM, G-K, and PCA are used. Each of these (except G-K that is initialized by the result of FCM) is initialized by three center initialization methods namely CCIA, kd-tree, and I_{12} respectively. Therefore, there are a total of nine fuzzy clustering algorithms denoted by FCM-CCIA, FCM-I_{12}, FCM-kd-tree, G-K-CCIA, G-KI_{12}, G-K-kd-tree, PCA-CCIA, PCA-I_{12} and PCA-kd-tree respectively. They are merged together into a combinatorial fuzzy C-Means algorithm, denoted by CFCM by using fuzzy simple majority voting rule. Unlike simple majority voting rule which assign a class label l_i to each data point $\boldsymbol{x}_i \;\; i = \{1,2, \dots, n\}$, the membership degree of $\boldsymbol{x}_i$ belonging to each cluster $\mathcal{X}_j \;\; j = \{1,2, \dots, m\}$ is calculated. Finally the center of each cluster is determined by aggregating all fuzzy clusters. This is done by computing the membership degree of $\boldsymbol{x}_i$ belonging to each class derived from all fuzzy clustering algorithms.

4. Proposed Method

In this section we propose our *Ensemble Clustering* method. Similar to definition of ensemble classification, the output obtained from number of individual SOM clustering is combined and the outcome of this fusion is considered as final clustering result. The primary goal in combining several clustering analyzers is to improve clustering results beyond what is achievable by single run of SOM clustering algorithm. This procedure is performed in three main steps.

- **Chaos Generation Step:** Generating the initial random values for initial SOMs weight network.
- **Matching Step**: Because the procedure is unsupervised, it may be required to rearrange cluster centers to unify the obtained clusters.
- **Fusion Step:** The main step of proposed algorithm which combine the attained clusters in order to determine the final results.

A schematic of this procedure is demonstrated in Figure 7.

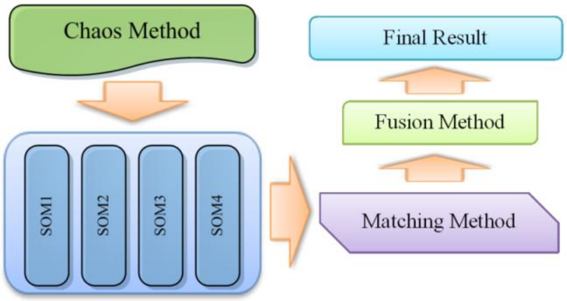


Figure 7. Schematic of proposed method to combine multiple Self Organizing Map algorithms

4.1. Chaos Generation Step

As mentioned before the present article seeks to combine the obtained result of several SOM clustering in order to come up with the more suitable clustering result. It is discussed that the SOM method is dependent on the starting point, meaning that with different starting points (the values of center matrix) different results will be obtained and it many cases it will be stopped in local minima. Although SOMs were executed over a common data set, different results were obtained due to different initial values. It issue are experimentally investigated and the result where demonstrated in Figure 2 and Figure 6. Therefore using a suitable method for determining an appropriate starting point can lead to acceptable final results.

We used the *Chaos Method* for generating random starting points. The *Chaos Method* use (7) formula to generate random points.

$$z(k+1) = \eta z(k)(1 - z(k)) \tag{7}$$

Where η is the chaos attractor. The range of chaos values is [0, 1] and $z(k+1)$ is the next value. The value of $z(k)$ is in (0, 1) and cannot include $\{0.25, 0.5, 0.75\}$. The reason for choosing chaos method is that it obtains random points that are not identical. As described, one of the randomness in SOM algorithm is the starting points (*e.g.*, cluster center matrix). Thus if two individual SOM algorithm start with the same initialization point, they both will produce an identical final results if the sequence of input data kept constant. However the input sequence does not governed by SOM algorithm and the weight vectors need to be random because we don’t know the cluster centers a priory and data points are given in a sequential manner. To stimulate this behavior, we need a method that gives us unequal random initial points.

4.2. Matching Step

An important issue in unsupervised learning and more specifically in clustering is that the order of produced clusters may change with every run. Similar problem can emerge in SOM algorithm because the center matrix may be obtained with different ordering in each run. Therefore, we have to come up with a method for ordering cluster centers before combining the answers. The following method will be used for matching cluster centers.

Since cluster centers may have different coordinates from each other, each cluster can be ordered based on the coordinates of the obtained centers. Therefore, for each run of clustering algorithm, the length and direction of the center vector in the d dimensional space will be considers as two parameters that can determine the real position of a center $\boldsymbol{v}_j$ in its corresponding center matrix V. By going through this procedure each cluster can be identified in every run of SOM algorithm.

4.2. Fusion Step

In this section we proposed a group of procedures to combine result of several SOM algorithms. In addition we explain the advantage or drawbacks of each method in detail. To ensure that these methods are helpful in SOM combining, extensive experimental result will be performed on the next section.

4.2.1. Simple Averaging Combination

One of the first methods that are proposed to ensemble several classifiers was the *Simple Averaging* method. Inspired by this work we propose the Simple Average Clustering Combination (SACC). In this method, the average of mean vectors obtained from each center matrix will be calculated. For example if the SOM algorithm performed 100 times on a data with three clusters, 100 answers will finally obtained for each of the three clusters. In the simple averaging method, the average of the 100 obtained answers will be considered as the final answer for each cluster. The average of each cluster will be calculated using (5).

$$\boldsymbol{v}_k = \frac{1}{t}\sum_{t=1}^{\rho}\sum_{j=1}^{m}\boldsymbol{v}_j^{(t)} \ , \quad k = \{1,2,\dots,m\} \tag{5}$$

Where ρ is the total number of SOM algorithms that is going to be combined. $\boldsymbol{v}_k$ is the final cluster center calculated from average of cluster centers $\boldsymbol{v}_j^{(t)}$ in every t runs.

4.2.2. Weighted Averaging Combination

In the simple averaging method, all the obtained answers (center matrixes) have an equal effect on the final answer regardless of the level of their correctness. Thus, the simple averaging method only yields correct answers when most of the solutions from clustering methods are correct. It seems more appropriate to rank the results obtained from each SOM in such a way that those which produced a better results receive a higher grade. To accomplish this, two method proposed as follows:

I. Scatter Ratio (SR)

Inspired by clustering validity measures, we propose to use compactness of each cluster as a measure of cluster quality. In this measurement, we want that the center vectors $\boldsymbol{v}_j$ where $j = \{1,2,\dots,m\}$ has considerable amount of distance to each other while the variance σ_j be a small value. By this measurement, the cluster has a high rank if the mean vector of that cluster lies on a far location while it has a small variance. This can be mathematically formulated as (6).

$$\omega^{(t)} = Rank_{SR}(t) = \sum_{r=1}^{d} \sum_{j=1}^{m} \frac{v_{rj}^{(t)}}{\sigma_{rj}^{(t)}} \tag{6}$$

Where $\omega^{(t)}$ is the rank of clustering algorithm t, d is the input data dimension and m is the number of desired clusters. In this formulation $v_{rj}^{(t)}$ indicate the r–th dimension of j-th mean vector corresponding to t-th run. Similarly $\sigma_{rj}^{(t)}$ represents the variance of r–th dimension of j-th cluster corresponding to t-th run. After calculating the corresponding rank for each run of SOM, we can use it as a measure of clustering quality. This way each clustering algorithm influences the mean point selection by its quality (*i.e.*, its rank). This influence can be explicitly added to the weighting process as (7).

$$\boldsymbol{v}_k = \frac{1}{\sum_{t=1}^{\rho} \omega_t} \sum_{t=1}^{\rho} \sum_{j=1}^{m} \omega_t \boldsymbol{v}_j^{(t)} \ , \quad k = \{1,2,\dots,m\} \tag{7}$$

In the weighted averaging method, the results from each clustering method will affect the final result based on its correctness. Thus we can lower the effect of incorrect answers and increase the effect of correct answers.

II. Fuzzy Scatter Ratio (FSR)

FCM is a clustering method that states how much a data belongs to all existing clusters (cluster membership) in a fuzzy manner with the sum of the levels being one. Therefore if the membership of a data point in a certain cluster is more than its membership in other clusters, it is assumed to be a member of that specific cluster. As a result, the membership value of every point in a specific cluster is a meaningful measure of cluster compactness. Thus it seems reasonable to incorporate such knowledge to the weighting process. The fuzzy membership value can be calculated by (8).

$$u_{ij} = \sum_{k=1}^{m} \left(\frac{\|\boldsymbol{x}_i - \boldsymbol{v}_j\|}{\|\boldsymbol{x}_i - \boldsymbol{v}_k\|} \right)^{-\frac{2}{\tau-1}} \tag{8}$$

Where θ indicate the fuzziness weighting exponent and has range $1 < \theta < \infty$ and regularly defined as $\theta = 2$. If $\theta \rightarrow 1$, the membership degree of each data point becomes more close to crisp version, *e.g.*, K-Means algorithm and if $\theta \rightarrow \infty$ the membership of each data for every clusters become more close to each other. After calculation of fuzzy membership value, the Fuzzy Scatter Ratio (FSR) criterion can be defined as (9).

$$\omega^{(t)} = Rank_{FSR}(t) = \sum_{r=1}^{d} \sum_{j=1}^{m} \frac{v_{rj}^{(t)} \bar{u}_j^{(t)}}{\sigma_{rj}^{(t)}} \tag{9}$$

Where $\bar{u}_j^{(t)}$ indicate the average membership value of data point in j-th cluster for t-th run of SOM algorithm and defined as (10).

$$\bar{u}_j^{(t)} = \frac{1}{n}\sum_{i=1}^{n} u_{ij}^{(t)} \tag{10}$$

The FSR rank of each cluster j can be incorporated into weighting average (7) and final center vector is calculated. Similar to previous discussion, a cluster with higher average membership values will have a higher rank and thus it will has more influence in determining the final center of each cluster.

4.2.3. Cluster Level Combination (CL)

In the previous criterions, we measured the cluster quality for each run of SOM algorithm. But it is clear that there may be a circumstance where in a specific run of SOM algorithm, some clusters are correctly determined while others does not. In this situation it is more appropriate to measure each clusters individually. In the other word the measurement is performed in cluster level instead of SOM run level. Without this, the clustering method may get a low rank because of an improper determined cluster regardless of other clusters which may be properly shaped. Another disadvantage of previous criterions is that every cluster is weighted by rank of whole SOM run. This is obviously not desirable. When a clustering method is ranked, all contained clusters will have an equal impact on the final answer. But it is desirable to have better clusters has a higher impact on final result.

Based on the above discussion, grading in cluster level is seems more favorable than FCM run level. Superior clusters will be chosen from set of the clustering results obtained from every run of FCM algorithms. In the other word, the clusters will be combined instead of combining the clustering method results. The first step is to find a suitable criterion for determining the correctness of each individual clusters. As described previously, a dense cluster has a higher membership values and are considered to be more proper. In addition, the center of a cluster is within the cluster and almost in the middle of the data. In other words, data elements have almost the same distance from the center of the cluster. Inspired by these properties, we need the following parameter to rank each individual clusters:

- **Distance of Data and the Cluster Center:** Shorter distance between each data and its corresponding cluster center indicates a suitable cluster center. Lower distance indicates that the center is close to data elements containing in the cluster.
- **Variance of Data Resides in a Cluster:** Lower variance of data distance from the center indicates that cluster center is close to the middle of the cluster. If the cluster center is not in the middle of the cluster, variance of data distance from cluster center will be more than the last situation.
- **Ratio of average cluster data to variance of cluster data:** this parameter has already been explained. Higher ratios of average cluster data to data variance indicate that the cluster is denser and data elements are closer to each other.

- **Average membership value of cluster data:** Calculated fuzzy membership value is a relative measure. If a data element has a higher membership value in a certain cluster, it belongs to that cluster. Higher membership values of data in a cluster indicate that the cluster is a suitable one and all data elements belong to the cluster with a high level of certainty.
- **Variance of membership values of cluster data:** if membership values are similar and all of them almost the same, we will have a low variance of membership values. Low variance and high membership values indicate that almost all the data belong to the cluster with a high level of certainty and if a cluster has this characteristic it means that it has been chosen correctly. Data in poor clusters often have low membership values.

Each cluster is ranked based on the above criterions defined by (11).

$$\omega_j^{(t)} = Rank_{CL_j}(t) = \frac{\gamma_j^{(t)} \times \bar{u}_j^{(t)}}{var(\mathcal{X}_j^{(t)}) \times var(u_j^{(t)})} \times \frac{1}{\frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}_i - \boldsymbol{v}_j^{(t)}\right\| \times var(\sum_{i=1}^{n}\left\|\boldsymbol{x}_i - \boldsymbol{v}_j^{(t)}\right\|)} \tag{11}$$

In this formulation $var(.)$ is the variance of argument and $\gamma_j^{(t)}$ is the average distance between each data point and its corresponding mean vector and can be calculated as (12).

$$\gamma_j^{(t)} = \frac{1}{\left|\mathcal{X}_j^{(t)}\right|}\sum_{\boldsymbol{x}_i \in \mathcal{X}_j^{(t)}} \left\|\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(t)}\right\|^2 \tag{12}$$

After calculating the ranking for each m individual clusters in every ρ run of SOM, we can insert this rank in (7) and calculate the final m centers.

Using this formulation, every clusters found by the clustering methods will have impacts on the final answer. Although very poorly constructed clusters have a small effect on the answer, but it still affects the final answer. Thus a slight division from the proper cluster center may be seen. In order to obtain better solutions, the clusters with lower grades will be excluded so that only suitable clusters will affect the calculation of final clusters. To determine the exact number of clusters that should be included in the result, the average rank for each cluster will be calculated and only ones with higher grades than the average will be included. This will help us to reject poor clusters with low grades. Therefore (11) is more suitable to be defined as (13).

$$\mathcal{S}_j = \{j|\omega_j^{(t)} > \frac{1}{\rho}\sum_{t=1}^{\rho}\omega_j^{(t)}\} \tag{13}$$

Thus the weighted average of final result can be written as (14).

$$\boldsymbol{v}_k = \frac{1}{\sum_{t\in\mathcal{S}_k}\omega_t}\sum_{t\in\mathcal{S}_k}\sum_{j=1}^{m}\omega_t\boldsymbol{v}_j^{(t)} \tag{14}$$

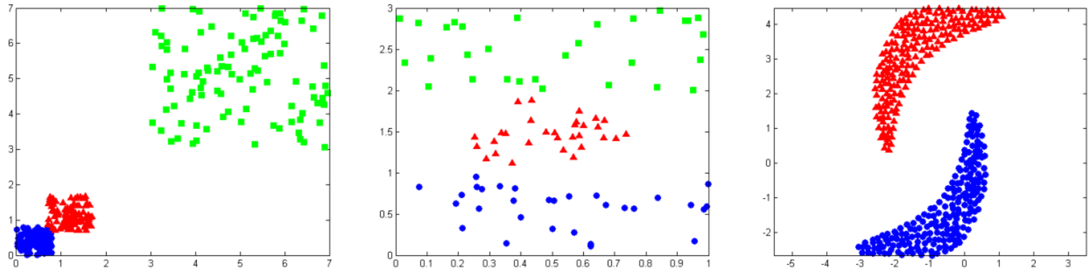
In the next section we investigate the effectiveness of these algorithms. To achieve this, we need to measure the quality of clusters. This is done by applying several clustering validation methods. Then these results are compared to the case when no combination is used.

5. Experimental Result

In this section we experimentally investigate the effectiveness of our proposed criterions in compared to two clustering combination method introduced in the related work section including: Multi-Clustering Fusion (MCF) proposed by Dimitrios and Yang's Fuzzy C-Mean Combination (FCC) method.

5.1. Setup

In order to be able to compare the described algorithms, we need to generate a set of SOM initialization point, *i.e.*, the fuzzy membership matrix. 100 initialization points is generated using the Chaos method described in 4.1 Section. These initialization points are fed to same number of SOM algorithm and their final membership matrix is stored. In this experiment, we used an Intel Quad-Core 2.5 GHZ computer with 4GB RAM on windows 7 64bit. Also Matlab 2012a is used as simulation software. For synthetic dataset effectiveness analysis we use previous toys datasets DATA1, DATA2. We also add another synthetic dataset which is to some extent a non-linear separable dataset. We refer to it as DATA3. These toy datasets are demonstrated in Figure 8 this section we experimentally investigate the effectiveness of our proposed criterions in compared to two clustering combination method introduced in the related work section including: Multi-Clustering Fusion (MCF) proposed by Dimitrios and Yang's Fuzzy C-Mean Combination (FCC) method.



a) DATA1 b) DATA2 c) DATA3

Figure 8. Demonstration of three toy datasets

We also used four UCI[1] repository dataset including: Iris, Protein, Diabetes and Breast-W in our experiment. Properties of these datasets are given in Table 1.

[1] Available at http://www.ics.uci.edu/mlearn/MLRepository.html.

Table 1. Properties of data sets used for experiments. As previously defined, n is number of data points, d is data dimension, m indicates number of clusters

	n	d	m	$\mathcal{X}$
DATA1	300	2	3	{100, 100 ,100}
DATA2	90	2	3	{30, 30, 30}
DATA3	400	2	2	{200, 200}
Iris	150	4	3	{50, 50, 50}
Wine	178	13	3	{59, 71, 48}
Diabetes	768	8	2	{268, 500}
Breast-W	683	9	2	{444, 239}

5.2. Cluster Validity Measurement

In order to evaluate each clustering results, we need to measure the quality of each clusters. As discussed this process is called cluster validity and is a well-studied subject. In this paper we used several clustering measurement including: Average Partition Density (APD) [9], Average Within Cluster Distance (AWCD) [13], Bezdek Partition Coefficient (F) [5] and Xie-Beni index (XB) [19]. This measure has specific properties and can be briefly described as follows:

- **Average Partition Density (APD):** This measure is based on cluster density. The higher APD values indicate higher densities for clusters. Thus any cluster with upper value of APD will be the most suitable cluster
- **Average within Cluster Distance (AWCD):** calculates the average distance of the data in each cluster from the center. Lower AWCD values indicate higher distance between the data and their centers which is desirable.
- **Bezdek Partition Coefficient (F):** In this measure, the cluster overlapping will be investigated. Lower F values indicate that clusters are more distant from each other and have lower overlapping. If F = 1 then all the clusters are separate from each other and there is no overlapping between the clusters. Thus the higher F values are desirable.
- **Xie-Beni index (XB):** The density of the clusters, the sparsity and distance of clusters from each other are measured in XB criterion. Lower XB values indicate that clusters are denser and have greater distance from each other, so lower XB values are desirable.

5.3. Quality Comparison by Given Labels

In this subsection, we compare the result of proposed measurement with other algorithm in term of ground truth label accuracy. Table 2 represents the result of such experiment which is repeated 25 times for each algorithm and its mean and variance is reported.

Table 2. Result of comparing proposed method and other clustering combination methods in term of accuracy

	DATA1	DATA2	DATA3	Iris	Wine	Diabetes	Breast-W
MCF	93.34±2.34	91.25±1.62	92.12±0.12	89.52±1.21	**69.54±4.15**	65.79±0.51	95.51±0.62
FCC	94.74±3.21	92.84±1.51	**92.13±0.08**	89.54±1.07	68.76±4.72	65.86±0.42	95.74±0.56
SR	93.52±3.65	92.70±2.15	92.04±0.24	86.12±1.61	67.15±7.17	65.61±0.96	94.13±0.57
FSR	96.26±2.74	93.25±1.73	92.10±0.14	**89.55±1.28**	68.73±4.51	65.89±0.65	95.81±0.26
CL	**97.26±2.41**	**94.61±1.62**	92.11±0.16	89.47±1.21	69.15±4.62	**66.02±0.84**	**96.21±0.75**

By this experiment, it can be seen that the CL and FSR achieved a good result in compared to other algorithms. It should be noted that in the two of these datasets the old algorithms showed a higher accuracy.

5.4. Quality Comparison by Measurement

In this experiment we use the defined measurement to investigate the quality of algorithm. Table 3 illustrates the results of running different validation functions on different data series. The objective is to maximize XB and AWCD values and minimize F and APD values. Result of XB, F, AWCD and APD experiment is given in Figure 4, 5, 6 and 7 respectively.

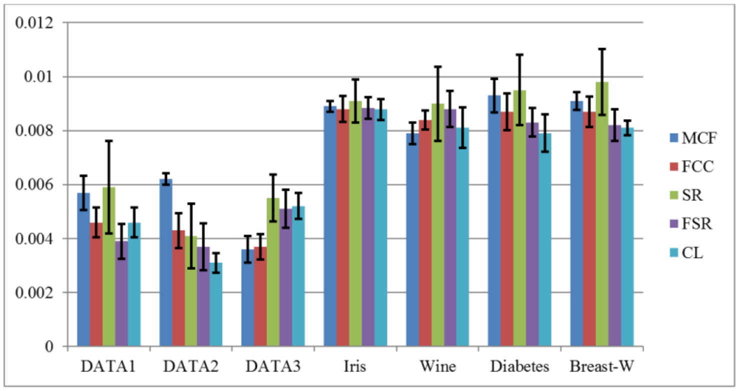


Figure 9. Demonstration of XB measure

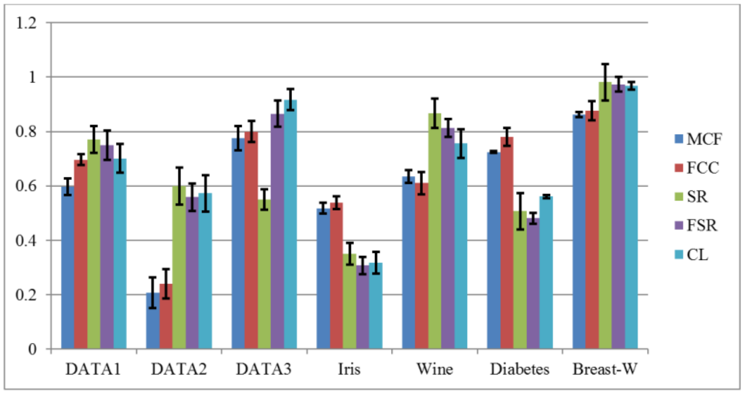


Figure 10. Demonstration of APD measure

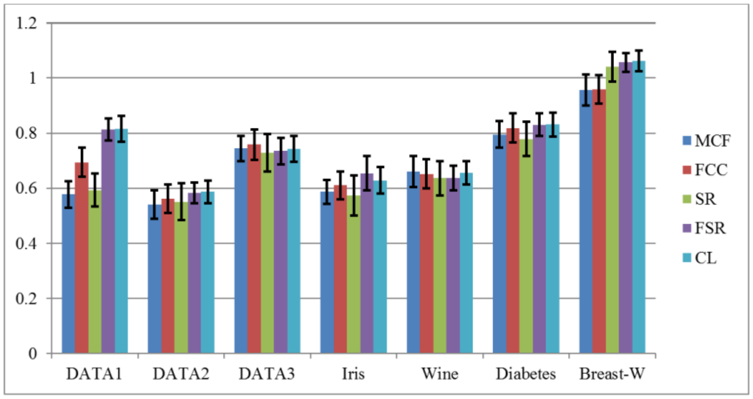


Figure 11. Demonstration of F measure

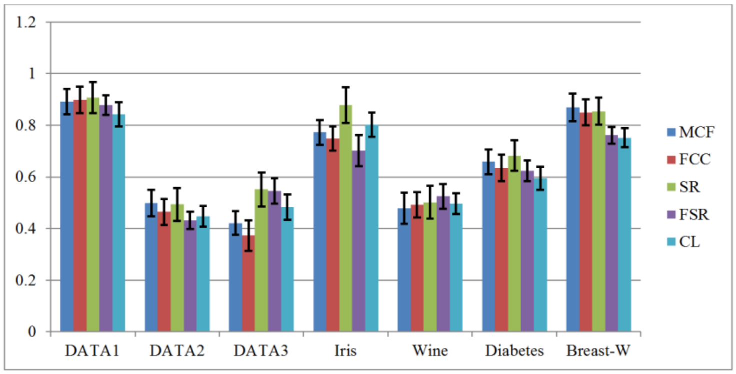


Figure 12. Demonstration of AWCD measure

By investigating the result, it can be clearly seen that the FSR and CL method can definitely improve the clustering results. Furthermore the SR method showed a high variance of results. This may be the result of using mean vector and its variance only. It should be noted that our method could not yield improvement in all datasets.

6. Conclusion

In this paper, we presented four new methods to combine the results of several FCM clustering algorithms. The primary goal of aggregation is to achieve more appropriate groups than the ones that can be obtained from single run of FCM clustering. In the FCM algorithm, due to random selection of initial fuzzy membership values, different clusters may be achieved in each execution of algorithm on the same data set. Although the FCM method is an unsupervised algorithm, finding the true result is not easy and we need to find a way to determine the accuracy of each FCM analyzer. The average and variance of data point in each cluster and the variance and average of their membership values can help us determine the accuracy of FCM analyzers. Considering the level of accuracy in each FCM, the combination algorithm can obtain the final results. A preliminary experiment revealed that ranking at the level of clustering methods can sometimes lead to incorrect final answers. Therefore, the last proposed method is based on ranking each individual clusters instead of whole clustering method. Each cluster is ranked based on several criterions, *e.g.*, average and variance of data distance from cluster centers and average and variance of data membership values in clusters. Clusters with higher rank affect the final answer more than others. Experimental results over synthetic and UCI data set shows the performance of these four methods.

References

[1] L. D. Alahakoon, S. K. Halgamuge and B. Srinivasan, "Dynamic self-organizing maps with controlled growth for knowledge discovery", Neural Networks, IEEE Transactions on, vol. 11, **(2000)**, pp. 601-614.

[2] E. Alhoniemi, J. Hollmén, O. Simula and J. Vesanto, "Process monitoring and modeling using the self-organizing map," Integr Comput Aided Eng, vol. 6, **(1999)**, pp. 3-14.

[3] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, **(2003)**, pp. 1713-1723.

[4] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," in Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, **(2000)**, pp. 407-416.

[5] D. A. Bell, J. Guan and Y. Bi, "On combining classifier mass functions for text categorization," Knowledge and Data Engineering, IEEE Transactions on, vol. 17, **(2005)**, pp. 1307-1319.

[6] J. Bezdek, "Pattern recognition with fuzzy objective function algorithms", NewYork: Plenum Press, **(1981)**.

[7] D. R. Chen, R. F. Chang and Y. L. Huang, "Breast cancer diagnosis using self-organizing map for sonography", Ultrasound in Medicine and Biology, vol. 26, **(2000)**, pp. 405-412.

[8] I. S. Dhillon, S. Mallela and R. Kumar, "A divisive information theoretic feature clustering algorithm for text classification", The Journal of Machine Learning Research, vol. 3, **(2003)**, pp. 1265-1287.

[9] K. Fukunaga, "Introduction to statistical pattern classification", Academic Press, San Diego, California, USA, vol. 1, **(1990)**, pp. 2.

[10] E. Gose, R. Johnsonbaugh and S. Jost, "Pattern recognition and image analysis", Prentice-Hall, Inc., **(1996)**.

[11] T. Honkela, V. Pulkki and T. Kohonen, "Contextual relations of words in Grimm tales analyzed by self-organizing map", in Proceedings of ICANN-95, international conference on artificial neural networks, **(1995)**, pp. 3-7.

[12] T. M. Khoshgoftaar, S. Zhong and V. Joshi, "Noise elimination with ensemble-classifier filtering for software quality estimation", Intelligent Data Analysis: An International Journal, vol. 9, **(2005)**, pp. 3-27.

[13] T. Kohonen, "The self-organizing map", Proceedings of the IEEE, vol. 78, **(1990)**, pp. 1464-1480.

[14]T. Kohonen, E. Oja, O. Simula, A. Visa and J. Kangas, "Engineering applications of the self-organizing map", Proceedings of the IEEE, vol. 84, **(1996)**, pp. 1358-1384.

[15] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences", Bioinformatics, vol. 22, **(2006)**, pp. 1658-1659.

[16] C. D. Manning and H. Schütze, "Foundations of statistical natural language processing", MIT press, **(1999)**.

[17] R. T. Ng and J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining", in Proc. of 20th Int. Conf. on Very Large DataBases, **(1994)**, pp. 144-155.

[18] S. Saha, C. Murthy and S. K. Pal, "Rough set Based Ensemble Classifier forWeb Page Classification", Fundamenta Informaticae, vol. 76, **(2007)**, pp. 171-187.

[19] P. Zhang, T. D. Bui and C. Y. Suen, "A novel cascade ensemble classifier system with a high recognition performance on handwritten digits", Pattern recognition, vol. 40, **(2007)**, pp. 3415-3429.

Author

Hadi Sadoghi Yazdi is currently an Associate Professor of Computer Science and Engineering at Ferdowsi University of Mashhad (FUM). He received his B.S. degree in Electrical Engineering from FUM in 1994, and received his M.S. and Ph.D. degrees in Electrical Engineering from Tarbiat Modares University in 1996 and 2005, respectively. Dr. Sadoghi Yazdi has received several awards including Outstanding Faculty Award and Best System Design Award in 2007. His research interests are in the areas of Pattern Recognition, Machine Learning, Machine Vision, Signal Processing, Data Mining and Optimization.