A New Clustering Ensemble Framework by Employing Modified Clustering Algorithm Based on Swarm Intelligence

Hamid Parvin
Young Researchers Club, Nourabad Mamasani Branch, Islamic Azad University, Iran
Iran
h.parvin.iust@gmail.com

Hojjatollah Ahangarikiasari
School of Computer Engineering, University of Hertfordshire, UK
United Kingdom
m.mirnabi@hotmail.com

Hamid Alinejad-Rokny
The University of Newcastle, Australia
Australia
h.parvin.iust@gmail.com

ABSTRACT: *Ensemble-based learning is a trustworthy option to reach a strong partition. In order to making up for the faults of each other, the classifiers in the ensemble can do the classification task more reliable than each of them. There is a straight way to induce a set of primary partitions that vary from each other, and then to gather the partitions through a gratifying functions to induce the final partition. Another option in the ensemble learning is to turn to fusion of different information from genuinely various sources. This article introduces a new clustering ensemble learning based on the Ant Colony clustering algorithm. Ensemble needs variability and swarm which is involved in randomness. Various runnings of ant colony clustering cause a number of diverse partitions. Considering these consequences as a new space datasets we make a final clustering by a simple partitioning algorithm to gather them in a gratifying partition. Experimental consequences on some real-world datasets are shown to present the effectiveness of the proposed method in inducing the final partition.*

Keywords: Ant Colony, Data Fusion, Diverse Partitions, Ensemble Framework, Swarm Intelligence, Clustering, Learning

Received: 11 June 2017, Revised 18 July 2017, Accepted 27 July 2017

1. Introduction

Data clustering or uncontrolled learning is an important and challenging problem. The purpose of clustering is to partition a set of unlabeled objects into congruous clusters [6]. There are many applications using clustering techniques to ascertain structures in data, such as data mining [6], information recapture, image segmentation, and machine learning [19]. In real-world problems, the clusters can be shown with various shapes, sizes, data sparseness, and degrees of separation. Clustering techniques need

the explanation of a likelihood measure between patterns. While lack of any prior knowledge about cluster shapes, selecting a specific clustering method are not easy [7]. Selecting a single clustering algorithm and suitable parameters for a given dataset needs both clustering expertise and insights about dataset itself. Instead of running the risk of choosing an inappropriate clustering algorithm, we can think of the different options available by applying all of them to dataset and then combining their clustering consequences. This is the basic thought behind cluster ensembles [4, 5, 8, 22, 24, 26, 28, 29].

Last few years studies have tended to combinational methods. Cluster ensemble methods attempt to find a better and more strengthening clustering solution by fusing data from several primary data partitionings [3, 6, 23, 25, 27]. Ensemble learning roots from cleverness of crowd in the society and humanity fields.

In a successful classifier ensemble the diversity between classifiers of the ensemble is necessary and inevitable. The diversity concept is also necessary for partitions to construct a successful ensemble. To ensure that a number of partitions are various one can use the same strategies as for making classifier ensembles that include [11, 8, 24]:

1. Different Subsets of Features: Each partition available in the ensemble is caught by applying a clustering algorithm on a various projection of dataset.

2. Different Clustering Algorithms: Each partition available in the ensemble is caught utilizing a various clustering algorithm.

3. Randomizing: Each partition existing in the ensemble is caught by applying a clustering algorithm with various initialization. Note that some clustering algorithms are inherently sensitive to its various initialization.

4. Different Datasets: Each partition existing in the ensemble is obtained applying clustering algorithm on a different resampled sub-dataset of original dataset.

There is a lot of gratifying function types that solve this problem heuristically. Most of them need the number of clusters to be specified a priori, but in practice the number of clusters is usually unknown. A new gratifying function is proposed for cluster ensemble based on swarm intelligence [10, 21] that point at this problem. In particular, given a set of partitions, we apply ant clustering to the co-association matrix computed from the ensemble to generate the final partition, and automatically determine the number of clusters.

The first ant colony clustering model was introduced by Deneubourg et al [17]. His model includes the swarm intelligence of real ants, and was embeded into a robot for the object collecting task. Lumer and Faieta [18] improved upon Deneubourg's model by adding the Euclidean distance formula to the similarity density function and giving ants three kinds of abilities: speed, short-term memory, and behavior exchange.

It is aroused by how ants organize their food in their nests. Ant clustering typically involves two key operations: picking up an object from a cluster and dropping it off into another cluster [13. At each step, some ants do pick-up and drop-off based on some notions of likelihood between an object and the clusters. Azimi et al. define a likelihood measure based on the co-association matrix [12]. Their clustering process is completely decentralized and self-organized, allowing the clustering structure to appear automatically from the data consequently, we can accurately determine the number of clusters in the data. The experimental consequences show that the proposed gratifying function is very effective in forecasting the number of clusters and also reach reliable clustering performance. In addition, by introducing some simple heuristics, we can discover the marginal and outlier samples in the data to improve our final clustering.

Liu et al. propose a method for incrementally building a knowledge model for a dynamically changing database, using an ant colony clustering. They use information-theoretic metrics to overcome some inherent problems of ant-based clustering. Entropy controls the pick-up and drop behaviors, while movement is guided by pheromones. They show that dynamic clustering can bring specific advantages over static clustering for a realistic problem scenario [16].

This article is trying to bring two successful concepts in the field of clustering: (a) ensemble concept and (b) swarm concept. Indeed ensemble requires variability vitally and swarm is inherently involved in randomness. We try to turn its randomness to the required variability of clustering ensemble. Indeed various runnings of ant colony clustering cause a number of diverse

partitions that the variability is caught by mentioned randomizing method. Considering these consequences as a new space datasets we apply a final clustering by a simple partitioning algorithm to gather them in a gratifying partition.

The rest of the article is organized as follows. Section 2 includes ant colony clustering. The proposed new space is demonstrated in Section 3. In Section 4, experimental consequences of the clustering algorithm over original feature space versus mapped feature space are compared. The article is concluded in Section 5.

2. Ant Colony Clustering

General form of ant colony clustering algorithm is demonstrated here. The algorithm includes a number of ants. Each ant is performing as an autonomous executor that reorganizes data patterns during exploration to reach an optimal clustering. Pseudo code of ant colony clustering algorithm is shown in Figure 1.

```
initializing parameter;
for each ant a
        place random a in a position not occupied by other ants;
end;
for each object o
        place random o in a position not occupied by other objects;
end;
for t=1:t_max
        for each ant a
                g=select a random number uniformly from range [0,1];
                r=position(a)
                if(loaded(a) and (is_empty(r)))
                        if(g<p_drop)
                                o=drop(a);
                                put(r,o);
                                save(o,r,q);
                        end;
                end;
                elseif(not (loaded(a) or (is_empty(r))))
                        if(g<p_pic)
                                o=remove(r);
                                pick_up(a,o);
                                search_and_jump(a,o);
                        end;    end;
                else
                        wander(a,v,N_dir);
                end;    end;    end;
```

Figure 1. Pseudo code of original ant colony clustering algorithm

At the first step each object shown by a multi-dimensional vector in the original feature space is randomly scattered in a two-dimensional space. In each step each ant randomly searches the space. They use its short-term memory to jump into a place that is potentially near to an object. They can pick up or drop an object using a probability density obtained by equation 1.

$$f(o_i) = \max\left\{0, \frac{1}{s^2} \sum_{o_j \in Neigh_{s\times s}(r)} \left[1 - \frac{d(o_i, o_j)}{\alpha(1 + \frac{v-1}{v_{max}})}\right]\right\} \quad (1)$$

$Neigh_{s\times s}(r)$ is the apparent local area (or the set of apparent rooms) for an ant placed at room r. $Neigh_{s\times s}(r)$ must be near to the location r. It is necessary to mention that each room including $Neigh_{s\times s}(r)$ and r is a two-dimensional vector. The function $d(o_i,o_j)$ is the distance between two objects o_i and o_j in the original feature space. It is computed by equation 2. Threshold α is a parameter that measures the distance between each pair of objects and speed parameter v control the volume of feature space that an ant searches in each epoch of algorithm.

$$d(o_i, o_j) = \sqrt{\sum_{k=1}^{m} (o_{ik} - o_{jk})^2} \quad (2)$$

Where m is the number of original features and where o_{ik} is k-th feature of object o_i. Probability that an unloaded ant takes an object that is in the room occupied by the ant, obtained from the equation 3.

$$P_{pick}(o_i) = \left(\frac{k_1}{k_1 + f(o_i)}\right)^2 \quad (3)$$

k_1 is a fixed threshold to control the probability of picking an object. The probability that a loaded ant lays down its object is obtained by equation 4.

$$P_{drop}(o_i) = \begin{cases} 2f(o_i) & \text{if } f(o_i) < k_2 \\ 1 & \text{if } f(o_i) \geq k_2 \end{cases} \quad (4)$$

k_2 is a fixed threshold to control the probability of dropping an object. Likelihood measure, speed parameter, local density and short-term memory are described in following.

2.1 Perception Area

Number of data objects observed by an ant in a two-dimensional area s. It is considered as one of the effective factors supervising the overall likelihood measure and consequently the accuracy and the calculation time of the algorithm. If s is large, it will result in the rapid formation of clusters and therefore generally less developed clusters. If s is small, it will result in the slower formation of clusters and therefore the number of clusters will be larger. It is necessary to mention that choosing this parameter is a very important factor. While choosing a large value can result in premature convergence of the algorithm, choosing a small value also result in late convergence of the algorithm.

2.2 Similarity Scaling Factor

Scaling parameter value α is defined in the interval (0, 1]. If α is large, then the similarities between objects will increase, so it is easier for the ants to lay down their objects and more difficult for them to lift the objects. So if α is large, fewer clusters are built and it will be highly likely that well-ordered clusters will not form. If α is small, the similarities between objects will decrease, so it is easier for the ants to pick up objects and more difficult for them to lay down their objects. So many clusters that can be well-shaped are created. On this basis, the appropriate setting of parameter α is very important and should not be information independent.

2.3 Speed Parameter

Speed parameter v can uniformly be selected form range $[1,v_{max}]$. Rate of removing an object or picking an object up can be influenced by the speed parameter. If v is large, few rough clusters can irregularly be formed on a large scale view. If v is small,

then many dense clusters can precisely be shaped on a small scale view. The speed parameter is a critical factor for the speed of convergence. A suitable setting of speed parameter v may cause faster convergence.

2.4 Short Term Memory

Each ant can remember the original real features and the virtual defined two-dimensional features of the last q objects it drops. Whenever ant takes an object it will search its short term memory to find out which object in the short term memory is alike the current object. If an object in memory is similar enough to satisfy a threshold, it will jump to the position of the object, hoping the present object will be dropped near the place of the similar object, else if there is no object in memory similar, it will not jump and will hold the object and will roam. This avoids the objects originally belonging to a same cluster to be spitted in different clusters.

2.5 Drawbacks of Original Ant Colony Clustering Algorithm

The original ant colony clustering algorithm demonstrated above suffers two major drawbacks. First many clusters are produced in the virtual two-dimensional space and it is hard and very time-consuming to blend them and this work is in suitable.

The second drawback arises where the density detector is the sole scale based on that the clusters are formed in the local similar objects. But it fails to find their dissimilarity properly. So a cluster without a significant between-object variance may not break into some smaller clusters. It may cause forming the wrong big clusters including some real smaller clusters provided the boundary objects of the smaller clusters are similar. It is because the probability of dropping or picking up an object is dependent only to density. So provided that the boundary objects of the smaller clusters are alike, they placed near to each other and other objects also place near to them gradually. Finally those small clusters form a big cluster, and there is no mechanism to break it into smaller clusters. So there are some changes on the original algorithm to handle the mention drawbacks.

2.6 Entropy Measure of Local Area

Merging the information entropy and the mean likelihood as a new metric to existing models in order to discover rough areas of spatial clusters, dense clusters and troubled borders of the clusters that are wrongly merged is employed.

Shannon entropy data has been widely used in many areas to measure the uncertainty of a specified event or the impurity of a casual collection of samples. Consider a discrete random variable X, with XN possible values $\{x_1, x_2..., x_N\}$ with probabilities $\{p(x_1), p(x_2)..., p(x_N)\}$. Entropy of discrete random variable X is obtained using equation 5.

$$H(X) = -\sum_{i=1}^{N} p(x_i) \log p(x_i) \tag{5}$$

Similarity degree between each pair of objects can be shown as a probability that the two belong to the same cluster. Based on Shannon information entropy, each ant can calculate the impurity of the objects observed in a local area L to determine if the object o_i in the center of the local area L has a high entropy value with group of object o_j in the local area L. Each ant can calculate the local area entropy using equation 6.

$$E(L \mid o_i) = -\sum_{o_j \in Neigh_{s\times s}(r)} p_{i,j} \times \frac{\log_2(p_{i,j})}{\log_2 \left| Neigh_{s\times s}(r) \right|} \tag{6}$$

Where the probability $p_{i,j}$ shows that we have a decisive idea about central object o_i considering a local area object o_j in its local area L. The probability $p_{i,j}$ is obtained according to equation 7.

$$p_{i,j} = \frac{2 \times \left| D(o_i, o_j) \right|}{n} \tag{7}$$

Where n ($n=|Neigh_{s\times s}(r)|$) is the number of neighbors. Distance function $D(o_i,o_j)$ between each pair of objects is measured according to equation 8.

$$D(o_i, o_j) = \frac{d(o_i, o_j)}{norm(o_i)} - 0.5 \tag{8}$$

Where $d(o_i,o_j)$ is Euclidian distance defined by equation 2, and $norm(o_i)$ is defined as maximum distance of object o_i with its neighbors. It is calculated according to equation 9.

$$norm(o_i) = \max_{o_j \in Neigh_{s \times s}(r)} d(o_i, o_j) \tag{9}$$

Now the function $H(L|o_i)$ is defined as equation 10.

$$H(L|o_i) = 1 - E(L|o_i) \tag{10}$$

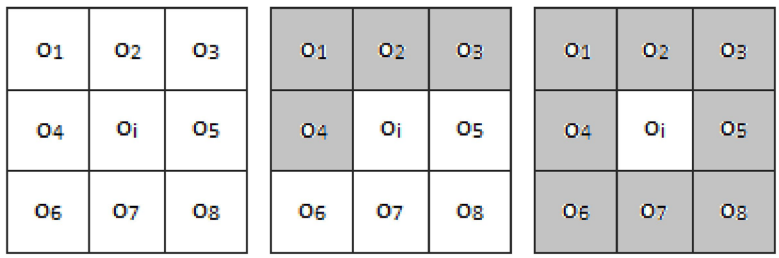

Figure 2. Three examples of local area objects

Three examples of local area objects on a 3×3 (=9) neighborhood depicted in the Figure 2. Different classes with different colors are shown.

When the data objects in the local area L and central object of the local area L exactly belong to a same cluster, i.e. their distances are almost the same and low values, such as the shape or the form depicted by the left rectangle of Figure 2, uncertainty is low and $H(L|o_i)$ is far from one and near to 0. When the data objects in the local area L and central object of the local area L belong to some completely various separate clusters, i.e. their distances are almost the same and high values, such as the shape or the form depicted by the right rectangle of Figure 2, uncertainty is again low and $H(L|o_i)$ is far from one and near to 0. But in the cases of the form depicted by the middle rectangle of Figure 2 where some data objects in the local area L and central object of the local area L exactly belong to a same cluster and some others does not, i.e. the distances are not the same, the uncertainty is high and $H(L|o_i)$ is far from 0 and close to 1. So the function $H(L|o_i)$ can provide ants with a metric that its high value shows the current position is a boundary area and its low value shows the current position is not a boundary area.

In ant-based clustering, two types of pheromone are employed: (a) cluster pheromone and (b) object pheromone. Cluster pheromone guides the loaded ants to valid clusters for a possible successful dropping. Object pheromone guides the unloaded ants to lose object for a possible successful picking-up.

Each loaded ant deposits some cluster pheromone on the current position and positions of its neighbors after a successful dropping of an object to guide other ants for a place to unload their objects. The cluster pheromone intensity deposited in location j, by m ants in the colony at time t is calculated by the equation 11.

$$rc_j(t) = \sum_{a=1}^{m} \left[\mu^{(t - t_a^l)} \times C \times E(L|o_j) \right] \tag{11}$$

Where C is cluster pheromone constant, t^l_a is the time step at that a-th cluster pheromone is dropped at position j, and μ is evaporation coefficient. On other hand, an unloaded ant deposits some object pheromone after a successful picking-up of an

object to guide other agents for a place to take the objects. The object pheromone intensity deposited in location *j*, by *m* ants in the colony at time *t* is calculated by the equation 12.

$$ro_j(t) = \sum_{a=1}^{m} \left[\mu^{(t-t_a^2)} \times O \times H(L \mid o_j) \right] \tag{12}$$

where *O* is object pheromone constant, and t^2_a is the time step at that *a*-th object pheromone is dropped at position *j*. Transmission probabilities of an unloaded ant based on that ant moves from the current location *i* to next location *j* from its neighborhood can be calculated according to equation 13.

$$P_j(t) = \begin{cases} 1/w & \text{if } \sum_{j=1}^{w} ro_j(t) = 0 \forall j \in N_{dir} \\ \dfrac{ro_j(t)}{\sum_{j=1}^{n} ro_j(t)} & \text{otherwise} \end{cases} \tag{13}$$

```
Input:
  QD, itr, q, AntNum, Data, O, C, k1, k2, vmax, period, thr, st, distributions of v, α and μ
initializing parameter using distributions of v, α and μ;
for each ant a
        place random a in a position not occupied by other ants in a plane QD*QD;
end;
for each object o
      place random o in a position not occupied by other objects in the plane QD*QD;
end;
success(1:ant)=0;
failure(1:ant)=0;
for t=1: itr
        for each ant a
                g=select a random number uniformly from range [0,1];
                r=position(a)
                if(loaded(a) and (is_empty(r)))
                        if(g<p_drop)
                                o=drop(a);
                                put(r,o);
                                save(o,r,q);
                        end;
                elseif(not (loaded(a) or (is_empty(r))))
                        if(g<p_pic)
                                o=remove(r);
                                pick_up(a,o);
                                search_and_jump(a,o);
                                success(a)=success(a)+1;
                        else
                                failure(a)=failure(a)+1;
                        end;    end;
                else
                        wander(a,v,N_dir); // considering the defined pheromone
                end;    end;
        if( t mod period==0)
                for each ant a
                        if(success(a)/(failure(a)+success(a))>thr)
                                α(a)=α(a)+st;
                        else
                                α(a)=α(a)-st;
                        end;    end;    end;    end;
```

Figure 3. Pseudo code of modified ant colony clustering algorithm

where N_{dir} is the set of possible w actions (possible w directions to move) from current position i. Transmission probabilities of a loaded ant based on that ant moves from the current location i to next location j from its neighborhood can be calculated according to equation 14.

$$P_j(t) = \begin{cases} 1/w & if \sum_{j=1}^{w} rc_j(t) = 0 \forall j \in N_{dir} \\ \dfrac{rc_j(t)}{\sum_{j=1}^{n} rc_j(t)} & otherwise \end{cases} \tag{14}$$

2.7 Modified Ant Colony Clustering

Merging the information entropy and the mean likelihood as a new metric to existing models in order to discover rough areas of spatial clusters, dense clusters and troubled borders of the clusters that are wrongly merged is employed.

After all the above mentioned modification, the pseudo code of ant colony clustering algorithm is demonstrated in the Figure 3. For showing an exemplary running of the modified ant colony algorithm, take a look at Figure 4. In the Figure 4 the final result of modified ant colony clustering algorithm over Iris dataset is shown.

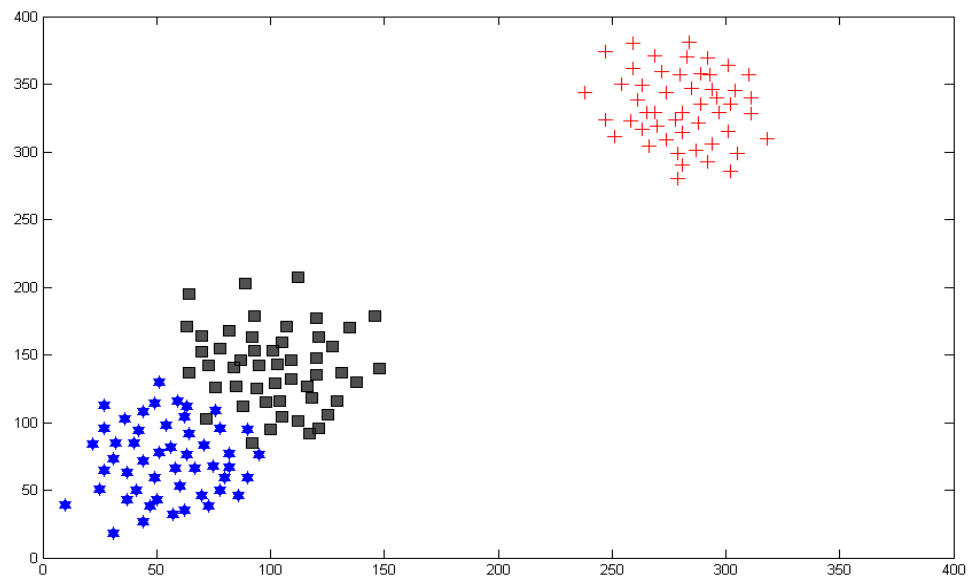

Figure 4. Final result of modified ant colony clustering algorithm over Iris dataset

It is worthy to mention that the quantization degree parameter (QD), queue size parameter (q), ant number parameter ($AntNum$), object pheromone parameter (O), cluster pheromone parameter (C), $k1$ parameter, $k2$ parameter, maximum speed parameter (v_{max}), period parameter, update parameter (thr) evaporation parameter μ and step of update for a parameter (st) are respectively set to 400, 5000000, 20, 240, 1, 1, 0.1, 0.3, 150, 2000, 0.9, 0.95 and 0.01 for reaching the result of Figure 4. Parameter α for each ant is extricated from uniform distribution of range [0.1, 1]. Parameter v for each ant is extricated from uniform distribution of range [1, v_{max}].

Consider that the result shown in the Figure 4 is a well separated running of algorithm. So it is a successful running of algorithm. The algorithm may also converge to a set of overlapping clusters in an unsuccessful running.

3. Proposed New Space Defined by Ant Colony Algorithm

The main idea behind presented method is using ensemble learning in the field of ant colony clustering. In the proposed

algorithm, a multiclass classifier is first trained. Its duty is to obtain confusion matrix over validation set. A set of distinct classifiers is used for each class as an ensemble which is to learn that class. Because of the reinforcement of the main classifier by some ensembles in erroneous regions, it is expected that the accuracy of this method outperforms a simple MLP or un-weighted ensemble [1, 2, 6, 9, 14, 15].

Due to the huge sensitiveness of modified ant colony clustering algorithm to initialization of its parameters, one can use an ensemble approach to overcome the problem of well-tuning of its parameters. The main contribution of the paper is illustrated in the Figure 5.

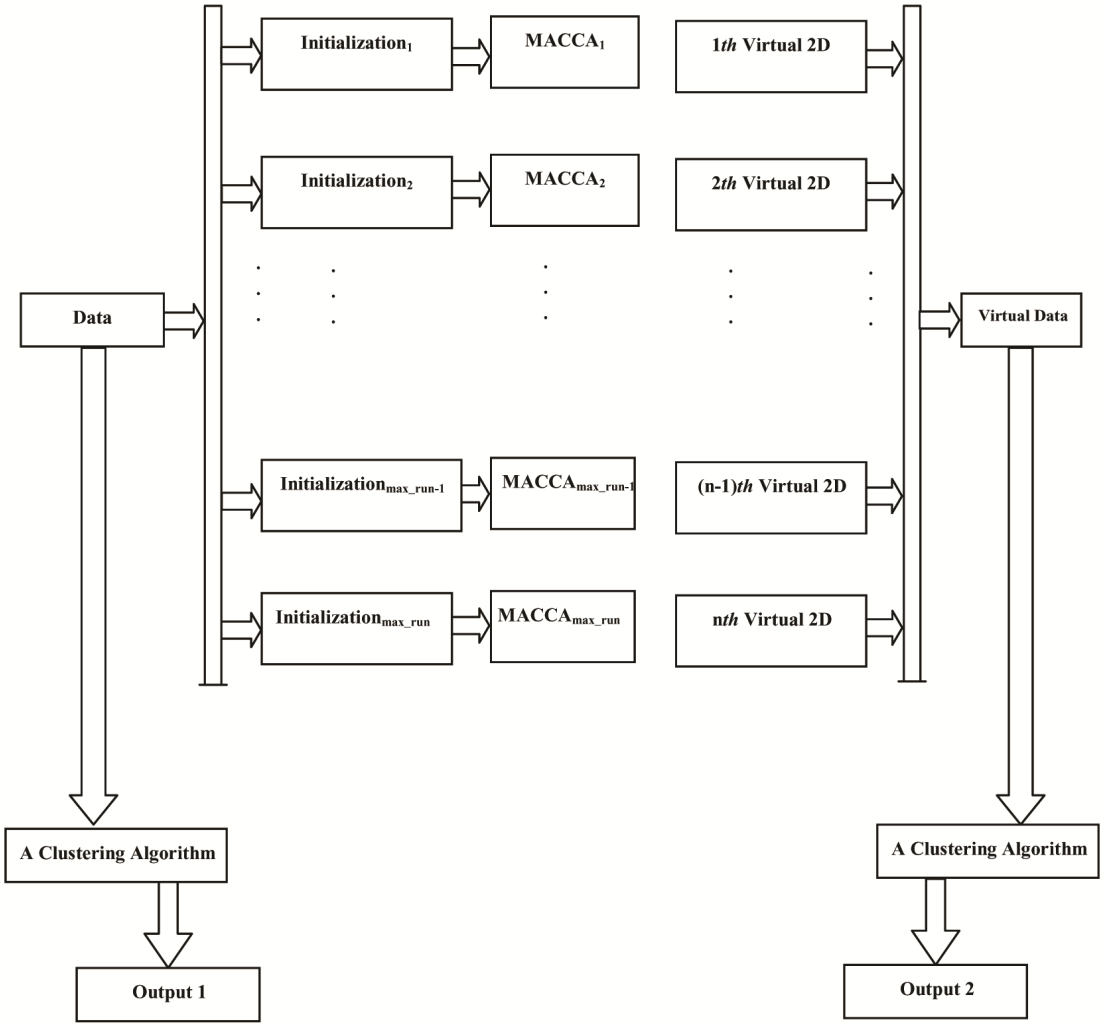

Figure 5. Proposed framework to cluster a dataset using ant colony clustering algorithm

As it is shown in Figure 5 a dataset is feed to as many as *max_run* various modified ant colony clustering algorithms with various initializations. Then we obtain *max_run* virtual 2-dimensions, one per each run modified ant colony clustering algorithm. Then by considering all these virtual 2-dimensions as new space with *2*max_run* dimensions, we reach a new data space. We can apply a clustering algorithm on the new defined data space.

4. Experimental Study

This section evaluates the consequence of applying proposed algorithm on some real datasets available at UCI repository (Newman et al. 1998). The main metric based on which a partition is evaluated is discussed in the first subsection of this section.

The details of the used datasets are given in the following section. Then the settings of experimentations are given. Finally the experimental results are presented.

4.1 Evaluation Metric

After producing the output partition, the most important question is "how good a partition is?". The evaluation of a partition is very important as it is mentioned. Here the normalized mutual data between the achievement partition and real labels of the dataset is considered as the main evaluation metric of the final partition (Strehl and Ghosh, 2002; Alizadeh et al., 2011; Parvin et al, 2011; Parvin et al., 2013).

The normalized mutual information between two partitions, P^a and P^b, is calculated as:

$$NMI(P^a, P^b) = \frac{-2\sum_{i=1}^{k_a}\sum_{j=1}^{k_b} n_{ij}^{ab} \log\left(\frac{n_{ij}^{ab}.n}{n_i^a.n_j^b}\right)}{\sum_{i=1}^{k_a} n_i^a \log\left(\frac{n_i^a}{n}\right) + \sum_{j=1}^{k_b} n_j^b \log\left(\frac{n_j^b}{n}\right)} \tag{16}$$

where n is the total number of samples and n_{ij}^{ab} expresses the number of shared patterns between clusters $C_i^a \in P_a$ and $C_j^b \in P_a^b$; n_i^a is the number of patterns in the cluster i of partition a; also n_j^b are the number of patterns in the cluster j of partition b.

Dataset Name	# of records	# of features	# of classes
Image-Segmentation	210	19	7
Zoo	101	17	7
Thyroid	215	5	3
Soybean	683	35	4
Iris	150	4	3
Wine	178	13	3
SAHeart*	462	9	2
Ionosphere*	351	34	2
Galaxy*	323	4	7

Table 1. Details of used dataset

Another alternative to evaluate a partition is the accuracy metric, provided that the number of clusters and their true assignments are known. To compute the final performance of a clustering algorithm in terms of accuracy, one can first re-label the obtained clusters in such a way that have maximal matching with the ground truth labels and then counting the percentage of the true classified samples. So the error rate can be determined after solving the correspondence problem between the labels of derived and known clusters. The Hungarian algorithm is applied to solve the minimal weight bipartite matching problem. It has been shown that it can efficiently solve this label correspondence problem [20].

4.2 Datasets

The proposed method is examined over 6 different standard datasets. Brief information about the used datasets is available in Table 1. More information is available in [5].

4.3 Experimental Settings

The quantization degree parameter (*QD*), queue size parameter (q), ant number parameter (*AntNum*), object pheromone parameter (O), cluster pheromone parameter (C), *k1* parameter, *k2* parameter, maximum speed parameter (v_{max}), period parameter, update parameter (*thr*) evaporation parameter μ and step of update for α parameter (*st*) are respectively set to 400, 5000000, 20, 240, 1, 1, 0.1, 0.3, 150, 2000, 0.9, 0.95 and 0.01 in all experimentations as before. Parameter a for each ant is extricated from the same distribution of range [0.1, 1]. Parameter v for each ant is extricated from uniform of range [1, v_{max}]. Fuzzy k-means (c-means) is

employed as the base clustering algorithm to perform final clustering over original dataset and new defined dataset. Parameter *max_run* is set to 30 in all experimentations. So the new defined space has 60 virtual features. Number of real cluster in each dataset is given to fuzzy k-means clustering algorithm in all experimentations.

Dataset Name	Fuzzy k-means output 1		Fuzzy k-means output 2	
	Accuracy	Normalized Mutual Information	Accuracy	Normalized Mutual Information
Image-Segmentation	52.27	38.83	54.39	40.28
Zoo	80.08	79.09	81.12	81.24
Thyroid	83.73	50.23	87.94	59.76
Soybean	90.10	69.50	94.34	80.30
Iris	90.11	65.67	93.13	75.22
Wine	74.71	33.12	76.47	35.96
SAHeart*	87.53	60.11	90.12	73.23
Ionosphere*	81.53	57.14	85.34	67.46
Galaxy*	79.42	59.79	82.86	62.74

Table 2. Experimental results in terms of accuracy

As it is inferred from the Table 2, the new defined feature space is better clustered by a base clustering algorithm rather than the original space.

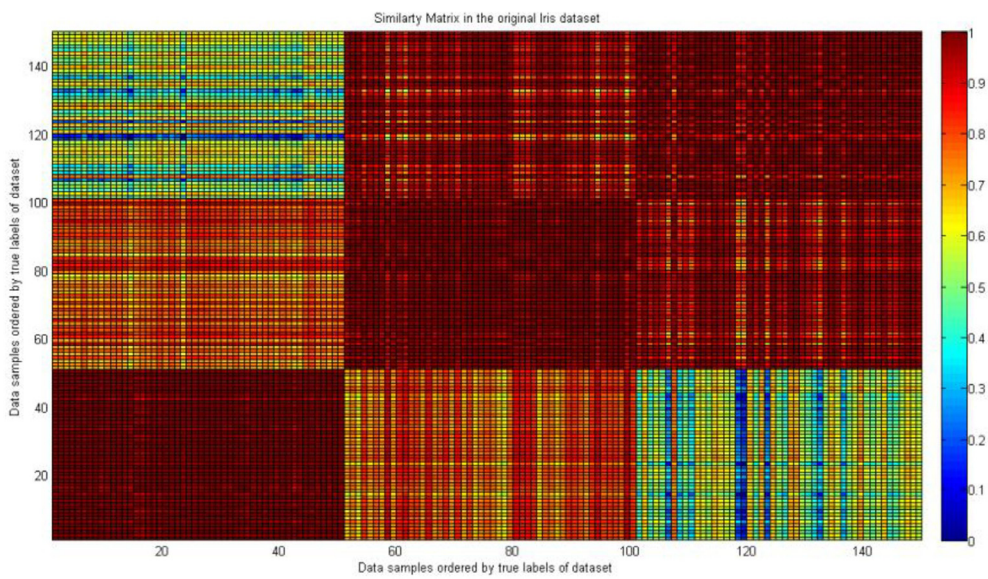

Figure 6. Similarity matrix of data points in original Iris dataset

4.4 Experimental Results

Table 2 shows the performance of the fuzzy clustering in both original and defined spaces in terms of accuracy and normalized mutual information. All experiments are reported over means of 10 independent runs of algorithm. It means that experimentations are done by 10 different independent runs and the final results are averaged and reported in the Table 2.

To demonstrate the effectiveness of the clustering framework, consider Figure 6 and Figure 7. In Figure 6 and Figure 7, a more comprehensive example of similarity matrices of data points in the original and mapped (by ant colony clustering) Iris dataset is presented. Figure 6 shows the similarity matrix of data points in original Iris dataset. It is necessary to mention that the order of data points in each presented similarity matrix is based on real labels of the dataset.

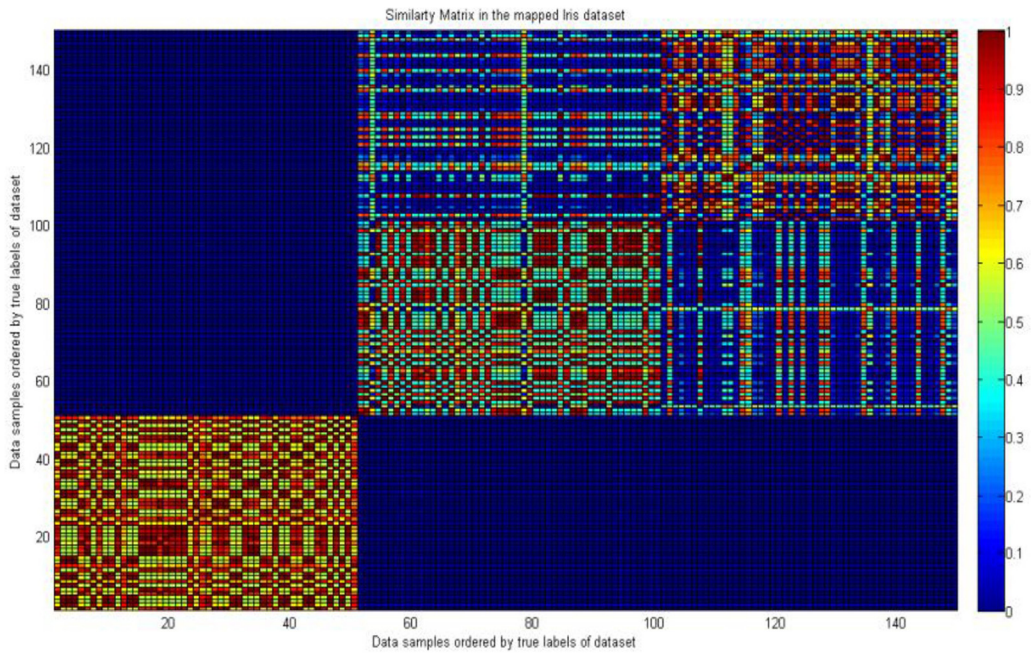

Figure 7. Similarity Matrix of data points in mapped Iris dataset

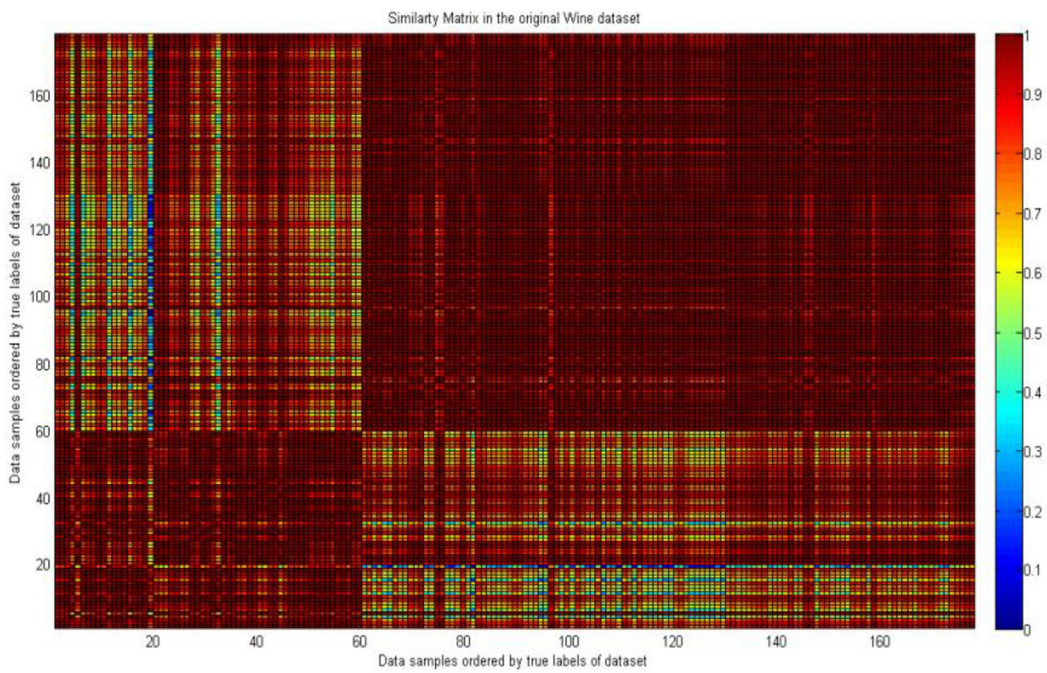

Figure 8. Similarity matrix of data points in original Wine dataset

Figure 7 shows the similarity matrix of data points in the mapped Iris dataset. As it is inferred from Figure 6 and Figure 7 comparatively, the similarity matrix in the mapped Iris dataset is more discriminative than the similarity matrix in the original Iris dataset.

To generalize the conclusion the same experimentation is repeated and demonstrated over Wine dataset. Figure 8 shows the

similarity matrix of data points in original Wine dataset. Figure 9 shows the similarity matrix of data points in the mapped Wine dataset. As it is again obvious from Figure 8 and Figure 9 comparatively, the similarity matrix in the mapped Wine dataset is more different than the similarity matrix in the original Wine dataset.

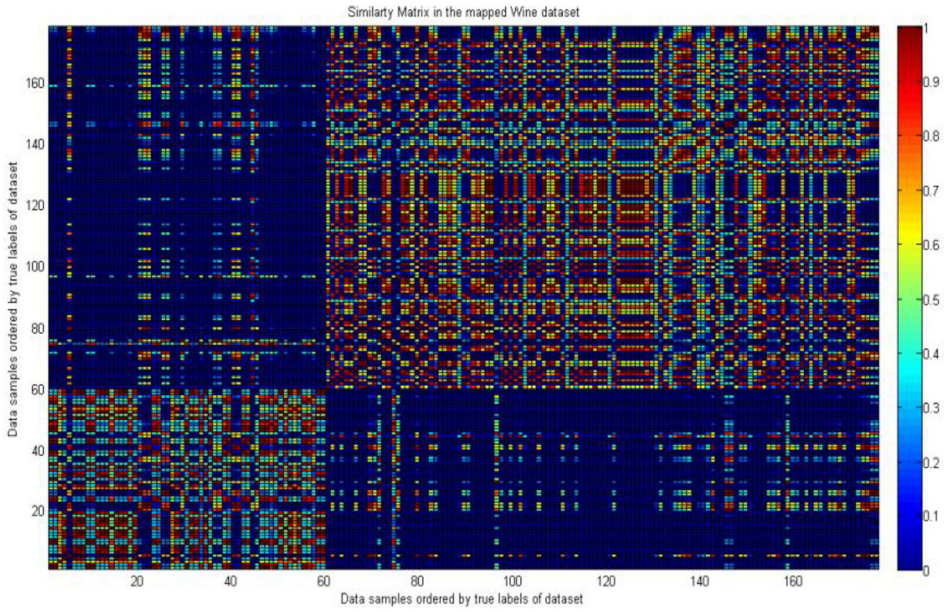

Figure 9. Similarity Matrix of data points in mapped Wine dataset

5. Conclusion and Future Works

This article tried to bring two successful concepts in the field of clustering: (a) ensemble concept and (b) swarm concept. Based on them a new clustering ensemble framework is suggested. Indeed various runnings of ant colony clustering lead to a number of diverse partitions. In the proposed framework we use a type of modified ant colony clustering algorithms and make an intermediate space considering their results totally as a defined virtual space. After producing the virtual space we apply a base clustering algorithm to obtain final partition.

The experiments show that the proposed framework outperforms in comparison with the clustering over original data space. It is concluded that new defined the feature space is better clustered by a base clustering algorithm rather than the original space. As it is shown the similarity matrix in the mapped dataset is more different than the similarity matrix in the original dataset.

For future work the researcher can use a weighting mechanism to participate the result of each ant colony clustering based on its fitness. Also the effect of recessing and elitism mechanisms can be studied. For another future work one can turn to other types of swarm based clustering algorithms.

6. Author Contributions

Conceived and designed the experiments: HP and HA-R. Analyzed the data: HP and HA-R. Contributed reagents/materials/analysis: HP, HA-R and HA. Wrote the paper: HP, HA-R and HA.

References

[1] Parvin, H., Minaei-Bidgoli, B., Alinejad-Rokny, H., Punch, W.F. (2013). Data weighing mechanisms for clustering ensembles. *Computers & Electrical Engineering*, 39 (5) 1433-1450.

[2] Parvin, H., Alinejad-Rokny, H., Parvin, S. (2011). Divide and Conquer Classification. *Australian Journal of Basic & Applied Sciences*, 5 (12) 2446-2452.

[3] Ayad, H.G., Kamel, M.S. (2012). Cumulative Voting Consensus Method for Partitions with a Variable Number of Clusters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30 160-173.

[4] Faceli, K., Marcilio, C.P., Souto, D. (2006). Multi-objective Clustering Ensemble, *In*: Proceedings of the Sixth International Conference on Hybrid Intelligent Systems.

[5] Newman, C.B.D.J., Hettich, S., Merz, C. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/˜mlearn/MLSummary.html

[6] Parvin, H., Minaei-Bidgoli, B., Alinejad-Rokny, H.(2013). A New Imbalanced Learning and Dictions Tree Method for Breast Cancer Diagnosis. *Journal of Bionanoscience*, 7 (6) 673-678.

[7] Roth, V., Lange, T., Braun, M., Buhmann, J. (2002). A Resampling Approach to Cluster Validation, *In*: Intl. Conf. on Computational Statistics, COMPSTAT.

[8] Strehl A., Ghosh J. (2002). Cluster ensembles - knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3 (Dec) 583-617.

[9] Parvin, H., Alinejad-Rokny, H., Asadi, M. (2011). An Ensemble Based Approach for Feature Selection. *Journal of Applied Sciences Research*, 7 (9) 33-43.

[10] Kennedy, J., Russell, S.(2001). Swarm Intelligence. Morgan Kaufmann, San Francisco, .

[11] Kuncheva L.I. (2005). Combining Pattern Classifiers, Methods and. Algorithms, New York: Wiley.

[12] Azimi, J., Cull, P., Fern, X. (2009). Clustering Ensembles Using Ants Algorithm, *In*: IWINAC, LNCS. 295-304.

[13] Tsang, C. H., Kwong, S. (2006). Ant Colony Clustering and Feature Extraction for Anomaly Intrusion Detection, *Studies in Computational Intelligence*, 34. 101-123.

[14] Parvin, H., Helmi, H., Minaie-Bidgoli, B., Alinejad-Rokny, H., Shirgahi, H.(2011). Linkage learning based on differences in local optimums of building blocks with one optima. *International Journal of Physical Sciences*, 6 (14) 3419-3425.

[15] Parvin, H., Alinejad-Rokny, H., Seyedaghaee, N., Parvin, S. (2012). A Heuristic Scalable Classifier Ensemble of Binary Classifier Ensembles. *Journal of Bioinformatics and Intelligent Control*, 1 (2) 163-170.

[16] Liu, B., Pan, J., McKay, R.I. (2006). Incremental Clustering Based on Swarm Intelligence, SEAL, LNCS, 189-196.

[17] Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chretien, L. (1991). The dynamics of collective sorting robot-like ants and ant-like robots. International conference on simulation of adaptive behavior: from animals to animates. MIT Press, Cambridge, MA, 356-363.

[18] Lumer, E.D., Faieta, B. (1994). Diversity and adaptation in populations of clustering ants. International conference on simulation of adaptive behavior: from animals to animates, MIT Press, Cambridge, MA, p. 501-508.

[19] Maheshkumar, S., Gursel S. (2003). Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context. *In*: Int. Conf. on Machine Learning, Models, Technologies and Applications, Las Vegas, Nevadat, USA, CSREA Press, 209-215.

[20] Munkres, J. (1957). Algorithms for the Assignment and Transportation Problems, *Journal of the Society for Industrial and Applied Mathematics,* 5 (1) 32-38.

[21] Van Dongen, S. (2000). Graph Clustering by Flow Simulation, Ph.D. thesis (University of Utrecht, Utrecht, Netherlands.

[22] Frey, B.J., Dueck, D. (2007). Clustering by Passing Messages between Data Points, *Science*, 315, 972-976.

[23] He, D., Wang, Z., Yang, B., Zhou, C.(2013). Genetic Algorithm with Ensemble Learning for Detecting Community Structure in Complex Networks, CoRR .

[24] Strehl, A., Ghosh, J.(2002). Cluster ensembles-a knowledge reuse framework for combining partitionings, *In*: Proc. AAAI Conference on Artificial Intelligence (AAAI'02) California, USA. 93-98.

[25] Parvin, H., Alinejad-Rokny, H., Minaei-Bidgoli, B., Parvin, S. (2013) A new classifier ensemble methodology based on subspace learning. *Journal of Experimental & Theoretical Artificial Intelligence*, 25 (2) 227-250.

[26] Jin, D., Liu, D., Yang, B., Liu, J., Baquero, C. ad He, D. (2011). Ant Colony Optimization with Markov Random Walk for

Clustering in Complex Networks, *In*: The 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2011. Part II, LNAI 6635, 123-134.

[27] Parvin, H., MirnabiBaboli, M., Alinejad-Rokny, H., (2015)Proposing a Classifier Ensemble Framework Based on Classifier Selection and Decision Tree. *Engineering Applications of Artificial Intelligence*, 37. 34-42.

[28] Chensong, P. and Zhong, Li.(2012), K-means Clustering algorithm based on kernel Fisher Discrimination. *INFORMATION*, 15 (11) 4641-4646.

[29] Ahmed, M.T., Abd El, B., Ragab, I.E. (2015). The Beta Generalized Inverted Exponential Distribution. *INFORMATION*, 18 (2) 421-430.