Robust Clustering Using Discriminant Analysis

Vasudha Bhatnagar and Sangeeta Ahuja

[1] Department of Computer Science,
University of Delhi, India
vbhatnagar@cs.du.ac.in
[2] IASRI, New Delhi, India
sangeeta@iasri.res.in

Abstract. Cluster ensemble technique has attracted serious attention in the area of unsupervised learning. It aims at improving robustness and quality of clustering scheme, particularly in scenarios where either randomization or sampling is the part of the clustering algorithm.

In this paper, we address the problem of instability and non robustness in K-means clusterings. These problems arise naturally because of random seed selection by the algorithm, order sensitivity of the algorithm and presence of noise and outliers in data. We propose a cluster ensemble method based on Discriminant Analysis to obtain robust clustering using K-means clusterer. The proposed algorithm operates in three phases. The first phase is preparatory in which multiple clustering schemes generated and the cluster correspondence is obtained. The second phase uses discriminant analysis and constructs a label matrix. In the final stage, consensus partition is generated and noise, if any, is segregated. Experimental analysis using standard public data sets provides strong empirical evidence of the high quality of resultant clustering scheme.

Keywords: K-means, Cluster Ensemble, Discriminant Analysis.

1 Introduction

Obtaining high quality clustering results is a challenging task because of several reasons including randomization inherent in the algorithm [1], sampling (to improve scalability) [2] and idiosyncracies of clustering algorithms [1]. In such situations, different solutions may appear equally acceptable in absence of apriori knowledge of the underlying data distribution [1]. Unfortunately in most real life applications data do not follow *nice* distributions documented in literature. Hence inherent assumptions (idiosyncracies) of the algorithm are often violated producing results that are far from reality, leading to errorenous decisions. Thus the choice of right clustering algorithm, which will reveal natural structures in the data is a difficult task.

Clustering ensemble technique aims to improve the clustering scheme by intelligently combining multiple schemes to yield a robust and stable clustering [1, 3, 4, 5, 2, 6, 7]. The technique has been recognized as an important method of information fusion as it improves robustness, stability and accuracy of the

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 143–157, 2010.

unsupervised learning methods. The technique is naturally amenable for parallelization and application in distributed environment [1]. Combining multiple partitions is the core task in cluster ensemble problem, which is accomplished by design of consensus function Γ.

K-means is one of the most common clustering algorithm used for data analysis in statistics [8], machine learning [9] and data mining [10]. The algorithm, proposed by MacQueen [11], is a center based clustering algorithm which iteratively partitions data till the specified quality criterion (minimum mean squared distances of each data point from centroids) is met. The popularity of the algorithm hinges on its simplicity and efficiency. After more than fifty years of extensive usage for data analysis in the fields of social, physical and biological sciences, it still interests data mining and machine learning community[1]. Bock [8] presents a historical view of K-means algorithm showing the importance and usefulness of the approach.

Interestingly, there are several known limitations of K-means algorithm. Random seed selection ensures that multiple execution of the algorithm on the same data set results into clustering schemes, which may sometimes be significantly different. Consequently user is confronted with the problem of scheme selection, since two different schemes may assign the same object in two different clusters with different properties. Thus there is a possibility of making a wrong decision if the selected scheme does not represent true structures in data. Sensitivity of the algorithm to the order in which the data is presented also contributes to the instability of the algorithm [10]. Presence of noise and outliers in data is a well known and understood cause of non-robustness of K-means clustering algorithm. Since it is not guaranteed to achieve global minimum, the number of iterations for convergence may be very large. Specification of the number of iterations by the user may result into variation of results.

In order to overcome the known weakness causing in stability and consequent non-robustness, a series of extensions and generalizations of K-means algorithm have been proposed [8]. Kanungo et al. [12] propose an effective implementation of K-means which uses a pre-computed kd-tree like structure. Use of this structure avoids reading original data at each iteration, and speeds up the execution. To overcome the effect of random initialization wrapper methods are practiced where the algorithm is executed multiple times and the best clustering is selected. The method has marked computational expense. Bradley et al. [13] propose a refinement scheme for choice of initial seed points. This strategy is particularly useful for large data sets where wrapper approach is infeasible. Since K-means algorithm can identify structures in linear data spaces, kernel K-means has been proposed to identify clusters in non-linearly separable input space [14].

Kuncheva and Vetrov examine the stability of K-means cluster ensemble with respect to random initialization [3]. They give empirical evidence of the stability of the ensemble using both pairwise and non pairwise stability metric. In

[1] On the day of writing this paper, google scholar search for "K-means clustering" yielded more than 76K hits for articles (excluding patents).

other work, Kuncheva and Hadjitodorov propose a variant of the generic pairwise ensemble approach which enforces diversity in the ensemble [15]. A fusion procedure has been proposed in [16] to handle initialization dependency and selection of the number of clusters.

1.1 Problem Definition

Let D denote a data set of N, d-dimensional vectors $x =< x_1, x_2, \ldots, x_d >$, each representing an object. D is subjected to a clustering algorithm which delivers a clustering scheme π consisting of K clusters. ($\pi = \{C_1, C_2, \ldots, C_K\}$). Let $\{\pi_1, \pi_2, \ldots, \pi_H\}$ be H schemes of D obtained by applying either same clustering algorithm on D or by applying H different clustering algorithms. Further, let $\lambda_\pi : D \rightarrow \{1, K\}$ be a function that yields labeling for each of the N objects in D. Let $\{\lambda_1, \lambda_2, \ldots, \lambda_H\}$ be the set of corresponding labelings of D. The problem of cluster ensemble is to derive a consensus function Γ, which combines the H partitions (via labelings) and delivers a clustering π_f, with a promise that π_f is more robust than any of constituent H partitions and *best* captures the natural structures in D. Figure 1 shows the process of construction of cluster ensemble.

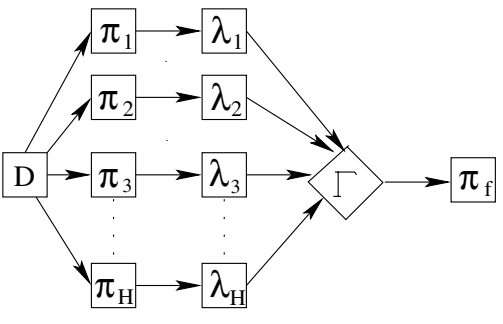

Fig. 1. The process of Cluster Ensemble

Combining the multiple partition is the core task in cluster ensemble problem which is accomplished by design of consensus function Γ. It is the design of Γ that distinguishes different cluster ensemble algorithms to a large extent. Hypergraph partitioning [1], voting approach [7], mutual information [17, 1], co-associations [16, 4, 18] are some of the well established approaches for design of consensus functions.

1.2 Our Approach

We propose to design a K-means cluster ensemble using a well known multivariate statistical analysis technique of Discriminant Analysis (DA). The motivation for using discriminant analysis comes from the ability of the technique to identify observed differences in multivariate data divided into two or more groups. The

identification is carried out by finding 'discriminants' whose numerical values are such that the groups are separated as much as possible [19]. The technique was first introduced by R.A.Fisher and is widely used in statistics for classification in situations where information is either incomplete or expensive [19].

Given H clustering schemes, first the cluster labels are rearranged so as to set correspondance among the clusters. Discriminant function is computed for each scheme and is used to predict the cluster labels of the tuples in D. This process yields NXH label matrix, which essentially consists of predicted labels by each of the partitions for data tuples in D. Based on the user specified threshold, consistent predictions form the part of final clusterings. Tuples with low consistency predictions are iteratively refined for membership, to the best extent possible. If no further refinement is possible they are reported as noise to the user.

Robust Clustering Using Discriminant Analysis (RCDA) algorithm has the following salient features
(i) The algorithm requires two scans of data after clustering.
(ii) Discriminant analysis, a non parametric statistical technique has been utilized for consensus.
(iii) The noisy data is filtered out.
(iv) Experimental analysis of several UCI Machine learning data sets show that the consensus clustering has improved the accuracy, quality, purity, stability and consistency of clustering as compared to the original clusterings.

The paper is organized as follows. Section 2 describes the recent works in the area of cluster ensemble. Section 3 describes the proposed algorithm in detail. Section 4 briefly describes the quality criteria used to evaluate the cluster ensembles. Section 5 describes experimental analysis and finally Section 6 concludes the paper.

2 Related Work

Cluster ensemble technique has been widely studied by machine learning and data mining research community. An informative survey of various cluster ensemble techniques can be found in [1]. We describe some of the well known approaches followed in design of consensus functions, from recent works in cluster ensemble.

In CESG [20], the authors propose a cluster ensemble framework for gene expression analysis to generate high quality and robust clustering results. This clustering has been based upon the concept of distance matrix and weighted graph. In this framework, the clustering results of the individual clustering algorithm are converted into the distance matrix, these distance matrices are combined and a weighted graph is constructed according to the combined matrix. Then a graph partitioning approach is used to cluster the graph to generate the final clusters.

The adaptive clustering ensemble proposed in [2] is inspired by the success of sampling techniques. Clustering is based upon the consistency indices and sampling probability. Individual partitions in the ensemble are sequentially generated

by clustering specially selected subsamples of the given data set. The sampling probability of each data point dynamically depends upon the consistency of its previous assignment in the ensemble.

Probabilistic model of finite mixture of multinomial distribution has been used in [5] for design of consensus function. In [21], authors investigate the commonalities and differences between the categorical clustering and cluster ensemble approaches. They propose a novel algorithm for designing cluster ensemble using concepts from categorical clustering. In ([4],[22]), authors proposed the data resampling approach for building cluster ensembles that are both robust and stable.

In [23], authors give the concept of cluster ensemble based upon multi-clustering fusion algorithm in which different runs of a clustering algorithm are appropriately combined to obtain a partition of the data that is not affected by initialization and overcomes the instabilities of clustering methods. Improvement in the performance of clustering analysis by using Cluster based Similarity Partitioning Algorithm (CSPA), Hypergraph Partitioning Algorithm (HGPA) and Meta Clustering Algorithm (MCLA) cluster ensemble approach has been claimed in [17].

3 Robust Clustering Using Discriminant Analysis

RCDA algorithm operates in three phases. In the first phase, it creates H clustering schemes from data set by applying K-means clustering algorithm as many number of times. Relabeling of the clusters in the partitions is also done during this phase. In the second phase the algorithm constructs discriminant functions corresponding to each partition. This is a compute intensive phase of the algorithm and needs no user parameter. Label of each tuple in D is predicted by each of the H discriminant functions and a NXH label matrix (L) is constructed.

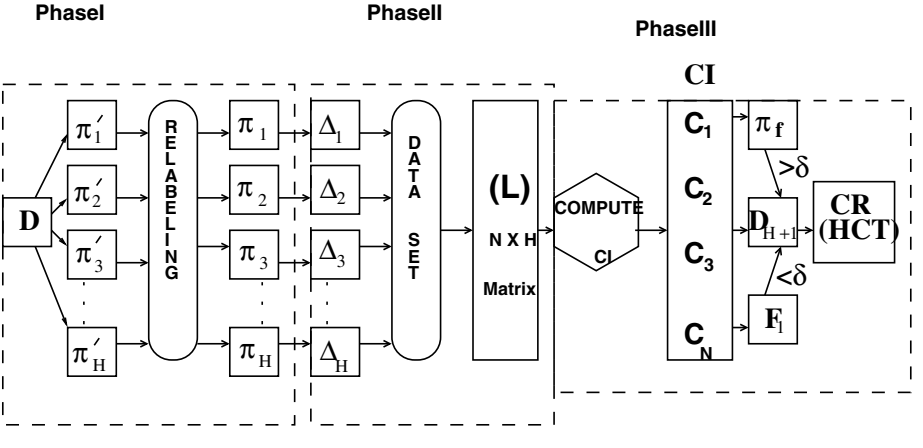

Fig. 2. Architecture of RCDA Algorithm

Finally, in the third phase tuples with consistent labels in L are assigned to clusters in the final partition, and the tuples with low consistency are iteratively refined.

Figure 2 describes the algorithm pictorially. $\pi_1^{'}, \pi_2^{'}, \ldots, \pi_H^{'}$ are the H partitions generated during phase I. Using one partition as the reference partition, relabeling is performed resulting into schemes $\pi_1, \pi_2, \ldots, \pi_H$. Discriminant function Δ_i is constructed for scheme $\pi_i\,(i = 1, \ldots, H)$, to predict the cluster label of each tuple in D. The NXH label matrix L is constructed in which l_{ij} is the label of the cluster in which tuple i falls (a member of), according to π_j. Finally, for each object consistency of prediction is assessed. Tuples which are found to be consistent are used to refine the remaining inconsistent tuples.

3.1 Initialization Phase

Phase I of the algorithm is preparatory in the sense that during this phase H partitions are obtained by as many applications of K-means algorithm on D.

```
    Input  : K :number of cluster, D :data set, H :number of clustering schemes
             (partitions)
    Output: H clustering schemes π1, π2, ..., πH with corresponding labeled clusters
 1  begin
 2      for i = 1 to H do
 3          Apply K-means algorithm on D to deliver partition πi
 4      end
 5      Arbitrarily select πi, (1 = 1, ..., H) as πref
 6      for i = 1 to H do
 7          if (πi <> πref) then
 8              Relabel clusters in πi using distance from centroids.
 9          end
10      end
11  end
```

Algorithm 1. Algorithm for Initialization Phase

Once H partitions have been obtained, the cluster labels need to be coordinated. Since there is no explicit correspondence between the clusters of different partitions, label correspondence problem is solved by taking arbitrarily one partition to be the reference partition π_{ref}. For relabeling partition π_i, the distances of centroids of the clusters in π_i are computed from those in reference partition. The cluster with the centroids closest to those in π_{ref} are assigned labels as in π_{ref}.

This method of relabeling has been chosen because of its efficiency. A more expensive bipartite graph matching based approach has been used in [4, 6] for this purpose. Algorithm 1 shows the steps for phase I.

Let $O(NKt)$ be the complexity of K-means algorithm, K being the number of clusters, t the number of iterations and N the number of tuples in D. The time complexity of phase I is $O(NKtH) + O(K^2)$. The former component is the cost of H runs of K-means algorithm and the latter is the cost of relabeling.

3.2 Predicting Labels Using Discriminant Analysis

Having set the correspondence between the clusters in H partitions, the algorithm proceeds to its core phase. For each partition, a discriminant function is constructed which is used to predict its membership of all records in the data set. One data scan is required for this purpose.

3.2.1 Applying Discriminant Analysis

Discriminant Analysis is a statistical method to separate between distinct classes in multivariate data. It establishes relationships between attributes for classifying objects into one of the several populations, by identifying attributes that best discriminate between the members of the group. In this method, one can judge the maximum discrimination of the tuple to the specific cluster through the discriminant score.

Given p-variate data from K populations $P_1, P_2, \ldots, P_K$ which map to K clusters $(C_1, C_2, \ldots, C_K)$. Cluster C_i contains n_i members, which are similar to each other. Let $X_1, X_2, \ldots, X_p$ denote the p attributes of data objects. The thrust in discriminant analysis is to form a linear function of these variables (Eqn. 1) for each of the K populations.

$$L = \beta_1 * X_1 + \beta_2 * X_2 + \ldots + \beta_p * X_p \tag{1}$$

L is subsequently used to assign a new object to one of the K populations. Computing the discriminant function for each population is the core task in discriminant analysis. This is done by taking into account the variability and correlations between the attributes for each population. We describe the method in detail adapting notation from [19].

Let x_{ijk} denote the value of attribute X_j, for k^{th} object $(1 \leq k \leq n_i)$ of the i^{th} cluster C_i. Thus $< x_{i1k}, x_{i2k}, \ldots, x_{ipk} >$ denotes the attribute vector of the k^{th} object in C_i. In order to determine the discriminant function (L), $\beta' s$ need to be determined in such a way that they provide maximum discriminating capabilities among the clusters. It is important to note that the focus of the estimation is the precision with which the discriminant function correctly classifies sets of observations, rather than the methods for optimization [19, 24].

Let $\overline{x_{ij}} = \frac{1}{n_i} \sum_{k=1}^{n_i} X_{ijk}$ be the observed mean of attribute j for the cluster C_i. Let $\overline{x}_i$ denote the centroid of the cluster C_i. Let $\sum_{k=1}^{n_i} (x_{ijk} - \overline{x_{ij}})^2$ be the squared sum of differences of values of the j^{th} attribute from the mean value for C_i.

Thus $s^i_{jj} = \frac{1}{n_i} \sum_{k=1}^{n_i} (x_{ijk} - \overline{x_{ij}})^2$ is the variance component estimation of the attribute j and $s^i_{jj'} = \frac{1}{n_i} \sum_{k=1}^{n_i} (x_{ijk} - \overline{x_{ij}})(x_{ij'k} - \overline{x_{ij'}})$ is the covariance component where $j \neq j'$ for C_i.

S_i gives the variance-covariance matrix of the cluster C_i.

$$S_i = \begin{pmatrix} s^i_{11} & s^i_{12} & \cdots & s^i_{1p} \\ s^i_{21} & s^i_{22} & \cdots & s^i_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ s^i_{p1} & s^i_{p2} & \cdots & s^i_{pp} \end{pmatrix}$$

```
   Input  : π_1, π_2, ..., π_H
   Output: NXH label matrix L
 1 begin
 2     for j = 1 to H do
 3         for i = 1 to K do
 4             Compute variance covariance matrix S_i
 5         end
 6         Compute pooled variance-covariance matrix Spooled for π_j
 7     end
 8     for x = 1 to N do
 9         for j = 1 to H do
10             for i = 1 to K do
11                 Compute DScore_i(x) for tuple x using discriminant function D_i
12                 l_xj ⟵ label of the cluster with maximum DScore
13             end
14         end
15     end
16 end
```

Algorithm 2. Algorithm for predicting cluster labels using Discriminant Analysis

Define $nm = (n_1 - 1)S_1 + (n_2 - 1)S_2 + \cdots + (n_K - 1)S_K$

$S_{pooled} = \frac{1}{n_1+n_2+\cdots+n_K-K} * nm$ is the pooled variance covariance matrix of the clustering schemes. The D Score of tuple x for the i^{th} cluster of the partition is computed as follows $DScore_i(x) = (\overline{x_i})' \; S_{pooled}^{-1}$ x - $\frac{1}{2}$ $(\overline{x_i})' \; S_{pooled}^{-1} \; \overline{x_i}$ + ln p_i, where p_i is the prior probability of cluster C_i and $\overline{x_i}'$ is the transpose of the centroid using the discriminant functions for each of the H clustering scheme, is class label of each record of D is predicted resulting into label matrix L . Detailed algorithm of this phase is presented in Algorithm 2.

This phase of RCDA is most compute intensive since $(H * K)$ variance-covariance matrices need to be computed. However, this phase does not require any user parameter and is purely data driven.

3.3 Refinement of Clustering Scheme

The label matrix L is processed in the final phase of the algorithm. The user specifies consistency threshold (δ) which is used to segregate consistent and non- consistent tuples. Non-consistent tuples are iteratively refined, till it is not possible to do so. The ones that are left are designated as noisy tuples or outliers. Consistency of the tuple is quantified by using an intuitive method suggested in [2].

Since the cluster correspondence is already established among the H partitions, for a stable clustering algorithm, each tuple must have same cluster label in all the partitions. Thus consistency of the label prediction can be estimated by $CI = Max_{(i=1,K)}(p_i)$, where $p_i = \frac{\text{Number of predictions of label } i}{H}$, $1 \leq i \leq K$ being the cluster label. Thus CI quantifies the maximum confidence with which

Input : NXH label matrix , δ where δ is the consistency threshold
Output: π_f and Noise

```
1  begin
2      for i = 1 to N do
3          Compute consistency score CI
4          if CI ≥ δ
5          Assign tuple to corresponding cluster in π_f
6      end
7      if (K cluster in π_f) then
8          Compute discriminant function D^f from π_f.
9          Predict remaining (inconsistent) tuples using D^f
10         Assign to corresponding cluster in π_f.
11     end
12     else
13         Recompute discriminant functions from
14         the remaining tuples in π_1, π_2, ..., π_H
15         Predict the remaining tuples
16         Compute consistency score of tuple i
17         Assign high score tuples to corresponding cluster in π_f.
18     end
19     Repeat Step 7 to 18 until no change in tuple status.
20     Report remaining tuples as noise.
21 end
```

Algorithm 3. Algorithm for Phase III(Refinement Phase)

cluster label i can be assigned to the tuple. Tuple with consistency above the user specified threshold δ are assigned to the corresponding cluster of the final partition π_f. The tuples that remain are the ones which do not have the desired level of consensus among their labels. If the number is very small and acceptable to the user, these can be discarded (or investigated) as noise, otherwise a refinement step is carried out as described below. If all K clusters are represented in π_f, a new discriminant function D^f is constructed from π_f. The labels of low consistency tuples are predicted by D^f and the tuples are added to the appropriate clusters in π_f.

In case there are outliers in the data, it is possible that all K clusters are not represented in π_f in the first iteration. In such a situation, all the tuples that belong to the missing cluster have low consistency score. Thus there is a need to iteratively improve the cluster quality of π_f. For the remaining tuples in the partitions $(\pi_1, \ldots, \pi_H)$, discriminant functions are recomputed and the tuples are predicted. This process is repeated till the consistency scores of the tuples do not improve beyond the threshold δ. The tuples whose consistency does not improve are reported as noise to the user. Detailed algorithm of this phase is described in the Algorithm 3.

3.4 Discussion

Though RCDA is targeted to overcome the instability of K-means algorithm, the approach is general enough to be applied in other cluster ensemble problems. Initial partitions $\pi_1^{'}, \pi_2^{'}, \ldots, \pi_H^{'}$ can be obtained in multiple ways depending on the environment and application at hand. The H folds of the data can be

created by random sampling without replacement. Each fold may be clustered independently yielding H partitions $\pi_1^{'}, \pi_2^{'}, \ldots, \pi_H^{'}$. In case the data is voluminous ($\geq 100K$ tuples) then in order to achieve scalability H random samples of same size may be drawn from data with replacement, to create $\pi_1^{'}, \pi_2^{'}, \ldots, \pi_H^{'}$.

RCDA algorithm is suitable for data with linearly separable clusters. Discriminant analysis technique captures linear relationship between attributes in a cluster. For non linear groupings in the data Kernel K-means is used [14]. However it is non-trivial to adapt discriminant analysis for this purpose. Further, since discriminant analysis requires computation of variance covariance matrices for computation of discriminant function, its algorithm does not scale well with increasing data dimensionality. The algorithm gives the best results when the number of natural clusters (K) in data is known.

4 Assessing Quality of Ensemble

The cluster ensemble is computationally expensive proposition and hence must deliver reasonable benefit to the user in terms of cluster quality. There is no *best* measure for evaluating the cluster quality. However a mix of internal and external quality criteria can be employed to empirically establish the superiority of the proposed method. We employ the following measures for this purpose as defined in [25].

1. Purity: Purity of a clustering scheme is an external quality criterion and is used when classes in the data are known. A class then corresponds to a cluster, and a cluster with all the objects belonging to one class is considered pure.
   Let there be K clusters in the data set D and size of cluster C_j be $|C_j|$. Let $|C_j|class = i$ denote number of objects of class i assigned to C_j. Purity of C_j is given by

$$Purity(C_j) = Max_{(i=1,K)} \frac{|C_j|class = i}{(|C_j|)} \tag{2}$$

   The overall purity of a clustering solution is expressed as a weighted sum of individual cluster purities

$$Purity = \sum_{j=1,K} \frac{|C_j| * Purity(C_j)}{|D|} \tag{3}$$

   In general, larger value of purity indicates better quality of the solution.
2. Normalized Mutual Information (NMI) : The optimal combined clustering should share the most information with the original clusterings [6, 17]. Normalized Mutual Information (NMI) captures the commonality between two clustering schemes as described below.

Let A and B be the random variables described by the cluster labellings $\lambda(a)$ and $\lambda(b)$ with $k(a)$ and $k(b)$ groups respectively. Let $I(A,B)$ denote the mutual information between A and B, and $H(A), H(B)$ denote the entropy of A and B respectively. It is known that $I(A,B) \leq \frac{H(A)+H(B)}{2}$. Normalized mutual information (NMI)[26] between the two clustering schemes is defined as

$$NMI(A,B) = 2I(A,B)/H(A) + H(B) \tag{4}$$

Naturally $NMI(A,A) = 1$. Eqn 4 is estimated by the labels provided by the clustering. Let $n^{(h)}$ be the number of objects in cluster c_h according to $\lambda(a)$ and let n_g be the number of objects in cluster c_g according to $\lambda(b)$. Let n_h^g be denote the number of objects in cluster c_h according to $\lambda(a)$ as well as cluster c_g according to $\lambda(b)$. The normalized mutual information criteria $\phi(NMI)$ is computed as follows

$$\phi^{(NMI)}(\lambda(a),\lambda(b)) = \frac{2}{n}(\sum_{h=1}^{k^{(a)}}\sum_{g=1}^{k^{(b)}}(n_g^{(h)})\log_{k^{(a)}k^{(b)}}\frac{n_g^{(h)} * n}{n^h * n_g}) \tag{5}$$

3. Adjusted Rand Index (ARI): The Adjusted Rand Index is an external measure of clustering quality which takes into account biases introduced due to distribution sizes and differences in the number of clusters. The quality of clustering $R(U,V)$ can be evaluated by using the Adjusted Rand Index as

$$R(U,V) = \frac{\sum_{(l,k)}(n_{lk}C2) - [\sum_l(n_lC2) * \sum_k(n_kC2)]}{(1/2) * [\sum_l(n_lC2) + \sum_k(n_kC2)] - [\sum_l(n_lC2) * \sum_k(n_kC2)]} \tag{6}$$

where l,k = clusters representation. n_{lk} = number of data items that have been assigned to both cluster l and cluster k. n_l = number of data items that have been assigned to cluster l. n_k = number of data items that have been assigned to cluster k. n = Total number of data items. The Adjusted Rand Index return values in the interval [0,1] and is to be maximized.

5 Experimental Analysis

RCDA (Robust Clustering Using Discriminant Analysis) algorithm was implemented as a multithreaded C++ program and tests for cluster quality were carried out using synthetic, standard UCI machine learning [28] and CLBME repository data sets [29].

Preliminary investigations were carried out on synthetic data generated using ENCLUS data generator [27]. Use of synthetic data allows validating the algorithm. Data set D was generated consisting of 1000 records, distributed in 4 clusters. The cluster sizes were 300, 300, 200, 200 respectively. RCDA was applied on D with $K = 4$ and H varying as 4, 8, 12, 16, 20. $H = 16$ was found to be partition giving best measures (Purity, Mutual Information and Adjusted Rand Index) as computed using Eqns 3 and 6. Results obtained for $H = 16$ are

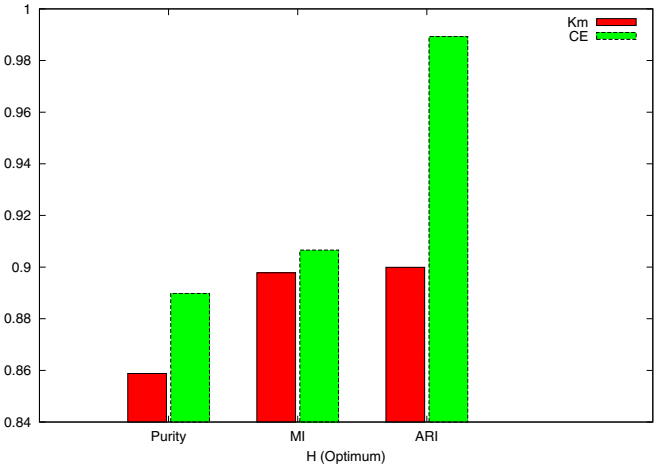

Fig. 3. Comparison of quality measures for synthetic data set. Km is the value of the best metric among all the 16 clustering schemes generated to create the ensemble. CE is the corresponding metric for RCDA ensemble.

Table 1. Details of the data sets; (1) from UCI repository; (2) from CLMBE repository; H: optimum number of partitions

Data Set	**Tuples**	**Dimensions**	**Classes**	H
Wine (1)	178	13	3	4
Winconsion Breast Cancer (1)	683	11	2	8
Respiratory (1)	85	17	2	4
Lymphography (1)	148	18	4	10
Iris (1)	150	4	3	8
Laryngeal (2)	353	16	3	6
Voice3 (2)	238	10	3	4
Voice9 (2)	428	10	9	12

plotted against the corresponding best measures among the sixteen partitions. It was further observed that the best measure values for purity, NMI and ARI come from different partition among the 16 clustering schemes (Figure3).

Five data sets from UCI [28] (Wine, Winconsion Breast cancer, Respiratory, Lymphography, Iris) and 3 data sets from CLBME [29] (Laryngeal, Voice3 and Voice9) were used for evaluation of RCDA algorithm. The characteristics of these data sets are shown in the Table 1. For each of these data sets, experimentation was made by varying the number of partitions in the ensemble and the value of H for which the best combination of purity, NMI and ARI was noted. The value of H that appears in the last column of the Table 1, was used for the evaluating the cluster quality. For each data set, an ensemble partition was constructed using

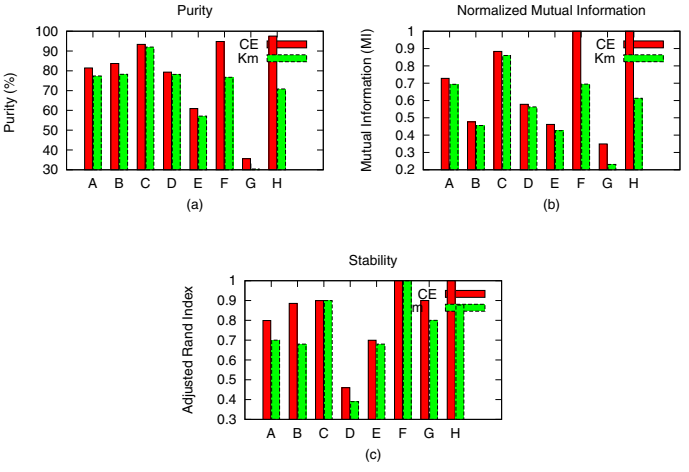

Fig. 4. Comparison of the three quality measures for UCI and CLBME data sets. Km is the value of the best metric among all the H clustering schemes and CE is the corresponding metric for RCDA ensemble. Data sets A:Wine, B:Winconsion Breast Cancer, C:Iris, D:Laryngeal, E:Voice3, F:Lymphography, G:Respiratory and H:Voice9.

the corresponding optimum H value. The three metrics were computed for each of the H partitions individually and the ensemble (π_f). Then for each metric the best value among H clustering schemes was ploted for comparison with RCDA ensemble (Figure 4). It is evident that all three measures are improved in RCDA ensemble for each of the datasets. However the extent of improvement varies for each dataset.

6 Conclusion and Future Work

We propose a novel algorithm Robust Clustering using Discriminant Analysis (RCDA) for designing a cluster ensemble using a well known statistical technique of discriminant analysis. The algorithm aims to overcome the instability of K-means algorithm that arise because of random initialization and data order sensitivity. The motivation for using discriminant analysis arises because of the non-parametric and parameterless nature of the method. The algorithm operates in three phases and requires two scans after the initial clusterings have been done (in phase I). During phase II discriminant functions are computed and cluster labels of all tuples in the data set are predicted. This is the compute intensive phase of the algorithm. In the final phase, the predictions are combined using consistency index and iterative refinement is carried out. The tuples that can not be refined are designated as noise. Preliminary experimentation on synthetic and publically available data sets demonstrates definite improvement in the cluster quality.

References

[1] Reza Ghaemi, M., Nasir Sulaiman, H.I., Mustapha, N.: A survey: Clustering ensembles techniques. In: Proceedings of World academy of science, Engineering and Technology 38, 2070–3740 (2070)
[2] Topchy, A., Behrouz Minaei-Bidgoli, A., Punch, W.F.: Adaptive clustering ensembles. In: ICPR, pp. 272–275 (2004)
[3] Kuncheva, L., et al.: Evaluation of stability of k-means cluster ensembles with respect to random initialization. IEEE Transcations on pattern analysis and machine intelligence 11(28), 1798–1808 (2006)
[4] Fred, A.L.N., Jain, A.K.: Data clustering using evidence accumulation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 835–850 (2002)
[5] Topchy, A., Jain, A.K., Punch, W.F.: A mixture model for clustering ensembles. In: SDM (2004)
[6] Strehl, A., Ghosh, J.: Relationship-based clustering and cluster ensembles for high-dim. data. PhD thesis (May 2002)
[7] Fischer, B., Buhmann, J.M.: Path-based clustering for grouping of smooth curves and texture segmentation. Transaction on Pattern Analysis and Machine Intelligence 25(4) (April 2003)
[8] Bock, H.H.: Origins and extensions of the k-means algorithm in cluster analysis. Electronic Journal for History of Probability and Statistics 4(2) (2008)
[9] Anderson, J., et al.: Machine Learning: An Artificial Intelligence Approach. Morgan Kaufmann, San Francisco (1983)
[10] Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn., Morgan Kaufmann Publishers, San Diego (August 2006)
[11] MacQueen, J.: Some methods for classification and analysis of mulivariate observations (2008)
[12] Tapas, K., et al.: An efficient k-means clustering algorithm:analysis and implementation. CIKM, Mcleen, Virginia, USA, vol. 24(7) (July 2002)
[13] Bradley, P.S., Fayyad, U.M.: Refining initial points for k-means clustering. In: ICML 1998, May 1998, vol. 24, pp. 91–99 (1998)
[14] Dhillon, I.S., Yuqiang Guan, B.K.: Kernel k-means, spectral clustering and normalized cuts. In: KDD, Seattle, Washigton, USA (August 2004)
[15] I, K.L., Hadjitodorov, S.T.: Using diversity in cluster ensembles. In: Proceedings IEEE International Conference on Systems, Man and Cybernatics, The Netherlands, pp. 1214–1219 (2004)
[16] Fred, A.L.N.: Finding consistent cluster in data partitions. MCS 19(9), 309–318 (2001)
[17] Strehl, A., Ghosh, J.: Cluster ensemble knowledge reuse framework for combining partitions (2002)
[18] Topchy, A., Jain, A.K., Punch, W.: Combining multiple weak clusterings. In: Proceedings of the Third IEEE International Conference on Data Mining (2003)
[19] Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice-Hall, Upper Saddle River (August 1979)
[20] Hu, X., Yoo, I.: Cluster ensemble and its applications in gene expression analysis. In: 2nd Asia-pacific Bioinformatics Conference, Dunedin, New Zealand
[21] He, Z., Xiaofei, X., Deng, S.: A cluster ensemble method for clustering categorical data. In: Department of Computer Science and Engineering, Harbin Institute of Technology, China, August, vol. (2), pp. 153–172 (2002)

[22] Minaei-Bidgoli, B., Topchy, A., Punch, W.F.: Ensembles of partitions via data resampling, Michigan State University, East Lancing, MI, USA
[23] Frossyniotis, D., Stafylopatis, M.A.: A multi-clustering fusion algorithm. Journal of Computer Science and Technology 17(2), 118–128 (2002)
[24] Narain, Malhotra, P.: Handbook of statistical genetics. IASRI, New Delhi-12 and Printed at S.C.Printers (1979)
[25] Maimon, O., Rokech, L.: Data Mining and Knowledge discovery Handbook. Springer, Heidelberg (2004)
[26] Ankerst, M., Breuig, M.M., Kriegel, H.P., Sander, J.: Optics: Ordering points to identify the clustering structure. In: ACM SIGMOD 1999 Int. Conf. on Management of Data, Philadelphia, PA (1999)
[27] Chang, C.H., Fu, A.W., Zhang, Y.: Entropy based subspace clustering for mining numerical data. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 1999), San Diego (August 1999)
[28] Uci repository, `http://www.ics.uci.edu`
[29] `http://www.clbme.bas.bg`