CRUDAW: A Novel Fuzzy Technique for Clustering Records Following User Defined Attribute Weights

Md Anisur Rahman and Md Zahidul Islam
Centre for Research in Complex Systems, School of Computing and Mathematics,
Charles Sturt University, Panorama Avenue, Bathurst, NSW 2795,
Australia.

{arahman,zislam}@csu.edu.au

Abstract

We present a novel fuzzy clustering technique called CRUDAW that allows a data miner to assign weights on the attributes of a data set based on their importance (to the data miner) for clustering. The technique uses a novel approach to select initial seeds deterministically (not randomly) using the density of the records of a data set. CRUDAW also selects the initial fuzzy membership degrees deterministically. Moreover, it uses a novel approach for measuring distance considering the user defined weights of the attributes. While measuring the distance between the values of a categorical attribute the technique takes the similarity of the values into consideration instead of considering the distance to be either 0 or 1. Complete algorithm for CRUDAW is presented in the paper. We experimentally compare our technique with a few existing techniques – namely SABC, GFCM, and KL-FCM-GM based on various evaluation criteria called Silhouette coefficient, F-measure, purity and entropy. We also use t-test, confidence interval test and time complexity in evaluating the performance of our technique. Four data sets available from UCI machine learning repository are used in the experiments. Our experimental results indicate that CRUDAW performs significantly better than the existing techniques in producing high quality clusters.

Keywords: Clustering, Fuzzy Clustering, Hard Clustering, Cluster Evaluation, Data Mining.

1 Introduction

Clustering is a process of grouping similar records in a cluster and dissimilar records in different clusters. The records within a cluster are more similar to each other than the records in different clusters (Han and Kamber 2006, Tan et al. 2005). Therefore, clustering extracts hidden patterns (from large data sets) that can help in decision making processes. It has a wide range of applications including social network analysis, DNA analysis, software engineering, crime detection, medical imaging, market segmentation, and search result grouping (Zhao and Zhang 2011, Haung and Pan 2006, Songa and Nicolae 2008, Lung et al. 2003, Grubesic and Murray 2001, Tsai and Chiu 2004, Zamir and Etzioni 1999, Masulli and Schenone 1998). Hence, it is important to produce good quality clusters for supporting decision making processes.

There is always room for further improvements in the existing clustering techniques. For example, a group of techniques select initial seeds randomly (Saha et al. 2010, Hasan et al. 2009, Redmond and Heneghan 2006). Due to the random selection of initial seeds, they end up producing different sets of clusters in different runs resulting in an uncertainty of cluster quality in a run. Good quality of initial seeds is crucial for good quality clusters (Rahman and Islam 2011). Some other clustering techniques require various user inputs including number of clusters which can often be very difficult for a user/data miner to provide (Lee and Pedrycz 2009, Chatzis 2011, Ahmad and Dey 2007a, Saha et al. 2010).

Moreover, most of the existing clustering techniques consider that all attributes of a data set are equally important for clustering. That is, the weights (significance levels) of all attributes of a data set are considered to be equal. In all clustering steps including measuring distance, between a record and a seed, all attributes are used with the same significance/importance level, say 1. The clustering techniques do not allow a data miner/user to assign different significance levels such as 1, 0.8, 0.2 and 0 to different attributes as appropriate/desired. A data miner can either ignore (i.e. assign significance level equal to 0) or consider (i.e. assign significance level equal to 1) an attribute while clustering the records. It would be very useful if a clustering technique could provide a data miner with the flexibility to assign different significance levels (anything between 1.0 and 0.0) to different attributes. This would help a data miner to explore various sets of clusters using different weight arrangements for the attributes of a data set.

In many existing fuzzy clustering techniques initial fuzzy membership degrees of the records are assigned randomly (Lee and Pedrycz 2009, Alata et al. 2008, Bezdek 1981, Hathaway and Bezdek 1988). Hence, a data miner may get different clustering results in different runs of a fuzzy clustering technique resulting in similar problems to the case where initial seeds are selected randomly.

Some techniques are suitable either for data sets having only numerical attributes or for data sets having only categorical attributes (Bai et al. 2011, Li et al. 2008, Guha et al. 1988, Zhang et al. 1996), while in reality data sets often have both numerical and categorical attributes. Although there are techniques that can handle both numerical and categorical attributes (Huang 1997, Ji et al. 2012), some of them do not consider any similarity

Copyright © 2012, Australian Computer Society, Inc. This paper appeared at the 10th Australasian Data Mining Conference (AusDM 2012), Sydney, Australia, December 2012. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 134, Yanchang Zhao, Jiuyong Li, Paul Kennedy, and Peter Christen, Ed. Reproduction for academic, not-for-profit purposes permitted provided this text is included.

between categorical values in the sense that if two categorical values (of an attribute belonging to two records) are different then the distance between the two records in terms of the attribute is considered to be 1 (regardless of the similarity of the values), and otherwise 0.

In this study, we present a novel fuzzy clustering technique called **C**lustering **R**ecords Following **U**ser **D**efined **A**ttribute **W**eights (CRUDAW). The key contributions of our proposed technique are as follows.

In CRUDAW, we use high quality initial seeds obtained through a deterministic process based on the density of the records of a data set. Besides, the number of clusters is automatically defined through the clustering process without requiring a user input on this. Moreover, it allows a user to assign different significance levels (ranging from 0.0 to 1.0) to different attributes and cluster the records accordingly. If the significance of an attribute is advised to be 0 the technique totally ignores the attribute while clustering the records, whereas if the significance of an attribute is something between (0.0, 1.0] CRUDAW considers the influence of the attribute according to its weight. For example, while calculating the distance between two records (as part of the steps of clustering) the technique considers the weights of the attributes, where the influence of an attribute is greater when the weight of the attribute is higher.

Note that CRUDAW offers more options than just the traditional two options; i.e. either consider an attribute or ignore the attribute while clustering records. A data miner may want to cluster people mainly based on career related information, but also may want to give some importance to the demographic information. In that case he/she may want to assign high weights (such as 0.9 and 0.7) on the career related attributes and low weights (such as 0.1 and 0.4) on demography related attributes, and zero weights on all other attributes. If a user chooses to cluster records considering say three attributes with weights 1.0, 0.7, and 0.2, respectively then he/she gets a clustering result that is likely to be different to the clustering result he/she would get if he/she had chosen even the same three attributes with different weights say 0.4, 0.9 and 0.6.

Another interesting property of our technique is that it calculates the distance between two categorical values based on their similarity, instead of considering the distance either 1 (if the values are different) or 0. The distance between two categorical values can therefore be anything between 0.0 and 1.0. Hence, our technique is suitable for data sets having only numerical, only categorical or both numerical and categorical attributes. Additionally, we determine the initial fuzzy membership degree of the records from the initial seeds that are selected deterministically, and thereby avoid the randomness of initial membership degree.

We compare the cluster quality of our proposed technique with a few other top quality exiting techniques called SABC, GFCM, and KL-FCM-GM (Ahmad and Dey 2007a, Lee and Pedrycz 2009, Chatzis 2011). We use four publicly available data sets that are obtained from UCI machine learning repository (UCI 2012). Several commonly used criteria namely Silhouette coefficient, F-measure, entropy, and purity (Chuang 2004, Tan et al. 2005, Kashef and Kamel 2009) are used to evaluate the technique. The experimental results clearly indicate that the quality of clusters produced by CRUDAW is better than the quality of clusters produced by the top class existing techniques.

The structure of the paper is as follows. In Section 2, we discuss some existing clustering techniques. Our novel clustering technique is presented in Section 3. We present experimental results in Section 4 and give concluding remarks in Section 5.

2 Literature Review

In this study, we consider a data set as a two dimensional table (see Table 1) with a number of columns (attributes) and rows (records). Attributes of a data set can be categorical and numerical. In our example data set there are ten records, six categorical attributes (Marital-Status, Qualification, Occupation, Professional-Training, Country-of-Origin, and First-Language), and one numerical attribute (Age). We can group the attributes of the data set into three categories namely demographic, career, and background as shown in Table 1. The domain values for categorical attribute Marital-Status are {Single, Married}. Similarly, the domain values of all other categorical attributes can be learnt from Table 1.

Clustering is a data mining task that groups similar records in a cluster and dissimilar records in different clusters. Similarity of records are typically measured based on their distances. For the purpose of clustering, the distance between two numerical attribute values can be measured based on Euclidian distance since numerical values exhibit a natural ordering among them. For a categorical attribute, the distance between two categorical attribute values are typically considered to be either zero or one. However, it may not be sensible to consider the distance between two categorical attribute values either zero or one. The distance between two categorical values can depend on their similarity (Islam and Brankovic 2011, Rahman and Islam 2011). The similarity between two categorical values are generally measured based on their co-appearance (connection) with the domain values of other categorical attributes among the records of a data set (Giggins 2009, Ganti et al. 1999).

To calculate similarity, a data set is first converted into a graph by considering all categorical attribute values of a data set as vertices of the graph (Giggins 2009). Co-appearances of two attribute values are used for drawing the edges between the vertices representing the values. Let, $S_{p,q}$ be the similarity for categorical attribute values p and q, v be the total number of vertices, a_{pt} be the number of edges between vertices p and t (where t represents the domain value of another categorical attribute), a_{tq}be the number of edges between vertices t and q, and $d(p)$ and $d(q)$ be the degrees of vertices for p and q, respectively. The similarity between two categorical attribute values (p and q) belonging to an attribute can be calculated with respect to another value t belonging to another attribute as follows.

$$S_{p,q} = \frac{\sum_{t=1}^{v}\sqrt{a_{pt} \times a_{tq}}}{\sqrt{d(p) \times d(q)}} \qquad (1)$$

If a data set has both categorical and numerical attributes, we suggest that the numerical attribute values can be first categorized and then the similarity between two categorical attribute values can be calculated based on both categorical and numerical (categorized) attribute values. Similarity of categorical attribute values can be useful in clustering records of a data set having categorical attributes.

For a data set having numerical attributes, K-Means is one of the most widely used clustering techniques. A user first needs to define the desired number of clusters. K-Means then selects as many seeds as the user defined number of clusters where each seed, which is a record, is chosen randomly (Han and Kamber 2006, Tan et al. 2005). Distances between a record and all the seeds are calculated. The record is assigned to the seed with which it has the minimum distance. Each record is assigned to only one seed. Records assigned to the same seed are considered to be a cluster.

	Demographic		Career			Background	
Record	**Age**	**Marital-Status**	**Qualification**	**Occupation**	**Professional-Training**	**Country-of-Origin**	**First-Language**
R_1	65	Married	PhD	Academic	No	Australia	English
R_2	30	Single	Master	Engineer	No	Bangladesh	Non-English
R_3	45	Married	Master	Engineer	No	India	Non-English
R_4	30	Single	Bachelor	Physician	Yes	Australia	English
R_5	55	Married	PhD	Academic	No	Australia	English
R_6	35	Single	Bachelor	Physician	Yes	India	Non-English
R_7	60	Married	PhD	Academic	No	Bangladesh	Non-English
R_8	45	Single	Bachelor	Physician	Yes	Australia	English
R_9	35	Single	Master	Engineer	Yes	India	Non-English
R_{10}	42	Married	Master	Engineer	No	Australia	English

Table 1: An Example Data Set

While calculating the distance between a record and a seed, typically Euclidian distance between two numerical values belonging to an attribute is used. However, for measuring the distance between two categorical values if both records have the same value for the attribute then their difference is considered to be zero, and otherwise it is considered to be one (Huang 1997, Ji et al. 2012).

K-Means then re-calculates the seeds based on the records belonging to each cluster. Generally a seed is calculated by taking the average value of a numerical attribute, and the mode value of a categorical attribute among all records belonging to a cluster. After new seeds are selected all records are again reorganized in such a way that a record is assigned to the cluster the seed of which has the minimum distance with the record. The process of reorganizing records and finding new seeds continues recursively until a termination condition is satisfied. Typically, a user defined number of iteration and/or a minimum difference between the seeds are considered as termination conditions.

Another existing technique (Ahmad and Dey 2007a) uses fuzzy seeds while performing clustering through a modified version of K-Means. We call the technique as SABC throughout the paper. SABC first randomly distributes the records of a data set among a user defined number of clusters. It then determines the center (seed) of each cluster in a way so that instead of having a single value of a categorical attribute, the seed contains all categorical values of an attribute proportionate to their frequencies within the records belonging to the cluster.

The distances between a record and all seeds are then calculated. In order to compute the distance between a record and a seed SABC calculates the distance between them for each attribute; both categorical and numerical. Distance between two categorical values is calculated with respect to their co-appearance with values of another attribute (Ahmad and Dey, 2007b). Another interesting property of SABC is that it automatically (not user defined) computes the significance of each numerical attribute which is then used the distance calculation function.

A record is then assigned to the seed with which it has the minimum distance among all seeds. After the allocation of all records to their nearest seeds SABC moves to the next iteration where it calculates a new set of seeds as before based on the new arrangement of the records. The process of record re-allocation and seed selection continues until a maximum number of iteration is complete or the clusters stabilise.

All clustering techniques discussed so far are hard clustering where a record belongs to only one cluster. There is another type of clustering called fuzzy clustering where a record has some attachment/relationship with all clusters, instead of just with one cluster. Fuzzy C-Means (FCM) is one of the most commonly used fuzzy clustering techniques, which explores such fuzziness nature of the records (Bezdek 1981, Huang and Ng 1999). For each record, FCM assigns a fuzzy membership degree for the record and a cluster in order to represent the level of attachment between the record and the cluster.

Some early FCM techniques can handle a data set having only numerical attributes. However, there are FCM techniques such as General Fuzzy C-Means (GFCM) that can handle data sets having both numerical and categorical attributes (Lee and Pedrycz 2009). General Fuzzy C-Means (GFCM) uses the following clustering steps.

1. Takes a user defined number of clusters.
2. Randomly assigns a fuzzy membership degree for a record and a cluster; for all records and all

clusters. Let, $\mu_{i,j}$ be the fuzzy membership degree of the ith record with the jth cluster. GFCM chooses initial fuzzy membership degrees in such a way so that $\sum_{j=1}^{k} \mu_{ij}=1; \forall i$, where k is the total number of clusters.

3. Using the fuzzy membership degrees of the records, cluster centers are re-calculated as we explain below in the description of the steps.
4. A new set of fuzzy membership degrees is calculated (as explained below) for every record considering the new cluster centers as calculated in Step 3.
5. Repeat Step 3 and Step 4 until a termination condition is met.

We now explain the process of cluster center calculation (Step 3) and fuzzy membership degree calculation (Step 4) in the following paragraphs.

Each attribute value of a center is calculated using the values of the attribute for all records of the data set. The seed value of a numerical attribute is calculated as the same way Fuzzy C-Means does. For a categorical attribute, GFCM uses a fuzzy seed value where the seed contains each value of the domain of the attribute according to a confidence degree (Kim et al. 2004). The confidence degree of an attribute value a_l is the sum of the membership degrees (with the cluster) of all records having a_l.

Once the cluster centres are calculated – based on the distance between a record and a centre, a new set of membership degrees is calculated for the center and the records (Step 4). The fuzzy membership degree of a record and a cluster center (seed) is inversely proportional to the distance between the record and the seed. For numerical attributes, GFCM calculates normal Euclidean distance. However, for categorical attributes it calculates distance based on frequency of attribute values.

GFCM repeats Step 3 and Step 4 until a termination condition is met i.e. either a user defined number of iteration is completed or the objective function is minimized.

Another fuzzy clustering technique called Kull-back-Leibler FCM (KL-FCM-GM) can handle a data set having both categorical and numerical attributes (Chatzis 2011). It is an extension of Gath-Geva algorithm, which is a well-known fuzzy clustering technique that works on a data set having only numerical attributes (Gath and Geva 1989).

It randomly chooses the initial fuzzy membership degree for a record in such a way so that $\sum_{j=1}^{k} \mu_{ij}=1$, where k is the user defined number of clusters and μ_{ij} is the fuzzy membership degree of ith record with jth cluster. It then calculates the weight of each cluster. The weight of each cluster is the summation of the fuzzy membership degrees of the records with that cluster divided by the total number of records of a data set.

Using the weights of the clusters and the distances between a record and the seeds, KL-FCM-GM then re-calculates the fuzzy membership degrees of each record. The membership degree of a record with a cluster is higher than the fuzzy membership degree of the record with another cluster, if the weight of the former cluster is higher than the weight of later cluster, and the distance between the record and the seed of the former cluster is less than the distance between the record and the seed of the later cluster.

KL-FCM-GM next repeats the steps to re-calculate the weights of the clusters and fuzzy membership degrees of the records. The clustering process continues until a user defined number of iterations is reached or the values of the objective function converge.

3 Our Clustering Technique

We present a novel clustering technique called "**C**lustering **R**ecords Following **U**ser **D**efined **A**ttribute **W**eights" (CRUDAW). In this section we first discuss the basic concepts used in our clustering technique. We then introduce different components used in various steps of the clustering technique.

3.1 Basic Concepts

Most of the existing clustering techniques only allow a user (data miner) either to consider or totally ignore an attribute while clustering the records. All attributes that are considered have equal influence in clustering the records. That is, most of the existing techniques do not allow a user to assign different significance levels (weights) to different attributes. All attributes that are considered for clustering have weight equal to 1 and all attributes that are ignored have weight equal to 0. Nevertheless, it can often be crucial for a user to consider a few attributes with high weights, and some other attributes with low weights while ignoring the remaining attributes for clustering. For example, different users may want to cluster records of our example data set (see Table 1) for different purposes such as grouping people according to their similarity based on career and finding groups of similar people based on their background. Therefore, a user may want to cluster the records from different perspectives including career and background related view point.

A user (data miner) wanting to explore the clusters from a career point of view may want to assign high weights on career related attributes, low weights on demographic attributes and zero weights (completely ignore) on background related attributes. An example of a weight distribution can be 0.3, 0.2, 0.8, 0.8, 0.6, 0.0 and 0.0 for the attributes Age, Marital-Status, Qualification, Occupation, Professional-Training, Country of Origin, and First Language, respectively. Note that the attributes Qualification, Occupation, and Professional-Training are considered to be career related attributes and therefore, given high weights 0.8, 0.8 and 0.6. Based on the weight distribution a possible clustering result can be three clusters having {R2, R3, R9, R10}, {R4, R6, R8} and {R1, R5, R7} records, respectively (see Table 1).

However, another user may want to cluster the records mainly based on background information and therefore, assign high weights on background related attributes. In that case, an example of possible weight assignment can be 0.2, 0.3, 0.0, 0.0, 0.0, 0.6, and 0.9 weights on the attributes, respectively. A possible clustering outcome can be a set of three clusters having {R1, R5, R7, R10}, {R2, R3, R6, R9} and {R4, R8} records, respectively. The records are clustered differently for the two users. For the first user a record (say R10) is clustered with a set of records (R2, R3, and R9) whereas the same record R10 is clustered together with a different set of records (R1, R5, and R7) for the second user, due to different weight assignments on the attributes.

Similarly, the clusters can be very different even for the first user if he/she assigns a different weight pattern for the attributes of his interest. For example, the first data miner could also use a different weight pattern 0.2, 0.3, 0.7, 0.8, 0.9, 0.0 and 0.0 for the attributes, respectively. Note that the first user still assigns high weights on career related attributes.

Therefore, following the underlying approach of some previous studies (Rahman and Islam 2011, Islam and Brankovic 2011, Islam 2008, Islam and Brankovic 2005) we propose a novel clustering technique allowing a user to assign different weights on different attributes. We find in the literature another clustering technique called SABC (Ahmad and Dey 2007a) that automatically (not user defined) calculates the weights of the attributes. However, the technique does not allow a user to assign weights according to the requirements of a user.

Moreover, in an initial experiment we do not find the calculated weights (by SABC) to be matching with the actual weights that are calculated according to an entropy analysis as follows. We carry out an experiment on the Breast Cancer data set available from UCI Machine Learning repository (UCI 2012). The data set has a natural class attribute (also called label of a record) to indicate the diagnosis for each patient. We then calculate the entropy, gain and gain ratio (Quinlan 1993, Quinlan 1996, Islam 2012) of each attribute in order to explore their significance levels with respect to the natural class attribute. We also calculate the significance (weight) of an attribute according to SABC. We find that the attributes having high weights according to the entropy analysis (i.e. low entropy, high gain and high gain ratio) do not necessarily have high weights according to SABC calculation.

Attribute Name	Significance (SABC)	Attribute Name	Entropy
inv-nodes	0.414614011	**deg-malig**	0.783292426
age	0.323475993	inv-nodes	0.789404198
menopause	0.232572616	tumor-size	0.810368462
node-caps	0.225719134	node-caps	0.815943182
tumor-size	0.215500036	irradiat	0.837121642
irradiat	0.186524756	*age*	0.851096449
deg-malig	0.157275391	*menopause*	0.860274546
breast-quad	0.14948599	breast-quad	0.863183885
breast	0.095167189	breast	0.870592539

Figure 1: Comparative study on significance of the attributes according to SABC and conventional entropy calculation using Breast Cancer data set

We understand that typically a data set used for clustering does not have a natural class attribute. However, one of the main purposes of clustering is to assign such a label (class value) to an unlabeled record. Therefore, following a reverse engineering approach the use of entropy analysis for finding possible significance levels is used.

In Figure 1, the first two columns show the attributes and their significance values, respectively as calculated by SABC. The third and fourth columns show the attributes and their entropy values, respectively. According to entropy calculation "deg-malig" is the most important attribute (having the least entropy), whereas according to significance calculation the attribute is only the 7th most important attribute. Similarly, according to the significance calculation "age" and "menopause" are the 2nd and 3rd most important attributes, respectively while they are only the 6th and 7th important attributes in terms of entropy (Figure 1).

Unlike many existing techniques (Ahmad and Dey 2007a, Lee and Pedrycz 2009, Chatzis 2011), our proposed technique uses a deterministic process (instead of a random process) in order to identify high quality initial seeds for clustering. High quality initial seeds are very important for a high quality clustering as evidenced in a previous study (Rahman and Islam 2011). However, unlike the previous technique, in this study we take an approach to identify high quality initial seeds with low time complexity.

Moreover, unlike many existing techniques our proposed technique does not require a user to give the number of clusters as an input. We argue that the estimating the number of clusters in advance can be a difficult job for a user. However, the proposed technique instead takes the radius for a seed as an input. Note that the radius is only used to find the initial seeds through a deterministic process. It is not used as the radius of the final clusters.

We identify the initial seeds and thereby initial fuzzy membership degrees in a deterministic way. Moreover, our proposed technique uses a novel approach for distance calculation, between two records, where the attributes having higher significance have higher influence in the distance calculation for two records. The proposed technique also calculates the similarity (anything between 0 and 1) among the values of a categorical attribute and uses the similarity in order to calculate their distance, unlike many existing techniques (Huang 1997, Ji et al. 2012) that consider the distance to be either 1 (if the values are different) or 0 (otherwise). Thus, our technique can handle better the data sets having numerical and/or categorical attributes.

3.2 CRUDAW: A Novel Fuzzy C-Means Clustering Technique

We now first introduce the components of CRUDAW and then use them to introduce the technique in details as follows.

SiDCAV: Similarity based Distances for Categorical Attribute Values

We use the similarity of two values C_1 and C_2 (belonging to a categorical attribute) to calculate their distance as follows.

$$distance\ (C_1, C_2) = 1 - similarity\ (C_1, C_2) \qquad (2)$$

The similarity of C_1 and C_2 can be calculated using an existing technique (Giggins 2009) that has been discussed in detail in Section 2 on Background Study. The similarity of any two categorical values can vary between 0 and 1.

NoNAV: Normalized Numerical Attribute Values

In order to maintain the consistency between a categorical and a numerical attribute in influencing the distance between two records, we normalize a numerical attribute so that its domain ranges between 0.0 and 1.0. Therefore, after normalization the distance between two values belonging to a numerical attribute can vary between 0.0 and 1.0, similar to a categorical attribute. The normalization is obtained as follows.

$$N(v) = \frac{(v - min)}{(max - min)} \qquad (3)$$

where $N(v)$ is the normalized numerical value, v is the original value of a numerical attribute, and min and max are the minimum and maximum values of the domain of the attribute.

WeDiF: Weighted Distance Function

We calculate the distance between the *i*th and the *j*th record, $\mathrm{R_i}$ and $\mathrm{R_j}$ using a novel weighted distance function as follows.

$$dist(R_\mathrm{i}, R_\mathrm{j}) = \frac{\sum_{a=1}^{m_1} w_a \left|R_{\mathrm{i,a}} - R_{\mathrm{j,a}}\right| + \sum_{a=\mathrm{m_1}+1}^{m} w_a * \mathrm{distance}\left(\mathrm{R_{i,a}}, \mathrm{R_{j,a}}\right)}{\sum_{a=1}^{m} w_a} \qquad (4)$$

Here, $R_{i,a}$ and $R_{j,a}$ are the *a*th attribute values belonging to the *i*th and *j*th record, w_a is a user defined weight (significance level) for attribute *a*, *n* is the number of numerical attributes (say, first m_1 attributes are numerical), m is the total number of attributes (both numerical and categorical), and $dist\left(R_{i,a}, R_{j,a}\right)$ is the similarity based distance (see SiDCAV) between records R_i and R_j for categorical attribute values of the *a*th attribute. According to the weighted distance function (WeDiF), the distance between two records $dist\left(R_i,\ R_j\right)$ can vary between zero and one. Besides, $\left|R_{i,a} - R_{j,a}\right|$ is the difference between the normalized values of the *a*th numerical attribute of records $\mathrm{R_i}$ and $\mathrm{R_j}$. The novel weighted function was first introduced in a previous study (Rahman and Islam 2011), but it is used for the first time in the clustering techniques and experiments of this study.

ISS: Initial Seed Selection

Unlike many existing techniques (Ahmad and Dey 2007a, Lee and Pedrycz 2009, Chatzis 2011), our proposed technique detects initial seeds using a deterministic process based on the density of the records in order to ensure a high quality of the initial seeds. We first calculate the number of records (density) within a user defined radius *r* of each record of a data set. That is, if there are *N* number of records in the data set within *r* distance (calculated using Equation 4 considering the user defined weight distribution of the attributes) of a record R_i then the density of R_i is *N*. We choose the record having the highest density as the first seed of the data set, provided the density of the first seed is greater than or equal to a user defined threshold T. We then remove all the records (including the first seed itself) that are within the *r* distance of the first seed while calculating the density of the remaining records of the data set. The record currently having the highest density is then picked as the second seed of the data set, if the density of the second seed is greater than or equal to T. We continue the process of seed selection while we find a seed having density greater than or equal to T.

```
Algorithm: Initial Seed Selection
Method 1: InitialSeed ()
Input: A dataset D, a user defined radius $r$, a user defined minimum number of records T, user defined attribute-weight-distribution W for all attributes.
Output: A set of Initial seeds S
-----------------------------------------------------------------------------------
/* Set initially the "set of initial seeds" to null*/
Set S← ∅
/* density of each record will be stored in the density vector $q$.  Set initially the density vector to null */
Set $q \leftarrow \emptyset$
/* index of the  record having the maximum density will be stored in max_density_rec variable. Set initially the max_density to null */
Set max_density_rec ← ∅
/* the set of records within r distance of the max_density_rec will be stored in $D_r$. */
Set $D_r$← ∅
/* the loop will continue while the remaining records of a data set is greater than or equal to T. */
WHILE |D| ≥ $T$   DO
    $q \leftarrow \mathbf{Density(D, r, W)}$  /* call Density (D, r, W) */
    /* the record having maximum density is returned by Index_max (q) */
    max_density_rec ←Index_max ($q$)
    /* if the maximum density max(q) is greater than or equal to T then the Index_max(q)record is considered to be an initial seed. */
    IF max ($q$) ≥ $T$
        S ← S ∪ max_density_rec
        /* Find_records (max_density_rec, r, W) returns the set of records that are within r distance of max_density_rec record*/
        $D_r$← Find_records (max_density_rec, r, W)
        D ← D − $D_r$
    ENDIF
    ELSE
        Break;
    END ELSE
ENDWHILE
Return S.
```

Figure 2: Algorithm for Initial Seed Selection

In our novel fuzzy clustering approach (CRUDAW) we then calculate initial fuzzy membership degree of each record of the data set from the initial seeds. A similar approach for initial seed selection was also taken by an existing technique (Andreopoulos 2006, Andreopoulos et al. 2007) that clusters records of a data set having categorical attributes. Our algorithm for initial seed selection is shown in the Figure 2 and Figure 3.

```
Algorithm: Initial Seed Selection
Method 2: Density()
Input: A data set $D$, a user defined radius $r$, user defined attribute-weight-distribution W
Output: A density vector $q$
--------------------------------------------------------------------------------
Set distance $d \leftarrow 0$, $q \leftarrow \emptyset$
FOR all records $R_i \in$ D   DO
    /*$c$ counts number of neighbor records of $R_i$ within its r distances */
    Set $c \leftarrow 0$
    FOR all records $R_j \in$ D   DO
        $d \leftarrow \text{distance}(\mathrm{R_i}, \mathrm{R_j}, W)$    /* call weighted distance function (WeDiF) */
        IF $d \leq r$
            c ++;
        END IF
    END FOR
    $q \leftarrow q \cup c$
ENDFOR
Return $q$;
```

Figure 3: Algorithm for Density calculation

FuMeD: Fuzzy Membership Degree

For our proposed Fuzzy clustering technique CRUDAW, we calculate the membership degree $\mu_{i,j}(0 \leq \mu_{i,j} \leq 1)$ of the ith record $\mathrm{R_i}$ with the jth cluster seed $\mathrm{S_j}$, $\forall\, i,j$ using the algorithm shown in Figure 4 following an existing membership degree calculation approach (Tang et al. 2010).

```
Algorithm: Fuzzy membership degree
Method: FuMeD()
Input: A dataset $D$, the set of seeds S, user defined attribute-weight-distribution W, a user defined fuzzy coefficient $\beta$.
Output: Fuzzy membership degree $\mu$ having size |$D$|*|S|
--------------------------------------------------------------------------------
Set $\mu \leftarrow \emptyset$
FOR all records $R_i \in$ D   DO
    FOR all seeds $S_j \in$ S   DO
        $\mu_{i,j} = \frac{1}{\sum_{l=1}^{|S|}\left(\frac{dist(R_i,S_j)}{dist(R_i,S_l)}\right)^{\frac{2}{\beta-1}}}$    /* the $dist(R_i,S_j)$ is calculated by using WeDiF using W */
    END FOR
    $\mu \leftarrow \mu \cup \mu_{i,j}$
END FOR

Return $\mu$;
```

Figure 4: Algorithm for Fuzzy membership degree

However, note that the distance between record R_i and seed S_j i.e. $dist(R_i,S_j)$ is calculated using our novel function for distance measure called WeDiF. A seed is considered to be structurally similar to a record in the sense that a seed has as many attributes as the number of attributes of a record.

SeCaF: Seed Calculation for Fuzzy Technique

Following traditional fuzzy C-means algorithms (Tang et al. 2010, Lee and Pedrycz 2009), we calculate the seed value $S_{j,a}$ of the jth cluster for the ath numerical attribute as follows.

$$S_{j,a} = \frac{\sum_{i=1}^{n}\mu_{i,j}{}^{\beta}R_{i,a}}{\sum_{i=1}^{n}\mu_{i,j}{}^{\beta}} \qquad (5)$$

Here, n is the total number of records in a data set. Note that we use normalized records while calculating the seed values for a numerical attribute.

```
Algorithm: Seed calculation for CRUDAW
Method: SeCaF()
Input: A dataset $D$ having altogether |A| number of attributes and |D| records, a set of fuzzy membership degrees $\mu$, a user defined fuzzy coefficient $\beta$, number of clusters |S|.
Output: A set of seeds S having size |S|*|A|
--------------------------------------------------------------------------------
Set $S \leftarrow \{S_1, S_2, \ldots S_{|S|}\}$ /* $\mathrm{S_j}$ is the $j$th seed. Initially $\mathrm{S_j}$ is a null set. */
FOR all $\mathrm{S_j} \in$ S   DO
    FOR all attributes $\mathrm{A_m} \in A$   DO
        IF $\mathrm{A_m}$ is categorical
                /* the summation of fuzzy
                membership degree for each value $v$ of
                the attribute $\mathrm{A_m}$ will be stored in M */
                Set M $\leftarrow$ 0
                FOR all domain values $v \in \mathrm{A_m}$
                    IF M$\leq \sum_{i=1}^{|D|} \mu_{i,j}{}^{\beta} \,|(R_{i,m} = v)$ DO
                        M$\leftarrow \sum_{i=1}^{|D|} \mu_{i,j}{}^{\beta} \,| R_{i,m} = v$
                        $S_{j,m} \leftarrow v$    /* $S_{j,m}$ is the $m$th attribute value of the $j$th seed */
                    END IF
                END FOR
        END IF
        ELSE    /* if attribute $\mathrm{A_m}$ is numerical */
                $S_{j,m} \leftarrow \frac{\sum_{i=1}^{|D|}\mu_{i,j}{}^{\beta}R_{i,m}}{\sum_{i=1}^{|D|}\mu_{i,j}{}^{\beta}}$
        END ELSE    /*$\mu_{i,j}$ is the fuzzy membership degree of $i$th record with $j$th cluster and $R_{i,m}$ is the $m$th attribute value of $i$th record */
        $S_j \leftarrow S_j \cup S_{j,m}$
    END FOR
END FOR

Return $S$;
```

Figure 5: Algorithm for Seed calculation in CRUDAW

However, following the approach taken by another existing fuzzy clustering technique (Kim et al. 2004), we calculate the seed value of a categorical attribute b as follows. Let, the domain values of a categorical attribute b are $\{\mathrm{b_1}, \mathrm{b_2}, \ldots\ldots\ldots, \mathrm{b_r}\}$, where r is the domain size for the attribute. The seed value of the attribute for the jth cluster, $\mathrm{S_{j,b}} = \mathrm{b_p}$ when the Equation 6 is satisfied.

$$\sum_{i=1;\ R_{i,b}=b_p}^{n} \mu_{i,j}^{\beta} \geq \sum_{i=1;\ R_{i,b}=b_q}^{n} \mu_{i,j}^{\beta}\,;\ \forall q \neq p \qquad (6)$$

Here, $1 \leq p \leq r$ and $1 \leq q \leq r$. That is, if the summation of the membership degrees (with the *j*th cluster) of all records having the value b_p (for the categorical attribute b) is greater than the summation of the membership degrees of all records having any other value (for all other values) then the seed value for the attribute b is equal to b_p for the *j*th cluster. The algorithm for seed calculation is shown in Figure 5.

TCFCM: Termination Conditions for Fuzzy Clustering Method

For CRUDAW, we use a weighted fuzzy objective function J_λ for the λ th iteration as follows.

$$J_\lambda = \sum_{i=1}^{n} \sum_{j=1}^{|S|} \mu_{i,j}^{\beta} * dist(R_i, S_j) \qquad (7)$$

If $|J_\lambda - J_{\lambda-1}| \leq \varepsilon$ or a user defined number of iterations η is completed then the clustering iteration terminates; otherwise it continues. $J_{\lambda-1}$ and J_λ are the objective function values in two consecutive iterations. Note that unlike the existing techniques we calculate $dist(R_i, S_j)$ using our WeDiF function.

FCFMD: Final Clustering based on Fuzzy Membership Degrees

```
Algorithm: CRUDAW Algorithm
Input:  A dataset $D$, a user defined radius $r$, a user defined minimum
number of records $T$, user defined attribute-weight-distribution W, a
use defined fuzzy coefficient $\beta$, a user defined objective function
threshold $\epsilon$, a user defined maximum number of iteration $\eta$
Output: A set of rigid clusters $C$, and a set of membership degree $\mu$
--------------------------------------------------------------------------------
Method:
Step 1: Normalize the data set D using NoNAV function
            D ← $Normalize\ (D)$    /* call NoNAV function*/
Step 2: Initial Seed Selection
            S← InitialSeed(D, r, T, W )   /* call ISS function*/
Set $J_{cur} \leftarrow 0, J_{prev} \leftarrow 0$
FOR ($\lambda$ =1 to $\eta$) DO   /* $\lambda$ counts the number of iteration*/
    Step 3: Fuzzy membership degree
            μ←FuMeD(D, S, W, $\beta$)   /* call FMD function*/
    Step 4: Seed calculation
            S← SeCaF(D, μ, $\beta$,|S|)             /* call SCF function*/
    Step 5: Termination conditions for CRUDAW
            $J_{cur}$ ← TCFCM (D, S, W, $\mu$, $\beta$)   /*call TCFCM
                                                 function*/
            IF $\lambda$>1 && ( $|J_{cur} - J_{prev}| \leq \varepsilon$ ) DO
                    Break;     /* terminate clustering as it meets
                               the termination condition*/
            END IF
                    $J_{prev} \leftarrow J_{cur}$
END FOR
Step 6: Produce the final clusters based on fuzzy membership degree
            C← FCFMD (μ, D, S)
Return C, $\mu$.
```

Figure 6: CRUDAW Algorithm

CRUDAW finally produces two outputs; the first output is a set of fuzzy membership degrees of each record with all cluster centers (seeds) and the second output is the rigid clustering where each record is assigned to the cluster for which it has the highest membership degree. This way a record is associated with only one cluster. It also returns the rigid clustering since a user often may need it for a number of purposes. We now present the algorithm (Figure 6) and block diagram (Figure 7) for CRUDAW integrating various components introduced above.

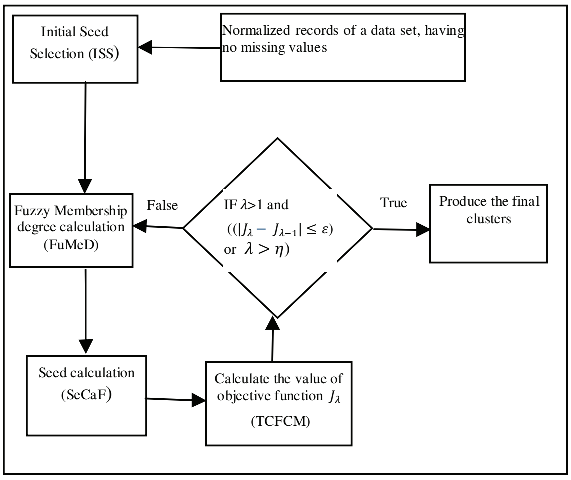
Initial Seed Selection (ISS)
Normalized records of a data set, having no missing values
Fuzzy Membership degree calculation (FuMeD)
False
IF λ>1 and (($|J_\lambda - J_{\lambda-1}| \leq \varepsilon$) or $\lambda > \eta$)
True
Produce the final clusters
Seed calculation (SeCaF)
Calculate the value of objective function J_λ (TCFCM)

Figure 7: Block Diagram of CRUDAW

4 Experimental Results

We implement our technique CRUDAW and a few existing techniques namely SABC (Ahmad and Dey 2007a), GFCM (Lee and Pedrycz 2009), and KL-FCM-GM (Chatzis 2011). We use a few evaluation criteria, specifically Silhouette Coefficient, F-measure (with $\delta = 1$), Entropy, and Purity (Tan et al. 2005, Chuang 2004, Kashef and Kamel 2009, Rahman and Islam 2011) to compare the performance of the technique. We also use t-test (Johnson and Bhattacharyya 1985, Moore 1995) and confidence interval test (Johnson and Bhattacharyya 1985, Moore 1995, Triola 2001) to estimate the statistical significance of the performance of the technique.

Data set	Records (size)	Categorical attributes (cat)	Numerical attributes (num)	Missing values	Classification Accuracy	Class size
Mushroom	8124	22	0	yes	95%	2
Credit Approval	690	9	6	yes	80%	2
Pima Indian Diabetes	768	0	8	no	72%	2
Contraceptive Method Choice(CMC)	1473	7	2	no	52.4%	3

Table 2: Information on data sets at a glance

We use four natural data sets namely Mushroom, Credit Approval, Pima Indian Diabetes, and Contraceptive Method Choice (CMC) – all of them are available from UCI machine learning repository (UCI

2012). Brief information on the data sets is presented in Table 2.

The Mushroom data set and Credit Approval data set have some missing values. We first remove all records having any missing values. After removing the records having missing values Mushroom, Credit Approval data sets have 5644 and 653 records, respectively. We also remove the class attributes from the data sets before we apply clustering techniques on them. The class attributes are used again for the cluster evaluation based on F-measure, Purity and Entropy.

In all experiments on CRUDAW, we use fuzzy coefficient β=2.2, fuzzy termination condition ε = 0.005, and for initial seed selection T=1% of the records of a data set. However, in all experiments on GFCM we use fuzzy coefficient β =1.3 and fuzzy termination condition ε = 0.0001 following the recommendation of the original paper (Lee and Pedrycz 2009) in order to achieve the best result from the technique. Similarly, for the experiments on KL-FCM-GM we use degree of fuzziness =1.5 and fuzzy termination condition ε =0.005 as recommended for obtaining the best result from the technique (Chatzis 2011). The maximum number of iterations for CRUDAW, SABC, GFCM and KL-FCM-GM are considered to be 50.

For CRUDAW, a user can assign weights on the attributes according to his/her requirement. A user assigns higher weights on the attributes that he/she considers to be more important for clustering the records, as discussed in Section 3.1. The evaluation criteria (used in this study) such as F-measure, Purity and Entropy focus on the ability of a clustering technique to group the records in homogeneous collections where in each collection all records have the same class value. Therefore, in order to match the focus of the evaluation criteria we (in this experiment) consider that a user assigns high weights on the attributes that are strongly related to the class attribute – i.e. high weights on the attributes having low entropy with respect to the class values (Quinlan 1993, Quinlan 1996, Islam 2012).

We argue that if CRUDAW can achieve good F-measure, Purity and Entropy values under such weight distribution then they should also achieve good clustering results (according to the purposes of the users) when the users assign a different weight distribution following their purposes.

Based on entropy values of the attributes of a data set we divide them into three categories namely the best attributes (BA) consisting of the attributes having low entropies, medium attributes (MA), and the worst attributes (WA). If the number of attributes of a data set is divisible by three then each category contains one third of the total number of attributes. Otherwise, the best and the worst categories have the same number of attributes while the medium category contains more attributes. In order to simulate different user attitudes we use different weight patterns to assign high weights on the best attributes (BA) and a combination of attributes from the best and medium categories (BM).

We now explain the weight patterns for the best attributes (BA), and the best and medium attributes (BM) using an example on CMC data set (see Table 3). Using the entropy of the attributes, we rank the attributes where attributes having low entropy (i.e. good attributes) are ranked low. Various weight patterns such as BA1, BA2, and BM1 are shown in Table 3.

Weights on best attributes (BA)										
Attribute	A1	A2	A3	A4	A5	A6	A7	A8	A9	Notations
Rank	2	1	3	8	7	9	5	4	6	
Best attributes (BA)	0.2	0.2	0.2	0	0	0	0	0	0	BA1
	0.2	0.4	0.2	0	0	0	0	0	0	BA2
	0.4	0.6	0.2	0	0	0	0	0	0	BA3
	0.6	0.8	0.4	0	0	0	0	0	0	BA4
	0.8	1	0.6	0	0	0	0	0	0	BA5
	Weights on best and medium attributes (BM)									
Best and Medium attributes (BM)	0.2	0.2	0	0	0	0	0	0.2	0	BM1
	0.2	0.4	0	0	0	0	0	0.4	0	BM2
	0.4	0.6	0	0	0	0	0	0.6	0	BM3
	0.6	0.8	0	0	0	0	0	0.8	0	BM4
	0.8	1.0	0	0	0	0	0	1	0	BM5

Table 3: Weights pattern for CMC data set

In the experiments of CRUDAW, we use three different r values for each data set in order to test the technique on different numbers of seeds or clusters. For comparing CRUDAW with the existing techniques we produce the same numbers of clusters for all techniques.

However, SABC, GFCM, and KL-FCM-GM do not produce initial seeds deterministically and therefore, produce different sets of clusters in different runs. That is if we run SABC twice to produce say 7 clusters we may get different sets of 7 clusters. Hence, we run each of these existing techniques ten times for each number of clusters. We then calculate the average Silhouette coefficients, F-measures, Entropy, and Purity for the ten runs.

Silhouette Coefficient						
Weights	r	k	CRUDAW	SABC (avg: 10 exp)	GFCM (avg: 10 exp)	KL-FCM-GM (avg: 10 exp)
BA1	0.05	6	0.5338[4]	0.5092[3]	0.3889[2]	0.0717[1]
	0.02	20	0.7777[4]	0.2702[3]	0.142[2]	0.0649[1]
	0.01	26	0.8536[4]	0.2144[3]	0.0877[2]	0.0425[1]
BA2	0.05	7	0.7609[4]	0.5028[3]	0.3831[2]	0.0983[1]
	0.02	17	0.6992[4]	0.2678[3]	0.1416[2]	0.078[1]
	0.01	26	0.8504[4]	0.214[3]	0.0877[2]	0.0413[1]
BA3	0.05	8	0.8409[4]	0.387[3]	0.2988[2]	0.2043[1]
	0.02	14	0.5612[4]	0.3241[3]	0.1756[2]	0.068[1]
	0.01	26	0.5612[4]	0.2165[3]	0.0868[2]	0.0404[1]
BA4	0.05	8	0.8771[4]	0.4031[3]	0.2975[2]	0.2068[1]
	0.02	12	0.6126[4]	0.4717[3]	0.1661[2]	0.0802[1]
	0.01	22	0.7243[4]	0.2819[3]	0.1109[2]	0.0579[1]
BA5	0.05	8	0.8872[4]	0.4036[3]	0.2964[2]	0.2101[1]
	0.02	11	0.7439[4]	0.4297[3]	0.2322[2]	0.0938[1]
	0.01	16	0.6335[4]	0.43[3]	0.1485[2]	0.0854[1]
Total Score			60	45	30	15

Table 4: Silhouette Coefficient based on best attributes (BA) of Mushroom data set

In Table 4, we present silhouette coefficients of CRUDAW for the weight patterns on the best attributes (BA) of Mushroom data set. We also present the *average* silhouette coefficients of SABC, GFCM, and KL-FCM-GM from *ten runs* of each technique for each number of clusters i.e. for each k value in Table 4. For weight pattern BA1 and r = 0.05, we get six initial seeds and therefore, six clusters (k=6) for CRUDAW. The

Silhouette coefficient for the six clusters of CRUDAW is 0.5338. For same number of clusters the average (of all ten runs) Silhouette coefficients of SABC, GFCM and KL-FCM-GM are 0.5092; and 0.3889 and 0.0717, respectively. Similarly, we also estimate the F-measure, Entropy and Purity (Table 5, Table 6 and Table 7).

F-measure						
Weights	r	k	CRUDAW	SABC (avg: 10 exp)	GFCM (avg: 10 exp)	KL-FCM-GM (avg: 10 exp)
BA1	0.05	6	0.9025[4]	0.8682[2]	0.83[3]	0.5252[1]
	0.02	20	0.9971[4]	0.7929[2]	0.8322[3]	0.6875[1]
	0.01	26	0.9971[4]	0.744[2]	0.8341[3]	0.7363[1]
BA2	0.05	7	0.9781[4]	0.8453[3]	0.8336[2]	0.5822[1]
	0.02	17	0.9971[4]	0.7728[2]	0.832[3]	0.6467[1]
	0.01	26	0.9971[4]	0.744[2]	0.8341[3]	0.7363[1]
BA3	0.05	8	0.9971[4]	0.843[3]	0.8356[2]	0.8252[1]
	0.02	14	0.9865[4]	0.8392[3]	0.8335[2]	0.6383[1]
	0.01	26	0.9908[4]	0.744[2]	0.8341[3]	0.7363[1]
BA4	0.05	8	0.9971[4]	0.843[3]	0.8356[2]	0.8252[1]
	0.02	12	0.9971[4]	0.8733[3]	0.8318[2]	0.6489[1]
	0.01	22	0.99[4]	0.7865[2]	0.8322[3]	0.6556[1]
BA5	0.05	8	0.9971[4]	0.843[2]	0.8356[3]	0.8252[1]
	0.02	11	0.9971[4]	0.8783[3]	0.8335[2]	0.6026[1]
	0.01	16	0.9971[4]	0.9036[3]	0.8365[2]	0.6934[1]
Total Score			60	37	38	15

Table 5: F-measure based on best attributes (BA) of Mushroom data set

Entropy						
Weights	r	k	CRUDAW	SABC (avg: 10 exp)	GFCM (avg: 10 exp)	KL-FCM-GM (avg: 10 exp)
BA1	0.05	6	0.3344[4]	0.3821[3]	0.535[2]	0.8536[1]
	0.02	20	0.0183[4]	0.3607[3]	0.5235[2]	0.6867[1]
	0.01	26	0.0183[4]	0.416[3]	0.5001[2]	0.637[1]
BA2	0.05	7	0.132[4]	0.3886[3]	0.5138[2]	0.7987[1]
	0.02	17	0.0197[4]	0.4124[3]	0.5124[2]	0.7075[1]
	0.01	26	0.0183[4]	0.416[3]	0.5001[2]	0.637[1]
BA3	0.05	8	0.024[4]	0.3653[3]	0.4935[2]	0.559[1]
	0.02	14	0.0526[4]	0.3098[3]	0.5173[2]	0.7415[1]
	0.01	26	0.0437[4]	0.416[3]	0.5001[2]	0.637[1]
BA4	0.05	8	0.0247[4]	0.3653[3]	0.4935[2]	0.559[1]
	0.02	12	0.0218[4]	0.2622[3]	0.5256[2]	0.7079[1]
	0.01	22	0.0437[4]	0.3756[3]	0.5012[2]	0.7219[1]
BA5	0.05	8	0.0247[4]	0.3653[3]	0.4935[2]	0.559[1]
	0.02	11	0.0218[4]	0.2793[3]	0.5201[2]	0.7699[1]
	0.01	16	0.0197[4]	0.2032[3]	0.5121[2]	0.6679[1]
Total Score			60	45	30	15

Table 6: Entropy based on best attributes (BA) of Mushroom data set

We now compare the techniques through their scores (as shown within the brackets/parentheses) based on a scoring rule where we assign 4, 3, 2, and 1 point for the techniques having the best, the 2nd best, the 3rd best, and the worst result, respectively. For each evaluation criteria, the Total Scores of CRUDAW are significantly better than the scores of any other techniques for the Mushroom data set. See Table 4, Table 5, Table 6 and Table 7 for more information. Note that all distances are calculated using Equation 4 following the weights assigned in a weight pattern.

Finally, in Table 8 and Table 9 we show the total scores of the techniques for BA and BM weight patterns for all evaluation criteria. The total score (as shown in the last row of Table 8 and Table 9) of each technique is also presented in Figure 8, which shows a clear domination of CRUDAW over all other techniques for all evaluation criteria. Similarly, from Figure 9 to Figure 11, we present total scores of the techniques for Credit Approval (CA), Pima Indian Diabetes (PID), and Contraceptive Method Choice (CMC) data sets.

Purity						
Weights	r	k	CRUDAW	SABC	GFCM	KL-FCM-GM
BA1	0.05	6	0.8986[4]	0.8862[3]	0.8532[2]	0.6723[1]
	0.02	20	0.9971[4]	0.8647[3]	0.8557[2]	0.7746[1]
	0.01	26	0.9971[4]	0.8389[3]	0.8573[2]	0.8026[1]
BA2	0.05	7	0.978[4]	0.8775[3]	0.8565[2]	0.7086[1]
	0.02	17	0.9971[4]	0.8464[2]	0.8551[3]	0.7555[1]
	0.01	26	0.9971[4]	0.8389[2]	0.8573[3]	0.8026[1]
BA3	0.05	8	0.9971[4]	0.8763[3]	0.858[2]	0.8493[1]
	0.02	14	0.9865[4]	0.8881[3]	0.855[2]	0.746[1]
	0.01	26	0.9907[4]	0.8389[2]	0.8573[3]	0.8026[1]
BA4	0.05	8	0.9971[4]	0.8763[2]	0.858[3]	0.8493[1]
	0.02	12	0.9971[4]	0.9051[3]	0.8551[2]	0.7581[1]
	0.01	22	0.9907[4]	0.8587[3]	0.8553[2]	0.7556[1]
BA5	0.05	8	0.9971[4]	0.8763[3]	0.858[2]	0.8493[1]
	0.02	11	0.9971[4]	0.906[3]	0.8561[2]	0.7318[1]
	0.01	16	0.9971[4]	0.9288[3]	0.8585[2]	0.7812[1]
Total Score			60	45	30	15

Table 7: Purity based on best attributes (BA) of Mushroom data

Dataset: Mush Room ; records=5644; cat=21								
Weight	Score based on Silhouette coefficient				Score based on F-measure			
	CRUDAW	SABC	GFCM	KL-FCM-GM	CRUDAW	SABC	GFCM	KL-FCM-GM
BA	60	45	30	15	60	37	38	15
BM	59	46	30	15	60	40	35	15
Total	119	91	60	30	120	77	73	30

Table 8: Score comparison based on Mushroom data set

Dataset: Mush Room ; records=5644; cat=21								
Weight	Score based on Entropy				Score based on Purity			
	CRUDAW	SABC	GFCM	KL-FCM-GM	CRUDAW	SABC	GFCM	KL-FCM-GM
BA	60	45	30	15	60	41	34	15
BM	59	46	30	15	60	42	33	15
Total	119	91	60	30	120	83	67	30

Table 9: Score comparison based on Mushroom data set

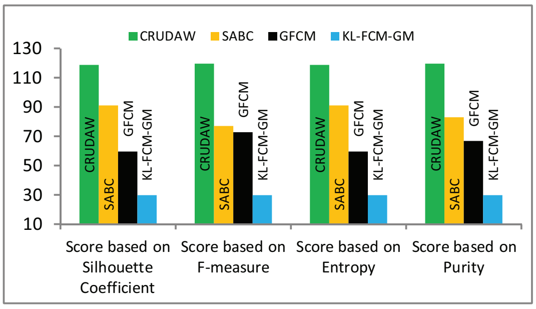

Figure 8: Score comparison based on Mushroom data set

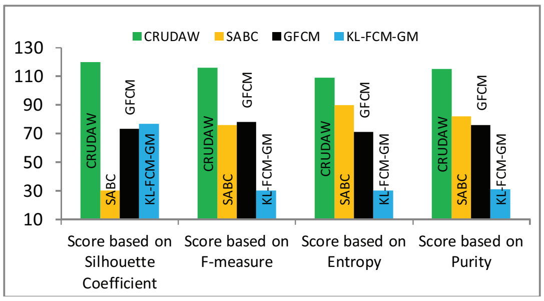

Figure 9: Score comparison based on Credit Approval data set

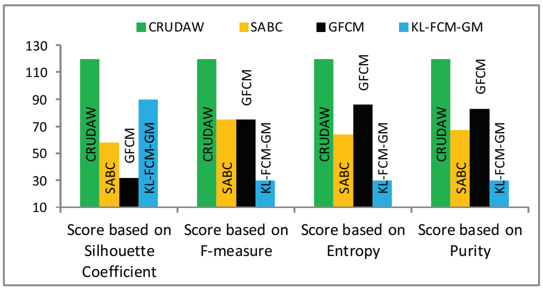

Figure 10: Score comparison based on Pima Indian Diabetes data set

Overall, CRUDAW performs clearly better than the other techniques. In Mushroom, Credit Approval, Pima Indian Diabetes, and Contraceptive Method (CMC) our technique score higher than all other techniques for all evaluation criteria as presented in the figures below (Figure 12 to Figure 15)

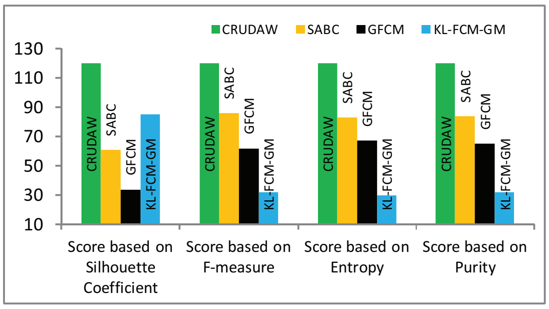

Figure 11: Score comparison based on CMC data set

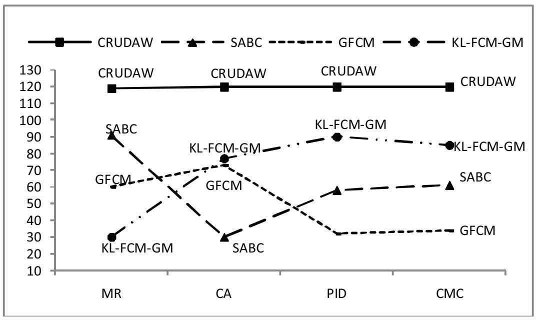

Figure 12: Score comparison for Silhouette Coefficient on all datasets

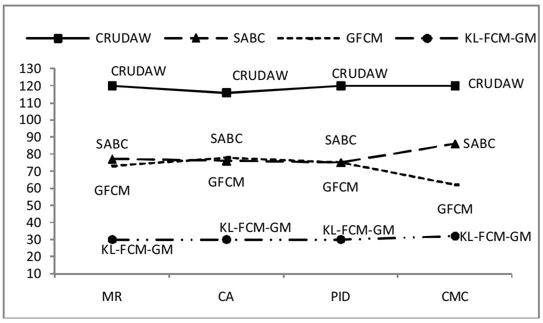

Figure 13: Score comparison for F-measure on all datasets

We now use statistical t-test (Johnson and Bhattacharyya 1985, Moore 1995) in order to explore whether the results of various evaluation criteria for our technique are significantly higher than the results for the existing techniques. In the t-tests, we considered p = 0.05 (i.e. 95% significance level) and degrees of freedom (df) =58. For p = 0.05 and df = 58 the t-ref value is 1.644 which we call "t-ref" (reference t-value).

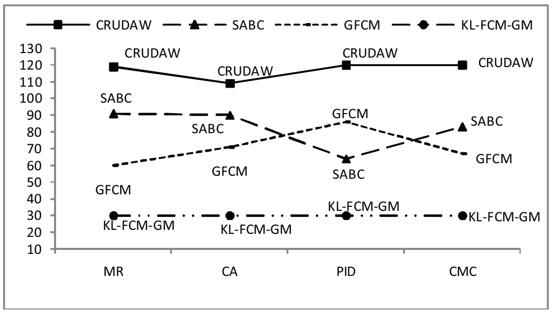

Figure 14: Score comparison for Entropy on all datasets

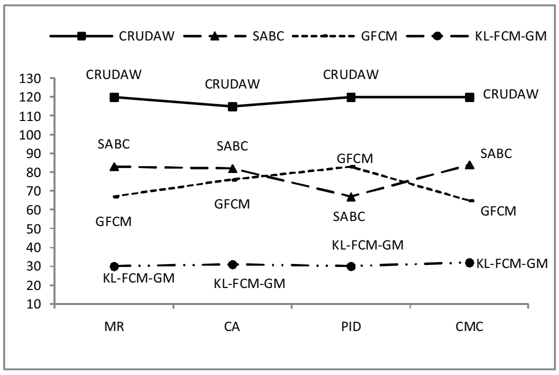

Figure 15: Score comparison for Purity on all datasets

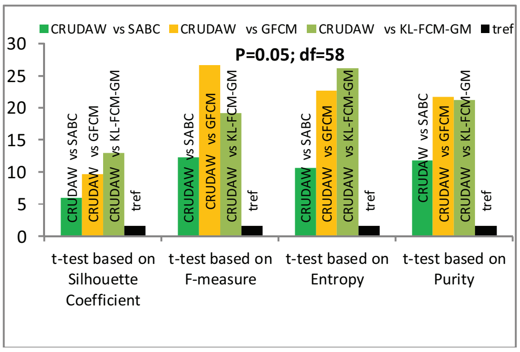

Figure 16: t-test for CRUDAW on Mushroom data set

In the figures (Figure 16 to Figure 18), we present t-test results of CRUDAW compared with other techniques on the Mushroom, Credit Approval, and Pima Indian Diabetes data sets. In Figure 16, the first bar from the left side ("CRUDAW vs SABC") is taller than the t-ref bar meaning that the actual Silhouette coefficient values (not the score) of CRUDAW are significantly better than the Silhouette coefficient values of SABC technique at 95%

significance level. The t-test results for CMC data set is presented in tabular form (see Table 10). We experience difficulties in presenting them in graphical form due to huge differences between the values.

We also carry out the Confidence Interval analysis (Johnson and Bhattacharyya 1985, Moore 1995, Triola 2001) at 90% confidence level for all data sets. The confidence intervals for actual silhouette coefficient for Mushroom (MR), Credit Approval (CA), Pima Indian Diabetes (PID), and Contraceptive Method Choice (CMC) are presented in Figure 19. Similarly, Figure 20 presents the confidence intervals for F-measure for the data sets

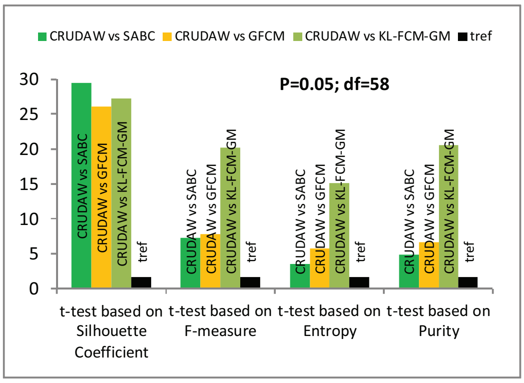

Figure 17: t-test for CRUDAW on Credit Approval data set

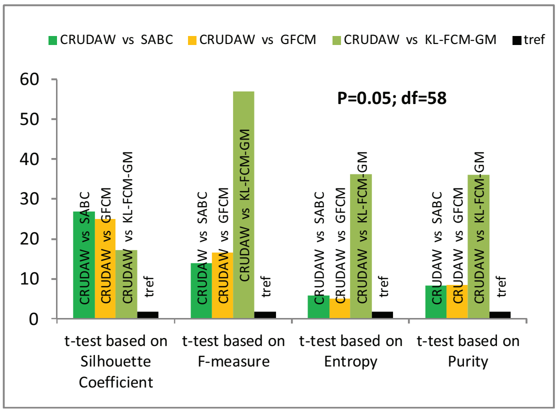

Figure 18: t-test for CRUDAW on Pima Indian Diabetes data set

Data set : CMC ; records:1473; c=7; n=2											
P=0.05; tref=1.644; df=58; T1 = CRUDAW ; KF = KL-FCM-GM											
t-value based on Silhouette coefficient			t-value based on F-measure			t-value based on Entropy			t-value based on Purity		
T1 vs SABC	T1 vs GFCM	T1 vs KF	T1 vs SABC	T1 vs GFCM	T1 vs KF	T1 vs SABC	T1 vs GFCM	T1 vs KF	T1 vs SABC	T1 vs GFCM	T1 vs KF
22.63	40.90	42.35	14.33	40.44	90.18	6.62	8.21	16.04	8.51	12.86	13.74

Table 10: t-test on CMC data set

According to Figure 19 and Figure 20, the average values of Silhouette coefficient (the most natural evaluation criterion) and F-measure (a combination of precision and recall) for CRUDAW are clearly better than other techniques for all data sets. Moreover, there is no overlap of the confidence intervals of CRUDAW with the intervals of other techniques. This is the case for other two evaluation criteria as well. We use results of 30 experiments for confidence interval calculation; 15 from BA categories and 15 from BM categories. However, each of the 30 results is the average value of 10 runs as explained before (see Table 3 to Table 7).

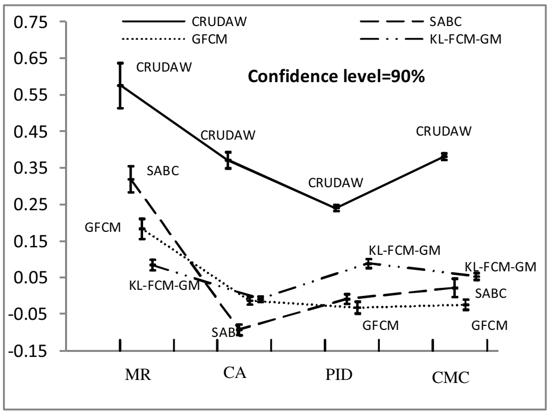

Figure 19: Confidence Interval based on Silhouette coefficient

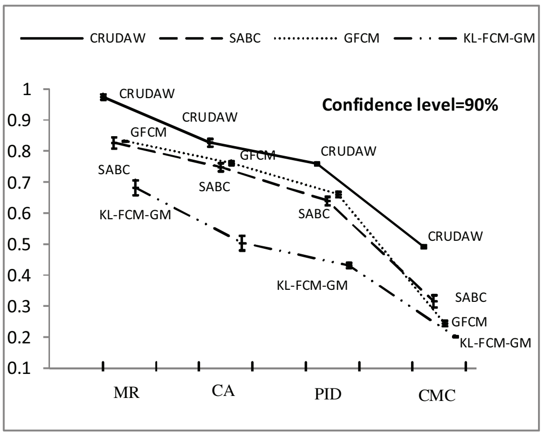

Figure 20: Confidence Interval based on F-measure

Overall average computational time (seconds) of the techniques for all data				
Data sets	CRUDAW	KL-FCM-GM	SABC	GFCM
Credit Approval	2.05	127.8125	2.235	0.1465
Pima Indian Diabetes	2.0995	38.806	1.169	1.152
Contraceptive Method Choice	4.7635	350.874	2.136	0.119
Mushroom	213.32	59.53	45.812	1.6775

Table 11: Overall average computational time (in seconds) of the techniques for all data set

We also calculate the overall time required for clustering by the techniques (Table 11). For experiments on time complexity analysis we use a shared computer system the configuration of which is 4x8 core Intel E7-8837 Xeon processors, 256 GB of RAM, and 23 TB of disk storage. Generally KL-FCM-GM technique requires the maximum amount of time in our experiments. Perhaps due to random selection of initial seeds SABC and GFCM require less computation time than

CRUDAW at the cost of relatively inferior quality of clusters. Finally Table 12 presents a comparison between the time complexity of CRUDAW and an existing technique called Seed-Detective.

Data set	Seed-Detective Execution Time (sec.)	CRUDAW Time (sec.)
CMC	35.01	4.76
CA	17.62	2.05

Table 12: Overall average computational time (in seconds)

5 Conclusion

In this study we present a novel clustering technique called CRUDAW. Our proposed technique (CRUDAW) allows a data miner to assign weights on the attributes of a data set based on their importance (to the data miner) for clustering. The technique uses a novel approach to select initial seeds deterministically using the density of the records of a data set. CRUDAW selects the initial fuzzy membership degrees deterministically. CRUDAW also uses a novel approach for measuring distance considering the user defined weights of the attributes. Moreover, while measuring the distance between the values of a categorical attribute the technique takes the similarity of the values into consideration. We also present complete algorithms for the technique.

We experimentally compare our technique with a few existing techniques namely SABC, GFCM, KL-FCM-GM based on various evaluation criteria called Silhouette coefficient, F-measure, purity and entropy. The experimental results strongly indicate the supremacy of our novel technique over the existing techniques. For all data sets used in this study, our technique scores higher than all other techniques for all evaluation criteria.

We carry out statistical t-tests to ensure the significance of the better result of our technique. We then also perform confidence interval tests at 90% confidence level. Both tests confirm the statistical significance of the superior results achieved by CRUDAW.

We also record the time complexity (during execution) of the technique. CRUDAW performs better than KL-FCM-GM and Seed-Detective. However, SABC and GFCM require less computation time than CRUDAW, perhaps due to their random seed selection approach, at the cost of relatively inferior quality of clusters. Hence, for non-time critical applications requiring good quality clusters, we believe CRUDAW is more suitable than the existing techniques tested in this study.

Our future research goals include a further improvement of the technique, reduction of time complexity, and automatic generation of attribute weights as a suggestion for a user.

6 References

Ahmad, A. and Dey, L. (2007a): A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering, 63*(2), 503-527. doi: 10.1016/j.datak.2007.03.016

Ahmad, A. and Dey, L. (2007b): A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set. *Pattern Recognition Letters, 28*(1), 110-118. doi: 10.1016/j.patrec.2006.06.006

Alata, M., Molhim, M. and Ramini, A. (2008): Optimizing of Fuzzy C-Means Clustering Algorithm Using GA *World Academy of Science, Engineering and Technology* (Vol. 39, pp. 224-229).

Andreopoulos, B., An, A. and Wang, X. (2007): Hierarchical Density-Based Clustering of Categorical Data and a Simplification. *Proc. 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2007)*, Nanjing, China.

Andreopoulos, W. (2006): Clustering Algorithms for Categorical Data. Ph.D. thesis. York University, Toronto, Ontario.

Bai, L., Liang, J. and Dang, C. (2011): An initialization method to simultaneously find initial cluster centers and the number of clusters for clustering categorical data. *Knowledge-Based Systems, 24*(6), 785-795. doi: 10.1016/j.knosys.2011.02.015

Bezdek, J. J. (1981): *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum.

Chatzis, S. P. (2011): A fuzzy c-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional. *Expert Systems with Applications, 38*(7), 8684-8689. doi: 10.1016/j.eswa.2011.01.074

Chuang, K.-T. and Chen, M.-S. (2004): Clustering Categorical Data by Utilizing the Correlated-Force Ensemble. *Proc. 4th SIAM International Conference on Data Mining (SDM 04),* Lake Buena Vista, Florida.

Ganti, V., Gehrke, J. and Ramakrishnan, R. (1999): CACTUS–Clustering Categorical Data Using Summaries. *Proc. Fifth ACM SIGKDD international conference on Knowledge discovery and data mining* San Diego, CA, USA.

Gath, I. and Geva, A. B. (1989): Unsupervised optimal fuzzy clustering. *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence,* 11(7), 773–781.

Giggins, H. P. (2009): Security of Genetic Databases. Ph.D. thesis. School of Electrical Engineering and Computer Science, The University of Newcastle, Australia.

Grubesic, T. H. and Murray, A. T. (2001): Detecting Hot Spots Using Cluster Analysis and GIS. *Proc. 5th Annual International Crime Mapping Research Conference,* Dallas, TX. USA.

Guha, S., Rastogi, R. and Shim, K. (1998): CURE: an efficient clustering algorithm for large databases. *Proc. ACM SIGMOD international conference on Management of data (SIGMOD'98)* ,New York, NY, USA

Han, J. Kamber, M. (2006): *Data Mining Concepts and Techniques* (2nd ed.). San Francisco: Morgan Kaufmann.

Hasan, M. A., Chaoji, V., Salem, S. and Zaki, M. J. (2009): Robust partitional clustering by outlier and density insensitive seeding. *Pattern Recognition*

Letters, 30(11), 994-1002. doi: 10.1016/j.patrec.2009.04.013

Hathaway, R. J. and Bezdek, J. C. (1988): Recent Convergence Results for the Fuzzy c-Means Clustering Algorithms. *Journal of Classification, 5*(2), 237-247. doi: DOI: 10.1007/BF01897166

Huang, D. and Pan, W. (2006): Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. *Journal of Bioinformatics 22*(10), 1259-1268. doi: doi>10.1093/bioinformatics/btl065

Huang, Z. (1997): Clustering large data sets with mixed numeric and categorical values. *Proc. First Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Singapore.

Huang, Z. and Ng, M. K. (1999): A Fuzzy k-Modes Algorithm for Clustering Categorical Data. *IEEE Transactions on Fuzzy Systems, 36*(2), 1615-1620. doi: 10.1016/j.eswa.2007.11.045

Islam, M. Z. (2012): EXPLORE: A Novel Decision Tree Classification Algorithm, *Data Security and Security Data,* LNCS, Vol. 6121, L.M. MacKinnon (Ed.), Springer, Berlin/Heidelberg, ISBN 978-3-642-25703-2, pg. 55-71.

Islam, M. Z. and Brankovic, L. (2011): Privacy Preserving Data Mining: A Noise Addition Framework Using a Novel Clustering Technique. *Journal of Knowledge-Based Systems*. Vol. 24, Issue 8, ISBN 0950-7051, DOI: 10.1016/j.knosys.2011.05.011.

Islam, M. Z. and Brankovic, L. (2005): DETECTIVE: A Decision Tree Based Categorical Value Clustering and Perturbation Technique in Privacy Preserving Data Mining. *Proc. 3rd International IEEE Conference on Industrial Informatics,* Perth, Australia.

Islam, M. Z. (2008): Privacy Preservation in Data Mining through Noise Addition. Ph.D. Thesis. School of Electrical Engineering and Computer Science, The University of Newcastle, Australia.

Ji, J., Pang, W., Zhou, C., Han, X. and Wang, Z. (2012): A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data. *Knowledge-Based Systems, 30*(0), 129-135. doi: 10.1016/j.knosys.2012.01.006

Johnson, R. and Bhattacharyya, G. (1985): *Statistics Principles and Methods*, Revised Printing, John Wiley and Sons.

Kashef, R. and Kamel, M. S. (2009): Enhanced bisecting -means clustering using intermediate cooperation. *Pattern Recognition, 42*(11), 2557-2569. doi: 10.1016/j.patcog.2009.03.011

Kim, D.-W., Lee, K. H. and Lee, D. (2004): Fuzzy clustering of categorical data using fuzzy centroids. *Pattern Recognition Letters, 25*(11), 1263-1271. doi: 10.1016/j.patrec.2004.04.004

Lee, M. and Pedrycz, W. (2009): The fuzzy C-means algorithm with fuzzy P-mode prototypes for clustering objects having mixed features. *Fuzzy Sets and Systems, 160*(24), 3590-3600. doi: 10.1016/j.fss.2009.06.015

Li, M. J., Ng, M. K., Cheung, Y.-m. and Huang, J. Z. (2008): Agglomerative Fuzzy K-Means Clustering Algorithm with Selection of Number of Clusters. *IEEE Transactions on Knowledge and Data Engineering, 20*(11), 1519-1534. doi: 10.1109/TKDE.2008.88

Lung, C.-H., Zaman, M. and Nandi, A. (2004): Applications of clustering techniques to software partitioning, recovery and restructuring. *Journal of Systems and Software, 73*(2), 227-244. doi: 10.1016/s0164-1212(03)00234-6

Masulli, F. and Schenone, A. (1999): A fuzzy clustering based segmentation system as support to diagnosis in medical imaging. *Artificial Intelligence in Medicine, 16*(2), 129-147. doi: 10.1016/s0933-3657(98)00069-4

Moore, D. S. (1995): *The Basic Practice of Statistics*. W. H. Freeman and Company, New York.

Quinlan, J. R. (1993): *C4.5: programs for machine learning,* San Francisco, CA, USA Morgan Kaufmann Publishers Inc.

Quinlan, J. R. (1996): Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research, 4*(1), 77-90.

Rahman, M. A. and Islam, M. Z. (2011): Seed-Detective: A Novel Clustering Technique Using High Quality Seed for K-Means on Categorical and Numerical Attributes. *Proc. Ninth Australasian Data Mining Conference: AusDM 2011*, vol. 121. University of Ballarat, Australia.

Redmond, S. J. and Heneghan, C. (2007): A method for initialising the K-means clustering algorithm using kd-trees. *Pattern Recognition Letters, 28*(8), 965-973. doi: 10.1016/j.patrec.2007.01.001

Saha, I., Maulik, U. and Plewczynski, D. (2011): A new multi-objective technique for differential fuzzy clustering. *Applied Soft Computing, 11*(2), 2765-2776. doi: 10.1016/j.asoc.2010.11.007

Song, J. and Nicolae, D. L. (2009): A sequential clustering algorithm with applications to gene expression data. *Journal of the Korean Statistical Society, 38*(2), 175-184. doi: 10.1016/j.jkss.2008.09.006

Tan, P.-N., Steinbach, M. and Kumar, V. (2005): *Introduction to Data Mining* (1st ed.): Pearson Addison Wesley.

Tang, C., Wang, S. and Xu, W. (2010): New fuzzy c-means clustering model based on the data weighted approach. *Data & Knowledge Engineering, 69*(9), 881-900. doi: 10.1016/j.datak.2010.05.001

Triola, M. F. (2001): Elementary Statistics, 8th ed. Addison Wesley Longman, Inc.

Tsai, C. Y. and Chiu, C. C. (2004): A purchase-based market segmentation methodology. *Expert Systems with Applications, 27*(2), 265-276. doi: 10.1016/j.eswa.2004.02.005

UCI (2012): UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/. Accessed 11 July 2012

Zamir, O. and Etzioni, O. (1999): Grouper: a dynamic clustering interface to Web search results. *Computer Networks: The International Journal of Computer and*

Telecommunications Networking, 31(11-16), 1361 - 1374 doi: doi>10.1016/S1389-1286(99)00054-7

Zhang, T., Ramakrishnan, R. and Livny, M. (1996): BIRCH: an efficient data clustering method for very large databases. *Proc. ACM SIGMOD international conference on Management of data* New York, NY, USA

Zhao, P. and Zhang, C.-Q. (2011): A new clustering method and its application in social networks. *Pattern Recognition Letters, 32*(15), 2109-2118. doi: 10.1016/j.patrec.2011.06.008