Journal of King Saud University – Computer and Information Sciences (2015) **27**, 113–128

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

ModEx and Seed-Detective: Two novel techniques for high quality clustering by using good initial seeds in K-Means

Md Anisur Rahman, Md Zahidul Islam *, Terry Bossomaier

Centre for Research in Complex Systems, School of Computing and Mathematics, Charles Sturt University, Australia

Received 19 November 2013; revised 21 February 2014; accepted 15 April 2014
Available online 23 March 2015

KEYWORDS

Clustering;
Classification;
K-Means;
Cluster evaluation;
Data mining

Abstract In this paper we present two clustering techniques called ModEx and Seed-Detective. ModEx is a modified version of an existing clustering technique called Ex-Detective. It addresses some limitations of Ex-Detective. Seed-Detective is a combination of ModEx and Simple K-Means. Seed-Detective uses ModEx to produce a set of high quality initial seeds that are then given as input to K-Means for producing the final clusters. The high quality initial seeds are expected to produce high quality clusters through K-Means. The performances of Seed-Detective and ModEx are compared with the performances of Ex-Detective, PAM, Simple K-Means (SK), Basic Farthest Point Heuristic (BFPH) and New Farthest Point Heuristic (NFPH). We use three cluster evaluation criteria namely *F*-measure, Entropy and Purity and four natural datasets that we obtain from the UCI Machine learning repository. In the datasets our proposed techniques perform better than the existing techniques in terms of *F*-measure, Entropy and Purity. The sign test results suggest a statistical significance of the superiority of Seed-Detective (and ModEx) over the existing techniques.

1. Introduction

Clustering is a process of partitioning similar records in one cluster and dissimilar records in different clusters. It helps in decision making processes by extracting hidden patterns from a large amount of data. Therefore, it is important to produce good quality clusters from a dataset.

K-Means is a very commonly used clustering algorithm. It requires a user input on the number of clusters. It then randomly selects the same number of initial seeds (i.e. records representing the centers of the clusters) as the user defined number of clusters (Ahmad and Dey, 2007; Bai et al., 2011; Huang, 1997; Khan, 2012; Tan et al., 2005). Each record is

* Corresponding author. Tel.: +61 2 63384214; fax: +61 2 6338 4649.
E-mail addresses: arahman@csu.edu.au (M.A. Rahman), zislam@csu.edu.au (M.Z. Islam), tbossomaier@csu.edu.au (T. Bossomaier).
URL: http://csusap.csu.edu.au/~zislam/ (M.Z. Islam).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2014.04.002

then assigned to the initial seed that has the minimum distance (out of all initial seeds) with the record. Thus the records are initially partitioned into a number of groups. Once the records are assigned to the seeds a new center point for each partition is again computed.

Using the new center points the records are again partitioned. The process of computing new center points and record partitioning continues until a termination condition is met.

Due to the simplicity of K-Means it is a commonly used clustering technique. However, for a user it is difficult to guess and provide the correct number of clusters of a dataset (Chuan Tan et al., 2011; Jain, 2010). Additionally, because of the randomness used in the initial seed selection, K-Means may select poor quality initial seeds resulting in poor quality clusters produced from a dataset (Bagirov, 2008; Bai et al., 2011; Maitra et al., 2010).

Therefore in this study we present a clustering technique called Seed-Detective, which obtains the number of clusters and a set of high quality initial seeds automatically by using a deterministic process. The high quality seeds are then input as the initial seeds of K-Means in order to produce the final clusters of a dataset. The high quality initial seeds are expected to produce high quality clusters through K-Means. Since Seed-Detective uses the deterministic initial seeds it also avoids the randomness of K-Means.

Moreover, we note that Seed-Detective discovers the high quality initial seeds by using another clustering technique called ModEx, which is also proposed in this paper. ModEx is a modified version of an existing clustering technique called Ex-Detective (Islam, 2008; Islam and Brankovic, 2011). ModEx addresses some limitations of Ex-Detective. While ModEx improves the clustering quality of Ex-Detective the application of ModEx in finding the high quality initial seeds for Seed-Detective improves the clustering quality achieved by Seed-Detective.

We compare the performance of Seed-Detective with ModEx, PAM (Han and Kamber, 2006), Simple K-Means (Huang, 1997; Tan et al., 2005), Basic Farthest Point Heuristic (BFPH) (He, 2006) and New Farthest Point Heuristic (NFPH) (He, 2006). We compare the performance of the techniques through three cluster evaluation criteria namely *F*-measure, Entropy and Purity by using four natural datasets that we obtain from the UCI machine learning repository (Bache and Lichman, 2013). The performance of ModEx is also compared with the performance of Ex-Detective, PAM (Han and Kamber, 2006), Simple K-Means (SK) (Huang, 1997; Tan et al., 2005), Basic Farthest Point Heuristic (BFPH) (He, 2006) and New Farthest Point Heuristic (NFPH) (He, 2006).

Please note that decision trees (like many other algorithms such as ANN) are typically used for the classification task where the records of a dataset are classified according to the values of the class attribute (also known as labels or class values) of the dataset (Han and Kamber, 2006; Islam, 2012). The datasets on which a classifier (like a decision tree (DT)) is applied need to have a class attribute. An example of a class attribute can be the attribute on "disease diagnosis" in a patient dataset. However, the datasets on which a clustering algorithm is applied do not need to have a class attribute. Clustering algorithms aim to group the records into clusters. Following the clustering, class values are typically assigned to the records. Once class values are assigned to the records a classifier can be built from the records in order to learn the patterns (such as logic rules) and predict the class values of the future records that do not have the labels.

Although decision trees are generally used for the classification task, some existing techniques called Detective and Ex-Detective (Islam and Brankovic, 2011) use decision trees for clustering. For example, Detective uses decision trees for finding similarities among the values of a categorical attribute and Ex-Detective uses decision trees for finding similar records. Following the existing approaches, our proposed techniques also use decision trees for clustering.

Fig. 1 shows a decision tree (DT) that considers the "Occupation" attribute of a synthetic dataset (the dataset is only used in Section 2.2 for the demonstration purpose and not used in the experiments of this study) as the class attribute. A decision tree is made of a number of nodes (the rectangles in the Fig. 1) and leaves (the ovals in the figure), where a node tests an attribute and divides the dataset into mutually exclusive horizontal segments based on the values of the attribute tested at the node (Islam, 2012; Quinlan, 1993, 1996). For example, the tree in Fig. 1 tests the attribute Qualification at the root node (the node at Level-0) and divides the dataset into segments. In the left most segment all records have PhD as the value for the Qualification attribute. A leaf represents a set of records (i.e. a horizontal segment) where all records ideally have the same class value. If all records of a leaf have the same class value then the leaf is called "homogeneous" and otherwise it is called "heterogeneous". For example, Leaf 1 is a heterogeneous leaf where there are altogether four records, three of which has "Acad" (a short form of Academics) as the class value and the remaining one has "Engr" as the class value. There is a logic rule for each leaf showing the preconditions and the classification of the records belonging to the rule i.e. satisfying the precondition/s of the rule. For example, the logic rule of Leaf 1 is *if Qualification = PhD Occupation = Acad (4:1)*.

The main contributions of the paper are as follows. We propose some modifications of an existing clustering technique called Ex-Detective. We call the modified version as ModEx. We also propose another clustering technique called Seed-Detective that uses ModEx in order to find a set of high quality initial seeds and then feeds them into the traditional K-Means in order to discover high quality clusters. We also implement both of our proposed techniques and a few existing techniques.

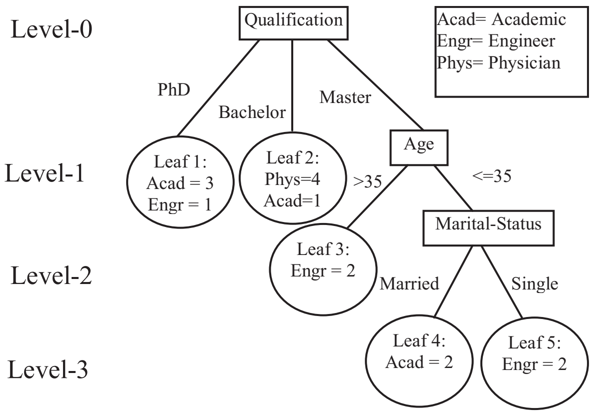


Figure 1 A DT considering Occupation as the class attribute.

The experimental results are presented on four datasets indicating a clear superiority of our techniques over the existing ones.

The structure of the paper is as follows. In Section 2 we discuss some background study related to our proposed techniques. In Section 3 we present our proposed techniques ModEx and Seed-Detective. The experimental results and discussion are presented in Section 4. The conclusion of the paper is presented in Section 5.

2. Background study

In this study we discuss some basic properties of a dataset. We also discuss Ex-Detective since it is used in one of our proposed techniques called ModEx.

2.1. Description of a dataset

We consider that a dataset D has n number of records $D = \{R_1, R_2 \ldots, R_n\}$, and m number of attributes $A = \{A_1, A_2 \ldots, A_m\}$. The attributes of a dataset can be categorical and/or numerical. We present a toy dataset in Table 1 which has 15 records and five attributes “Age”, “Marital-Status”, “Qualification”, “Occupation” and “Professional-Training”. “Age” is a numerical attribute and the others are categorical attributes.

The domain values of “Marital-Status”, “Qualification”, “Occupation” and “Professional-Training” are {Single, Married}, {PhD, Master, Bachelor}, {Academic, Engineer, Physician} and {Yes, No}, respectively. The upper and lower limit values of the numerical attribute “Age” are 65 and 30, respectively. Therefore, the domain of “Age” is [30, 65]. R_i represents the ith record of a dataset and R_{ij} denotes the jth attribute value of the ith record. For example, $R_{5,3}$ represents PhD, which is the value of the 3rd attribute (i.e. Qualification) of the 5th record R_5.

2.2. Description of Ex-Detective

Ex-Detective is a decision tree based clustering technique (Islam, 2008; Islam and Brankovic, 2011). It is an extended version of Detective (Islam, 2008; Islam and Brankovic, 2005). The main steps of Ex-Detective are as follows.

Step 1: Build a decision tree for each categorical attribute.
Step 2: Find the intersections of the leaves.
Step 3: Perform K-Means.

2.2.1. Step 1: Build a decision tree for each categorical attribute

Ex-Detective builds a decision tree (DT) for each categorical attribute separately, considering each categorical attribute as the class attribute. It uses an existing decision tree algorithm such as C4.5 to build the decision trees (Quinlan, 1993, 1996). In order to demonstrate the step, we present two decision trees in Fig. 1 and Fig. 2 considering Occupation and Qualification as the class attributes, respectively. The depth of the decision trees are 4 (Level-0 to Level-3) and 3 (Level-0 to Level-2), respectively. In Fig. 1, the sets of records $\{R_1, R_5, R_7, R_{11}\}$, $\{R_4, R_6, R_8, R_{13}, R_{14}\}$, $\{R_3, R_{10}\}$, $\{R_{12}, R_{15}\}$ and $\{R_2, R_9\}$ belong to Leaf 1, Leaf 2, Leaf 3, Leaf 4 and Leaf 5, respectively. Similarly, in Fig. 2, the sets of records $\{R_4, R_6, R_8, R_{14}\}$, $\{R_2, R_3, R_9, R_{10}, R_{11}\}$, $\{R_1, R_5, R_7\}$ and $\{R_{12}, R_1 3, R_{15}\}$ belong to Leaf 6, Leaf 7, Leaf 8 and Leaf 9, respectively.

In Ex-Detective, a data miner can assign different weights (level of importance) on different categorical attributes instead of considering all categorical attributes equally important for

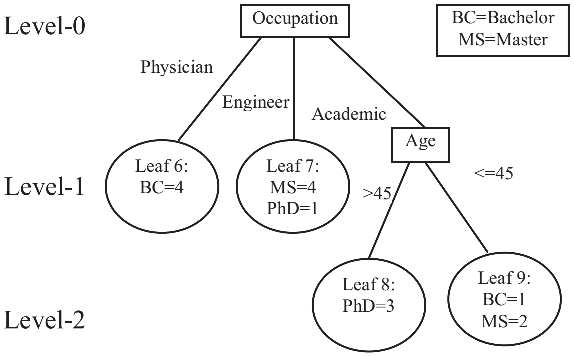


Figure 2 A DT considering Qualification as the class attribute.

Table 1 A toy dataset.

Record	Age	Marital – Status	Qualification	Occupation	Professional – Training
R_1	65	Married	PhD	Academic	No
R_2	30	Single	Master	Engineer	No
R_3	45	Married	Master	Engineer	No
R_4	30	Single	Bachelor	Physician	Yes
R_5	55	Married	PhD	Academic	No
R_6	35	Single	Bachelor	Physician	Yes
R_7	60	Married	PhD	Academic	No
R_8	45	Single	Bachelor	Physician	Yes
R_9	35	Single	Master	Engineer	Yes
R_{10}	42	Married	Master	Engineer	No
R_{11}	32	Single	PhD	Engineer	No
R_{12}	35	Married	Master	Academic	No
R_{13}	45	Single	Bachelor	Academic	Yes
R_{14}	35	Married	Bachelor	Physician	No
R_{15}	35	Married	Master	Academic	Yes

clustering. The weight of a categorical attribute A_i is used for pruning the decision tree T_i that considers the attribute A_i as the class attribute. If the weight of a categorical attribute is w_i and the depth of the decision tree T_i is t_i then the depth (t'_i) of the pruned tree T'_i is $t'_i = t_i * w_i$. The weight w_i can vary from 0 to 1. If the weight w_i of a categorical attribute A_i is 1 then there is no pruning of T_i i.e. the decision tree T_i remains as it is. However, if the weight w_i of the categorical attribute A_i is zero then Ex-Detective performs the maximum pruning for T_i, where T_i contains only one leaf node. In that case all records of the dataset will belong to the leaf node.

We now explain the pruning process of Ex-Detective with examples. Suppose, a data miner assigns weights $w_3 = 0.6$ and $w_4 = 0.5$ on the attributes Qualification (which is the 3rd attribute in Table 1) and Occupation (which is the 4th attribute in Table 1), respectively. Therefore, the tree shown in Fig. 1 is T_4 and the tree shown in Fig. 2 is T_3. The depths of the trees T_4 and T_3 are $t_4 = 4$ and $t_3 = 3$, respectively. The depths after pruning are $t'_4 = t_4 * w_4 = 4 * 0.5 = 2$ and $t'_3 = t_3 * w_3 = 3 * 0.6 = 1.8 \cong 2$, respectively. The pruned tree T'_4 of T_4 is presented in Fig. 3. Similarly, the pruned tree T'_3 of T_3 is presented in Fig. 4. In Fig. 3 the sets of records $\{R_1, R_5, R_7, R_{11}\}$, $\{R_4, R_6, R_8, R_{13}, R_{14}\}$ and $\{R_2, R_3, R_9, R_{10}, R_{12}, R_{15}\}$ belong to Leaf 10, Leaf 11 and Leaf 12, respectively. Similarly, in Fig. 4 the sets of records $\{R_4, R_6, R_8, R_{14}\}$, $\{R_2, R_3, R_9, R_{10}, R_{11}\}$ and $\{R_1, R_5, R_7, R_7, R_{12}, R_{13}, R_{15}\}$ belong to Leaf 13, Leaf 14 and Leaf 15, respectively.

Ex-Detective builds and prunes a set of decision trees considering each categorical attribute as the class attribute one by one. Let us consider that the first z attributes of the set of attributes $A = \{A_1, A_2 \ldots A_m\}$ are categorical attributes and the remaining attributes are numerical. That is, the set of categorical attributes is $A_c = \{A_1, A_2, \ldots A_z\}$. For the z number of categorical attributes $A_c = \{A_1, A_2, \ldots A_z\}$, Ex-Detective builds z. number of decision trees $T_c = \{T_1, T_2 \ldots T_z\}$, where the decision tree T_i is built considering A_i as the class attribute. If the weights on the categorical attributes $w_c = \{w_1, w_2, \ldots w_z\}$, Ex-Detective prunes all the trees in T_c based on the weights in w_c. The pruning process generates z number of pruned trees $T'_c = \{T'_1, T'_2, \ldots T'_z\}$.

2.2.2. Step 2: Find the intersections of the leaves

Ex-Detective next performs intersections among the record that belong to the leaves of the decision trees. If p is the number of leaves in T'_i and q is the number of leaves in T'_j then there will be $p * q$ number of intersections from the trees T'_i andT'_j. For example, the total number of intersections obtained from T'_4 (see Fig. 3) and T'_3 (see Fig. 4) is nine, since the number of leaves in each tree equal to three.

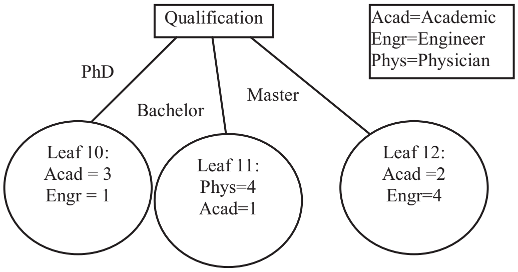


Figure 3 Pruned tree on Occupation attribute.

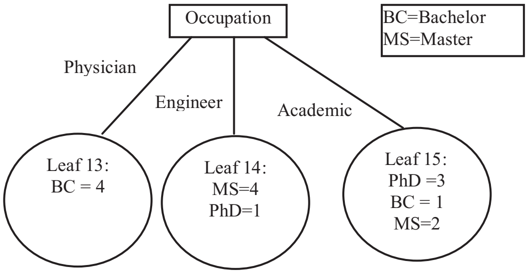


Figure 4 Pruned tree on Qualification attribute.

We now explain the intersection process of Ex-Detective with T'_4 and T'_3. The records that belong to Leaf 10 of T'_4 (see Fig. 3) can intersect with the records that belong to Leaf 13, Leaf 14 and Leaf 15 of T'_3 (see Fig. 4). The intersection between the records that belong to Leaf 10 and Leaf 15 produces a new set of records as denoted by Leaf 10 ∩ Leaf 15 = $\{R_1, R_5 , R_7\}$. Ex-Detective considers the set of records produced by the intersection of two leaves as a preliminary cluster. For example, the set of records $\{R_1, R_5, R_7\}$ is considered as a preliminary cluster. The records that belong to Leaf 11 can intersect with the records that belong to Leaf 13, Leaf 14 and Leaf 15. Similarly, the records that belong to Leaf 12 can intersect with the records that belong to Leaf 13, Leaf 14 and Leaf 15.

The intersection operation is carried out among all the records that belong to the leaves of all decision trees to produce a set of preliminary clusters. For example, if we have z number of pruned trees $T'_c = \{T'_1, T'_2, \ldots T'_z\}$, where a tree T'_i has l_i number of leaves then we get a total of $\prod_i^z l_i$ number of possible intersections. The records within each intersection are considered as a preliminary cluster. Ex-Detective next applies K-Means on each preliminary cluster to produce the final clusters.

2.2.3. Step 3: Perform K-Means

If there is any numerical attribute in a dataset, Ex-Detective performs K-Means (Huang, 1997; Tan et al., 2005) on the records belonging to a preliminary cluster obtained in Step 2. During the application of K-Means only numerical attributes values are taken into consideration. However, the original studies (Islam, 2008; Islam and Brankovic, 2011) did not clearly discuss the process of defining the number of clusters for K-Means that is being applied on the records of a preliminary cluster.

K-Means continues until the termination conditions are satisfied. There are two termination conditions in K-Means. The first termination condition is that the absolute difference between the values of the objective function in two consecutive iterations of K-Means is less than a user defined threshold (ε). A user defined maximum number of iterations are considered as the second termination condition.

3. Our proposed clustering techniques

3.1. The proposed Modified Ex-Detective (ModEx)

We now discuss some issues related to Ex-Detective and then propose some modifications as follows.

3.1.1. Some limitations of Ex-Detective

In the original studies (Islam, 2008; Islam and Brankovic, 2011), the numerical attribute values are not normalized before the application of K-Means. The normalization of the numerical attribute values has a great effect in K-Means (Kim et al., 2006; Visalakshi and Thangavel, 2009). Let us explain the effect of normalization with three records (R_1, R_2 and R_3) and two numerical attributes (Salary and Age) as shown in Table 2 . In R_1 and R_2 the values of Salary are the same (50,000) but in R_2 the value of Age is two times bigger than the value of Age in R_1. Moreover, in R_1 and R_3 the values of Age is equal (25) and in R_3 the value of Salary is slightly bigger than the value of Salary for R_1.

The city block distance (Malik and Baharudin, 2013) between R_1 and R_2 is 25 and the distance between R_1 and R_3 is 1000. Note that, although the value of Age in R_2 is two times bigger than the value of Age in R_1 the distance between them is smaller than the distance between $R_1$and R_3. This is due to the difference of domain sizes of the two attributes Salary and Age. Since the domain size of Salary is very large a slight change in the attribute results in a huge distance, while a big change in Age does not cause a big distance. Therefore, it is important to normalize the numerical attribute values before applying K-Means (Kim et al., 2006; Visalakshi and Thangavel, 2009).

We also observe that after the intersection process Ex-Detective can produce a big number of small sized clusters. The records belonging to a small sized cluster may not have enough support to be considered as an important cluster and may not reveal an interesting pattern to a data miner. The records that belong to a small sized cluster also have a tendency to be merged with other suitable clusters (Maqbool and Babri, 2006; Noordam et al., 2002) indicating their limitation in revealing interesting patterns.

The original studies (Islam, 2008; Islam and Brankovic, 2011) also did not clearly discuss the process of defining the number of clusters for K-Means that is being applied on the records of a preliminary cluster. However, this is an important requirement for the operation of a clustering technique.

3.1.2. Modifications

To address the limitations of Ex-Detective, we propose the following three modifications.

3.1.2.1. Modification 1: Normalization of numerical attributes. We propose that numerical attribute values belonging to an attribute should be normalized within the range of 0–1. The normalization aims to give the same emphasize on each numerical attribute regardless of their actual domain sizes. The normalization needs to be carried out before the K-Means algorithm is applied on the numerical attributes. If R_{ij} is the jth attribute value of the ith record and *min* and *max* are the minimum and maximum domain values of the jth attribute; then the normalized value R'_{ij} is calculated as follows. Normalization of each attribute is carried out considering its min and max values. Two different attributes use two different pairs of min and max values for the normalization of the values belonging to the attributes.

$$R'_{ij} = \frac{R_{ij} - min}{max - min} \qquad (1)$$

Table 2 A dataset for normalization.

Record	Salary	Age
R_1	50,000	25
R_2	50,000	50
R_3	51,000	25

3.1.2.2. Modification 2: Merging. We realize that Ex-Detective is likely to produce some extremely small sized preliminary clusters. Therefore, we propose to merge the records of a small sized preliminary cluster (obtained in Step 2 of Section 2.2) with other suitable clusters. For merging, first we find the smallest cluster from the set of all preliminary clusters produced in Step 2. If the number of records in the smallest cluster is less than a user defined threshold θ then we merge all the records of the smallest cluster with other suitable clusters as follows.

A record R_i belonging to the smallest cluster C_i is merged with another cluster C_k, where R_i has the minimum distance with the seed S_k of C_k. Let R_i be the record of the smallest cluster, S_k be the seed/center of the kth cluster (C_k) then we merge R_i with C_k, if $dist(R_i, S_k) < dist(R_i, S_l); \forall l \neq k$. Therefore, all records of the smallest cluster are reassigned one by one to other suitable clusters.

We then again find the smallest cluster that has a number of records less than θ and similarly merge the records with other suitable clusters. We continue the process of finding the smallest cluster and merging the records until the number records in the smallest cluster is greater than or equal to θ.

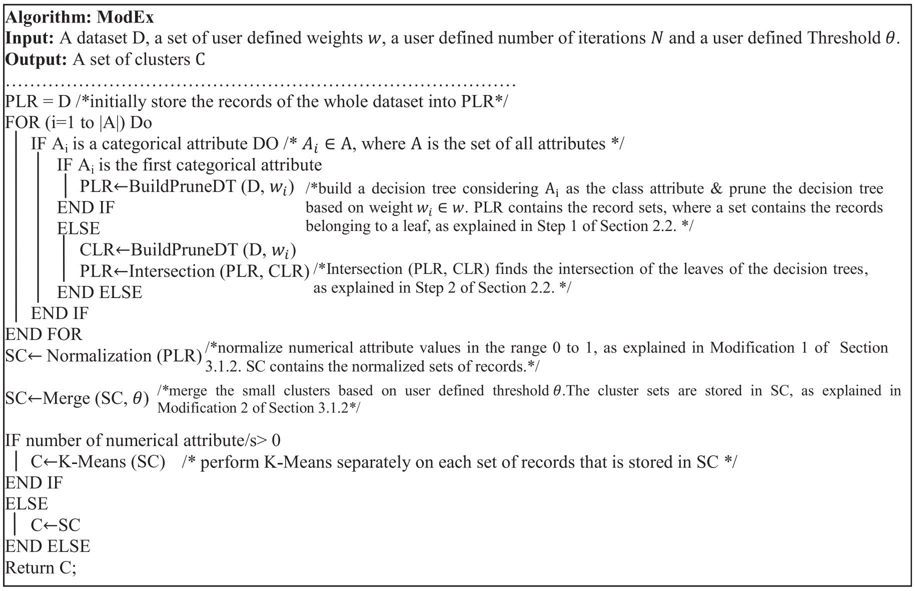
3.1.2.3. Modification 3: Number of clusters. In ModEx, the number of clusters for K-Means is considered to be $log_{10}|\mathrm{D}|$, where $|\mathrm{D}|$ is the number of records on which K-Means is applied. In this case, $|\mathrm{D}|$ is the number of records that belong to a preliminary cluster, when K-Means is applied on the preliminary cluster. Note that, we round the fractional value of $log_{10}|\mathrm{D}|$ to its nearest integer value, since the cluster number cannot be fractional.

If the number of cluster is greater than or equal to 2 we then apply K-Means otherwise we do not apply K-Means on the preliminary cluster. If the value of $log_{10}|\mathrm{D}|$ is less than 1.5 we then produce a single cluster since K-Means is not applied. However, if the value of $log_{10}|\mathrm{D}|$ is greater than 1.5 then it is rounded to its nearest integer number. For example, the value of $log_{10}32$ is 1.5051 and we round the fractional value to 2. Similarly, the rounded value of $log_{10}317$ is 3. In Fig. 5 we present the algorithm of ModEx.

3.2. Seed-Detective: A proposed clustering technique

3.2.1. Motivation behind Seed-Detective

In this section we discuss some limitations of K-Means and ModEx. In K-Means, the number of clusters (k) need to be provided by a user. However, from a user point of view often it is difficult to estimate the proper number of clusters of a dataset (Chuan Tan et al., 2011; Jain, 2010). Additionally, K-Means may select poor quality initial seeds because of its random seed selection criteria. The poor quality initial seeds may produce the poor quality clusters from a dataset (Bagirov, 2008; Bai et al., 2011; Maitra et al., 2010).



```
Algorithm: ModEx
Input: A dataset D, a set of user defined weights $w$, a user defined number of iterations $N$ and a user defined Threshold $\theta$.
Output: A set of clusters C
..................................................................................
PLR = D /*initially store the records of the whole dataset into PLR*/
FOR (i=1 to |A|) Do
    IF $A_i$ is a categorical attribute DO /* $A_i \in$ A, where A is the set of all attributes */
        IF $A_i$ is the first categorical attribute
            PLR←BuildPruneDT (D, $w_i$)   /*build a decision tree considering $A_i$ as the class attribute & prune the decision tree
        END IF                            based on weight $w_i \in w$. PLR contains the record sets, where a set contains the records
        ELSE                              belonging to a leaf, as explained in Step 1 of Section 2.2. */
            CLR←BuildPruneDT (D, $w_i$)
            PLR←Intersection (PLR, CLR) /*Intersection (PLR, CLR) finds the intersection of the leaves of the decision trees,
        END ELSE                         as explained in Step 2 of Section 2.2. */
    END IF
END FOR
SC← Normalization (PLR) /*normalize numerical attribute values in the range 0 to 1, as explained in Modification 1 of Section
                         3.1.2. SC contains the normalized sets of records.*/
SC←Merge (SC, $\theta$) /*merge the small clusters based on user defined threshold $\theta$.The cluster sets are stored in SC, as explained in
                         Modification 2 of Section 3.1.2*/

IF number of numerical attribute/s> 0
    C←K-Means (SC)   /* perform K-Means separately on each set of records that is stored in SC */
END IF
ELSE
    C←SC
END ELSE
Return C;
```

Figure 5 The algorithm of ModEx.

We also discuss two main limitations of ModEx as follows. In ModEx, a record that belongs to a preliminary cluster does not have any chance to move into another preliminary cluster since K-Means is applied on the records of each preliminary cluster separately. Therefore, even if it was better to assign a record in a different preliminary cluster ModEx is unable to do that once a record is allocated to a preliminary cluster. Hence, the application of K-Means on a preliminary cluster may not be able to bring the full advantage of K-Means. The second limitation of ModEx is that during the application of K-Means the initial seeds are selected randomly. This may again lead to the selection of poor quality seeds within a preliminary cluster. Moreover, due to the random selection of the seeds different runs of ModEx may produce different clustering results.

We present Seed-Detective in order to address the above issues of K-Means and ModEx and thereby obtain a set of good quality clusters.

3.2.2. Seed-Detective: The proposed clustering technique

Seed-Detective is a combination of a modified version of ModEx and Simple K-Means (Huang, 1997; Tan et al., 2005). The main advantages of Seed-Detective are as follows.

- Seed-Detective automatically obtains the number of clusters from a dataset through a deterministic process.
- It produces good quality initial seeds which are then fed into K-Means in order to produce good clusters.
- Unlike Ex-Detective, ModEx and K-Means, Seed-Detective avoids the randomness by using deterministic initial seeds for K-Means to produce final clusters.

Seed-Detective obtains the number of clusters and high quality initial seeds from a dataset by using a modified version of ModEx. It then provides the high quality initial seeds to K-Means to produce the final clusters of a dataset. The high quality initial seeds are expected to produce high quality clusters through K-Means. Unlike ModEx, in Seed-Detective K-Means is applied on a whole dataset instead of applying on the records of each preliminary cluster separately. Since Seed-Detective uses the deterministic initial seeds for K-Means, Seed-Detective avoids the randomness in initial seed selection. The basic steps of Seed-Detective are as follows.

Step 1: Produce a set of preliminary clusters by using a modified version of ModEx.
Step 2: Calculate the seeds of the preliminary clusters.
Step 3: Provide the seeds to K-Means to produce final clusters.

3.2.2.1. Step 1: Produce a set of preliminary clusters by using a modified version of ModEx. In Step 1, a set of preliminary clusters are produced by using a modified version of ModEx. We propose two modifications of ModEx to use it in Seed-Detective for the purpose of initial seed selection. The modifications of ModEx are as follows.

- The first modification allows a data miner to assign weights on numerical attributes, in addition to categorical attributes of a dataset.
- The second modification of ModEx is the exclusion of K-Means since in Seed-Detective K-Means is applied in Step 3.

In Seed-Detective, we build a decision tree (Islam, 2012; Quinlan, 1993, 1996) for each individual attribute both categorical and numerical separately considering the attribute as the class attribute. To build a decision tree considering a numerical attribute as the class attribute we first categorize the values belonging to the attribute into a user defined number of categories (Berzal et al., 2003, 2004). The number of categories of a numerical attribute is considered as the root over of the domain size of the numerical attribute. For example, the domain size of numerical attribute Age is [30, 65]. Therefore, the number of categories of Age is defined as

$\sqrt{35} \cong 6$. The values of Age are then divided into 6 equal sized categories; 30–35, 36–41, 42–47, 48–53, 54–59 and 60–65. The different categories may have different number of records. The categorization is done only for the purpose of building a decision tree, when the class attribute is originally a numerical attribute. All other non-class numerical attributes are considered in their original numerical form.

If a dataset has m number of attributes $A = \{A_1, A_2 \ldots A_m\}$ then Seed-Detective builds m number of decision trees $T = \{T_1, T_2 \ldots T_m\}$, where the decision tree T_i is built considering attribute A_i as the class attribute. If the weights of the attributes are $w = \{w_i, w_2 \ldots w_m\}$ and the depths of trees are $t = \{t_1, t_2 \ldots t_m\}$ then Seed-Detective prunes all the trees in T and produces $T' = \{T'_1,\ T'_2, \ldots T'_m\}$. The depth of the pruned tree T'_i is defined as $t'_i = t_i * w_i$, where w_i is the weight of attribute A_i and t_i is the depth of the tree T_i. The weight w_i of an attribute A_i can vary from 0 to 1. If the weight w_i of an attribute A_i is 1 then there is no pruning of T_ii.e. the pruned tree T'_i is same as T_i. However, if the weight w_i of attribute A_i is zero then Seed-Detective performs the maximum pruning for T_i, i.e. the pruned tree T'_i has only one leaf node, which contains all records of the dataset.

After decision trees are built, Seed-Detective next performs intersection operation among the records that belong to the leaves of the trees and produces a set of preliminary clusters. The intersection operation in Seed-Detective is carried out in the same way as ModEx does. To demonstrate the intersection operation we use a dataset that has five attributes A_1, A_2, A_3, A_4 and A_5. The attributes A_1, A_2 and A_3 are categorical whereas the attributes A_4 and A_5 are numerical. The domain values of attributes A_1, A_2 and A_3 are $\{a_{11}, a_{12}, a_{13}\}, \{a_{21}, a_{22}, a_{23}\}$ and $\{a_{31}, a_{32}\}$, respectively. Let us assume that we build a decision tree considering the attribute A_1 as the class attribute as shown in Fig. 6. The tree has three leaves Leaf 1, Leaf 2 and Leaf 3. Each leaf of the tree contains a set of records that have the same (in case of a homogeneous leaf) or similar (in case of a heterogeneous leaf) class value/s. The records having the same/similar class value/s are considered to be similar due to the representational power of the class values. Typically, a class value has the capacity to represent a record significantly better than any other non-class values. For example, if we consider "Diagnosis" as the class attribute of a patient dataset then the class values "Cancer" and "Fever" for two patients represent the patients very clearly. As soon as we learn the class value to be Cancer for a patient we get an image of the patient. Similarly, by knowing the class value to be Fever for another patient we also get an image of the patient. However, if we learn the values of another attribute say "Age" to be 27 and 34 for the two patients we do not get such a significant information about the patients.

Since the records belonging to a leaf have the same/similar class values they can be considered as similar resulting in a cluster. Records belonging to a leaf also share the same values for all categorical attributes tested in the path from the root to the leaf. The records also share the similar numerical values (that fall in the same range) for all numerical attributes tested in the path. This makes the argument of similarity for the records even stronger.

For example, the records that belong to Leaf 2 in Fig. 6 are considered as a cluster, since they have the same class value a_{13}, $A_4 \leqslant 7$, and $A_3 = a_{31}$. Let us assume that the records that belong to Leaf 1, Leaf 2 and Leaf 3 form clustersC_1, C_2 and C_3, respectively. The clusters C_1, C_2 and C_3 are presented in Fig. 7, where the dots represent the records. Note that in Fig. 7 the records are presented in a two dimensional space just for the purpose of demonstration .The actual dimensions (number of attributes) in this case is five. Additionally the curved lines used for partitioning the records into different clusters are not factual. They are also used for the demonstration purpose only. Similarly, we build another decision tree considering the attribute A_2 as the class attribute as shown in Fig. 8. The records belonging to Leaf 4, Leaf 5 and Leaf 6 form clusters C_4, C_5 and C_6 that we present in Fig. 9. We then intersect the clusters C_1, C_2 and C_3 of Fig. 7 with the clusters C_4, C_5 and C_6 of Fig. 9 in the same way as ModEx does. The intersection operation produces another set of preliminary clusters C_7, C_8, C_9, C_{10}, C_{11}, C_{12} and C_13 as shown in Fig. 10.

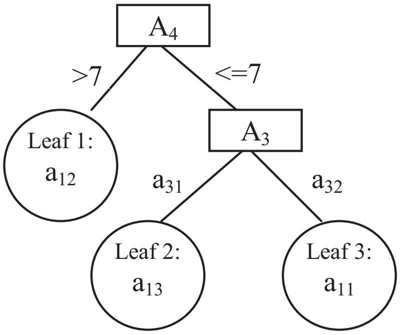


Figure 6 DT on attribute A_1.

We observe that the intersection operation of Seed-Detective is likely to produce too many small clusters. For example, C_{12} in Fig. 10. seems to be a small sized cluster. The limitations of having such small clusters have been discussed in Section 3.1.1. Therefore, in Seed-Detective, we merge the records that belong to a small cluster with other suitable clusters. The records belonging to a small cluster are merged with other suitable clusters by using the process as discussed in Modification 2 of ModEx.

During merging operation we need to calculate the distance between a record of a small cluster and the seeds of other clusters. In distance calculation, we use normalized values of a numerical attribute. We normalize the numerical attribute values belonging to an attribute within the range of 0–1 by using the process discussed in Modification 1 of ModEx. However, for a categorical attribute during distance calculation 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 and otherwise 0 (Huang, 1997). The main purpose of the normalization of

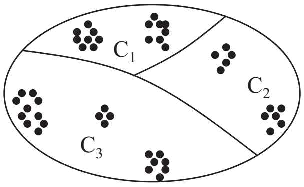


Figure 7 The clusters based on attribute A_1.

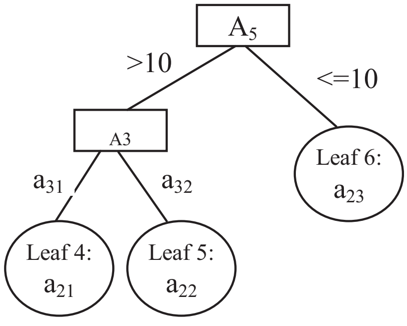


Figure 8 DT on attribute A_2.

numerical attribute is to give equal emphasis on all attributes as discussed in Modification 1 of ModEx.

3.2.2.2. Step 2: Calculate the seeds of the preliminary clusters. The seeds of the preliminary clusters are calculated in Step 2. The value of a numerical attribute in a seed is the average value of the attribute for all records belonging to the preliminary cluster. For a categorical attribute, the value having the maximum frequency among the records of a preliminary cluster is considered as the seed value for the categorical attribute.

The aim for initial seed selection is to minimize the sum of the squared error (SSE) of the distances between the initial seed of a preliminary cluster and the records belonging to the same preliminary cluster. Therefore, we calculate the seed of a preliminary cluster by taking the average of a numerical attribute and the value having the highest frequency for a categorical attribute. K-Means also computes the seed of a cluster in the same way. The aim of the seed selection in our approach is similar to the objective function used in K-Means. Let C_i be the ith cluster, S_i be the seed of the ith cluster, k be the number of clusters, R_j^i be the jth record belonging to the ith cluster, and dist (R_j^i, S_i) is the distance between R_j^i and S_i. The objective function of K-Means is as follows.

$$\mathrm{SSE} = \sum_{i=1}^{k}\sum_{j=1}^{|C_i|}\mathrm{dist}(R_j^i, S_i)^2 \tag{2}$$

In Seed-Detective, we first build a decision tree (as mentioned in Step 1). Since a decision tree algorithm such as C4.5 aims to minimize the entropy for the class values within a leaf it attempts to increase the possibility of having the same class value among all records belonging to a leaf. Let, H be the entropy of the class values in a leaf, $|q|$ be the domain size of the class attribute, $p(q_l)$ be the probability of the lth class value in the leaf. H can be represented as follows.

$$H = -\sum_{l=1}^{|q|}p(q_l)log_2p(q_l) \tag{3}$$

Therefore, it is clear that in order to minimize H we need to maximize the proportion of a class value q_1. When H is equal to zero then all records of a leaf have the same class value. Therefore, the seed of a preliminary cluster also have the same or similar values for the significant attributes since all significant attributes are considered as the class attribute one by one. Since we categorize a numerical significant attribute, all records belonging to a cluster have values (for the attribute) belonging to the same or similar category.

3.2.2.3. Step 3: Provide the seeds to K-Means to produce final clusters. In Step 3, we first normalize the dataset by using the process discussed in Modification 1 of ModEx. We then provide the seeds of the preliminary clusters as initial seeds to K-Means to produce the final clusters. For K-Means, we consider the number of clusters is equal to the number of initial seeds. K-Means continues until the termination conditions are satisfied. There are two termination conditions in K-Means. The first termination condition is that the absolute difference between the values of the objective function in two consecutive iterations of K-Means is less than a user defined threshold (ε). A user defined maximum number of iterations are considered as the second termination condition. If k is the number of clusters; R_j^i is the jth record of the ith cluster (C_i) and S_i is the seed of the ith cluster then the objective function of K-Means is as follows.

$$\mathrm{SSE} = \sum_{i=1}^{k}\sum_{j=1}^{|C_i|}\mathrm{dist}(R_j^i, S_i)^2 \tag{4}$$

During the application of K-Means, the distance between a record and the seed of a cluster can be calculated in one of the two different ways. The first way is the conventional way where the distance is calculated based on all attributes. In the second way the distance can be calculated based on only the significant attributes using their level of significance as follows.

$$dist(R_j, S_i) = \frac{\sum_{\mathrm{a}=1}^{|A_r|} w_a|R_{j,a} - S_{i,a}| + \sum_{a=|A_r|+1}^{m}\mathrm{dist}(R_{j,a}, S_{i,a})}{\sum_{a=1}^{m} w_a} \tag{5}$$

Here $R_{j,a}$ is the ath attribute value of jth record, $S_{i,a}$ is the ath attribute in the seed of ith cluster, w_a is the user defined weight (significance level) for ath attribute, $|A_r|$ is the number of numerical attributes and m is the total number of attributes in a dataset.

Therefore, the attributes having zero weight are ignored in the K-Means process. Moreover, the attributes having high weights have more influence in distance calculation than the attributes having low weights. Note that attribute weights used

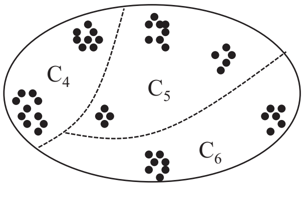


Figure 9 The clusters based on A_2.

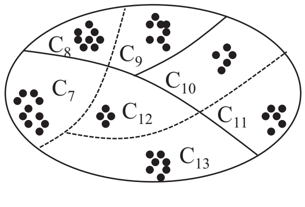


Figure 10 The preliminary clusters based on A_1 and A_2.

in distance calculation in Eq. (5) can be different from the attribute weights used in initial seed selection. For initial seed selection a user may want to assign non-zero weights on low number of attributes in order to reduce the complexity while getting high quality seeds. However, for distance calculation purpose a user may want to assign non-zero weights on many important attributes. We consider in this approach that a user knows his dataset well and therefore, can identify the important attributes and guess appropriate weights. If a user does not know the important attributes he/she can either try different weight sets and explore the results or use the weight 1 for all attributes. In Seed-Detective, after application of K-Means we obtain the final clusters of a dataset. We then de-normalize the records that belong to a cluster to obtain original records in a cluster. For the records that belong to a cluster we find the indexes (position of a record in a dataset) of the records. Based on the indexes of the records we then collect the records from the original dataset. In order to give better idea on Seed-Detective, we present an algorithm of Seed-Detective in Fig. 11.

3.3. Analysis on ModEx and Seed-Detective

In ModEx, a record that belongs to a preliminary cluster does not have any opportunity to move into another preliminary cluster even if it suits better with another cluster. Each preliminary cluster only gets divided into sub clusters due to the application of K-Means within each cluster separately. However, in Seed-Detective the preliminary clusters are only used to obtain the initial seeds for K-Means, therefore a record can move into any cluster based on the distance between the record and the seed of a cluster. For example, if a record R_j has minimum distance with seed S_i of cluster C_i then the record will be assigned to cluster C_i. During various iterations of K-Means R_j can

```
Algorithm: Seed-Detective
Input: A dataset D, a set of user defined weights $w$, a user defined number of iterations $N$ and a user defined Threshold $\theta$
Output: A set of clusters C
………………………………………………………………………………
/*Step 1: Produce a set of preliminary clusters*/
    FOR (i=1 to |A|) Do
        IF $A_i$ is a numerical attribute DO /* $A_i \in A$, where $A$ is the set of all attributes */
            $D_i \leftarrow$ Categorization (D, $\mathrm{A_i}$) /*categorize the values of a numerical attribute to a build decision tree
        End IF                                  considering the attribute as the class attribute*/
        ELSE
            $D_i \leftarrow$ D
        END ELSE
        IF $i = 1$ DO                          /*build a decision tree considering $\mathrm{A_i}$ as the class attribute & prune the
            PLR←BuildPruneDT($D_i$, $\mathrm{A_i}$, $w_i$) decision tree based on weight $w_i \in w$. PLR contains the record sets, where
        END IF                                  a set contains the records belonging to a leaf */
        ELSE
            CLR←BuildPruneDT($D_i$, $\mathrm{A_i}$, $w_i$)
            PLR←Intersection (PLR, CLR) /*Intersection (PLR, CLR) finds the intersection of the leaves of the
        END ELSE                                decision trees*/
    END FOR
    SC← Normalization (PLR) /*normalize numerical attribute values in the range 0 to 1*/
    SC← Merge (SC,$\theta$) /*merge the small clusters based on user defined threshold $\theta$ and after merging the clusters are stored in SC*/
/*Step 2: Calculate the seeds of the preliminary clusters */
    S ← SeedOfPreCluster (SC) /*calculate the seeds of the preliminary clusters, S is a set of seeds */
/*Step 3: Give input the seeds (S) to K-Means to produce final clusters */
    $D'$ ← Normalization (D) /*normalize numerical attribute values in the range 0 to 1, $D'$ contains normalize dataset*/
    Set $\mathrm{O_{cur}}$ ←0, $\mathrm{O_{prev}}$ ←0 /*$\mathrm{O_{cur}}$ contains current value of the SSE and $\mathrm{O_{prev}}$ contains previous value of the SSE */
    FOR (t= 1 to N) DO   /* t counts the number of iteration*/
        /*Step 3.1: Partitions the records into clusters */
        C←PartitionRecord ($D'$, S) /*C is a set of clusters, where each cluster is a set of records*/
        /* Step 3.2: Seed calculation */
        S ← CalculateSeed (C)
        /* Step 3.3: Calculate the value of objective function (SSE) */
        $\mathrm{O_{cur}}$ ← SSE ($D'$, S) /* calculate sum of square error (SSE)*/
        /* Step 3.4: Check termination conditions */
        IF (t>1 &$|\mathrm{O_{cur}} - \mathrm{O_{prev}}| \le \varepsilon$ ) DO
            Break
        END IF
        $O_{prev}$ ← $\mathrm{O_{cur}}$
    END FOR
    /* Step 3.5: Produce the final clusters */
    C←PartitionRecord ($D'$, S)
    C←Denormalize ($D$, C)
Return C;
```

Figure 11 The algorithm of Seed-Detective.

change its cluster if R_j has minimum distance with the seed of another cluster. It is expected to produce better quality final clusters. Unlike ModEx, Seed-Detective does not have any randomness since it uses deterministic initial seeds for its K-Means operation. The K-Means in ModEx uses random initial seeds within each preliminary cluster.

Another important difference between ModEx and Seed-Detective is that Seed-Detective first categorizes a numerical attribute and then builds decision tree that considers the categorized numerical attribute as the class attribute, whereas ModEx does not categorize (discretize) a numerical attribute and does not build a decision tree for the numerical attribute. As a result, for a dataset that has only numerical attributes ModEx behaves in the same way as K-Means. That is, the initial seeds of ModEx will be randomly chosen, same as K-Means. Note that the main motivation for the development of ModEx was to improve K-Means so that K-Means can handle categorical attributes as well. It was important for K-Means to be able to handle categorical attributes especially when all attributes of a dataset are categorical. On the other hand, Seed-Detective should produce good quality seeds even for a dataset that has all numerical attributes. Of course, the quality of seeds may depend on the quality of categorization. There are many categorization (discretization) techniques available in the literature (Kurgan and Cios, 2004; Yang and Webb, 2009). Any existing good technique can be used along with Seed-Detective and should improve the clustering quality of Seed-Detective. A thorough experimentation on this can be an interesting future work.

4. Experimental results and discussion

We implement Seed-Detective (SD), ModEx, Ex-Detective (ED) (Islam, 2008; Islam and Brankovic, 2011), PAM (Han and Kamber, 2006), Simple K-Means (SK) (Han and Kamber, 2006; Huang, 1997; Tan et al., 2005), Basic Farthest Point Heuristic (BFPH) (He, 2006), and New Farthest Point Heuristic (NFPH) (He, 2006). The performances of the techniques are compared in terms of F-measure, Entropy and Purity (Chuang and Chen, 2004; Tan et al., 2005).

In the experimentation we use four natural datasets that we obtain from the UCI machine learning repository (Bache and Lichman, 2013). In Table 3 we present a brief introduction on the datasets. The CA dataset has 690 records and 15 attributes (where 9 of them are categorical and 6 of them are numerical attributes) excluding the class attribute. In the CA dataset there are records that have some missing values; therefore, from the CA dataset we first remove the records having any missing values. After removing the records the CA dataset has 653 records. The class size of the CA dataset is 2 meaning that the domain size of the class attribute is 2. That is the class attribute has two possible values. Note that we remove the class attribute from a dataset before applying any clustering technique on it, since typically the datasets on which clustering techniques are applied do not have class attribute i.e. labels for the records. The class attribute of a dataset is used again for the purpose of cluster evaluation through the metrics such as F-measure, Purity and Entropy (Chou et al., 2004; Tan et al., 2005).

The Credit Approval (CA) dataset contains information regarding the applications for a credit card. In the dataset, the names of all attributes are changed to some meaningless symbols (such as A_1 and A_2) to protect the confidentially of the dataset (Bache and Lichman, 2013).

The Contraceptive Method Choice (CMC) dataset is a subset of the 1987 National Indonesia Contraceptive Method Prevalence Survey (Bache and Lichman, 2013). It contains the pregnancy information of married women. It is used to predict the contraceptive method (that women use) based on demographic and socio-economic information of the women. In the dataset, there are 1473 records and 9 attributes (where 7 of them are categorical and 2 of them are numerical) excluding the class attribute. The categorical attributes are wife's education, husband's education, wife's religion, wife's now working, husband's occupation, standard-of-living index and media exposure. The numerical attributes are wife's age and number of children ever born. The name of the class attribute is "contraceptive method used". The domain values of the class attribute are no-use, long-term and short-term.

The Statlog Heart (SH) dataset contains information about heart disease. It is used to predict the absence or present of heart disease. It has 270 records and 13 attributes (where 7 of them are categorical and 6 of them are numerical). The names of categorical attributes are sex, chest pain type, fasting blood sugar, resting electrocardiographic results, exercise induced angina, the slope of the peak exercise ST segment and thal. The names of numerical attributes are age, resting blood pressure, serum cholesterol, maximum heart rate achieved, old peak and number of major vessels. The domain values of the class attribute are absence or presence (Bache and Lichman, 2013).

The German Credit Approval (GCA) dataset contains customer information regarding credit card application. It is used to predict a customer either as good or bad. In the dataset, there are 1000 records and 20 attributes, where 13 of them are categorical and 7 of them are numerical. The names of the categorical attributes are status of existing checking account, credit history, purpose, savings account/bonds,

Table 3 A brief introduction to the datasets.

Datasets	Records with any missing values	Records without any missing values	No. of categorical attributes	No. of numerical attributes	Class size
Contraceptive Method Choice (CMC)	1473	1473	7	2	3
Credit Approval (CA)	690	653	9	6	2
German Credit Approval (GCA)	1000	1000	13	7	2
Statlog Heart (SH)	270	270	7	6	2

present employment since, personal status and sex, other debtors/guarantors, property, other installment plans, housing, job, telephone and foreign worker. The names of numerical attributes are duration in month, credit amount, installment rate in percentage of disposable income, present residence since, age in years, number of existing credits at this bank, and number of people being liable to provide maintenance for. The domain values of the class attribute are good or bad (Bache and Lichman, 2013).

In ModEx and Seed-Detective the user defined minimum number of records θ, which is required for merging, is considered to be 10% of all records of a dataset. In the experiments for ModEx, Seed-Detective, Ex-Detective, Simple K-Means, BFPH and NFPH the number of iterations is considered to be 50 and a user defined threshold ε is considered to be 0.005. Note that in Ex-Detective the number of clusters used for the K-Means part of Ex-Detective has not been defined clearly (Islam, 2008; Islam and Brankovic, 2011), but in these experiments we use $log_{10}|\mathrm{D}|$ as the number of clusters. During a distance calculation between two records if the categorical values (of an attribute) belonging to the records are different then the distance between the two records in terms of the attribute is considered to be 1 and otherwise 0 (Huang, 1997).

As mentioned above, the maximum number of iterations of K-Means, I_{max} to be 50 as a termination condition. We run an empirical analysis to justify the selection of $I_{max} = 50$. We run K-Means on the CMC dataset and the CA dataset 50 times with just one termination condition $\varepsilon = 0.005$.

Fig. 12 and Fig. 13 present the frequency versus iteration graphs where the X-axis shows the number of iterations required by K-Means before it is terminated, and the Y-axis shows the number of times (out of 50 runs) K-Means is terminated for a particular X-axis value. For example for the CMC dataset, in 10 out of 50 runs K-Means terminates in 7 iterations.

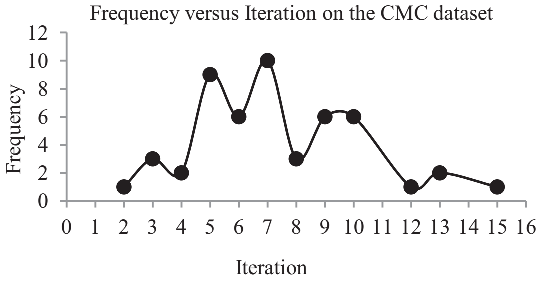


Figure 12 The iteration versus frequency of K-Means on the CMC dataset.

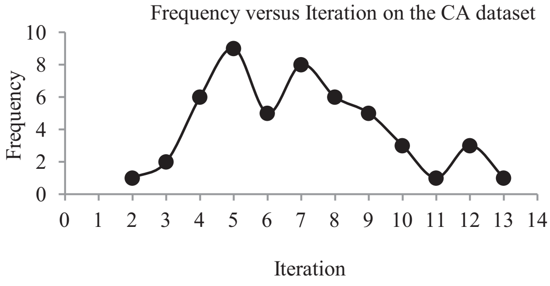


Figure 13 The iteration versus frequency of K-Means on the CA dataset.

For both datasets K-Means terminates well below 50 iterations. Therefore, we consider $I_{max} = 50$ as a safe condition since in that case K-Means will not be terminated prematurely due to the user defined number of iterations. However, $I_{max} = 50$ will terminate K-Means in some rare and unusual cases, where it is not terminated by the $\varepsilon = 0.005$ condition.

The evaluation criteria *F*-measure, Purity and Entropy test the ability of a clustering technique to group the records in such a way so that all records in a group have the same class value i.e. the same value for the class attribute. Therefore, in order to match the evaluation criteria (*F*-measure, Purity and Entropy) it is a sensible approach to consider that a data miner assigns high weights on the attributes that are strongly related to the class attribute, i.e. the attributes that have high influence in classifying the class values.

The influence of an attribute A_i on classifying the class attribute can be measured by entropy of the class values when the dataset is divided into horizontal segments based on the values of A_i, as it is measured during the splitting point selection for building a decision tree (Quinlan, 1993, 1996). Note that if there is a numerical attribute A_i in a dataset, we first categorize the numerical attribute values in the same way as mentioned in Step 1 of Seed-Detective.

The attributes with low entropy are more capable in classifying the class values. Therefore, we assign high weights on the attributes with low entropy and vice versa. It would be unfair to assign high weights on other attributes and test the cluster quality using the evaluation criteria *F*-measure, Entropy and Purity. We argue that if ModEx and Seed-Detective can achieve good *F*-measure, Purity and Entropy values by using suitable weight patterns then it should also achieve good quality clusters (according to the purposes of a data miner) when a data miner assigns a different weight distribution suitable for his/her purpose.

Based on the entropy values of all attributes, we divide the attributes into three categories: best attributes (BA), medium attributes (MA) and worst attributes (WA). In each category the number of attributes is approximately one third of the total attributes in a dataset. In the BA category, we assign weights on the best attributes and in the BM category we assign weights on the best and the medium attributes. For the experimentation purpose we use five different sets of weight patterns in each category. The weight patterns in the BA category are BA1, BA2, BA3, BA4 and BA5.

We now explain the weight patterns of the BA and BM categories by using CMC dataset (see Table 4). We rank the attributes where an attribute that has lower entropy (i.e. a good attribute) given a higher rank (a smaller number) and vice versa.

For each individual weight pattern of the BA and BM category, we produce a set of clusters using ModEx and evaluate them by means of *F*-measure, Entropy and Purity. Similarly, for each individual weight pattern we produce a set of clusters using Ex-Detective and evaluate them. We then compare the performance of ModEx with the performance of Ex-Detective. We also compare the performance of ModEx with SK, PAM, BFPH and NFPH. To compare ModEx with SK, for each individual weight pattern we produce the same number of clusters by SK that ModEx produces. That is if the number of clusters produced by ModEx for a weight pattern

Table 4 The weights patterns to the CMC dataset.

Attribute	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	Weight pattern
Rank	2	1	3	8	7	9	5	4	6	
Weights on best attributes (BA)										
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

is k then we produce k clusters by SK. Similarly, in BFPH and NFPH we produce k clusters.

For a particular number of clusters k, SK, PAM and BFPH randomly select k records as the initial seeds of the clusters. Because of the random seed selection criteria, different runs of SK, PAM and BFPH may produce different clustering results even for the same the number of clusters. Therefore, in these experiments we run SK, PAM and BFPH 10 times for each number of clusters and take the average cluster evaluation results to compare with the results of ModEx. In every run of SK and BFPH we use 50 iterations.

In Table 5, we present the F-measure values of ModEx, Ex-Detective, PAM, SK, BFPH and NFPH to the CA dataset. We present the number of clusters in the 3rd column that uses a "/" sign. The number at the left side of "/" denotes the number of clusters produced by ModEx, whereas the number at the right side of "/" denotes the number of clusters produced by Ex-Detective. We also calculate the average F-measure values of the techniques that we present at the last row of Table 5 A higher F-measure value indicates a better clustering quality. From Table 5 , we see that the average F-measure value of ModEx is better than the average F-measure values of Ex-Detective, SK, BFPH and NFPH. The values in the square brackets (see the last row of Table 5) present the scores of the techniques where the best technique gets a score of 6 and the worst technique gets 1.

We also compute the entropy and purity values of the clusters obtained by the techniques for each weight category. In Table 6 we present the average entropy and purity values for

Table 5 The F-measure values to the CA dataset.

Weight category	Weight pattern	Number of clusters (NoC)	ModEx	Ex-Detective	PAM	SK	BFPH	NFPH
Best attributes (BA)	BA1	3/3	0.7011	0.7011	0.7031	0.7011	0.7056	0.7056
	BA2	5/5	0.7682	0.7682	0.6843	0.6935	0.6992	0.6992
	BA3	5/5	0.7682	0.7682	0.6843	0.6935	0.6992	0.6992
	BA4	10/10	0.7746	0.7746	0.7036	0.7025	0.6995	0.6986
	BA5	10/10	0.7746	0.7746	0.7036	0.7025	0.6995	0.6986
Best and medium attributes (BM)	BM1	3/3	0.7011	0.7011	0.7031	0.7011	0.7056	0.7056
	BM2	5/5	0.7682	0.7682	0.6843	0.6935	0.6992	0.6992
	BM3	5/5	0.7682	0.7682	0.6843	0.6935	0.6992	0.6992
	BM4	10/14	0.7832	0.7792	0.7036	0.7025	0.6995	0.6986
	BM5	10/14	0.7832	0.7792	0.7036	0.7025	0.6995	0.6986
Average			0.7591[6]	0.7583[5]	0.6958[1]	0.6986[2]	0.7006[4]	0.7002[3]

Table 6 The average F-measure, Entropy and Purity values on the CA and CMC datasets.

Datasets	Evaluation metric	ModEx	Ex-Detective	PAM	SK	BFPH	NFPH
CA	Average F-measure (higher the better)	0.7591[6]	0.7583[5]	0.6958[1]	0.6986[2]	0.7006[4]	0.7002[3]
	Average Entropy (lower the better)	0.7861[6]	0.7867[5]	0.8899[4]	0.9114[3]	0.9367[2]	0.9374[1]
	Average Purity (the higher the better)	0.7299[6]	0.7289[5]	0.6609[4]	0.6396[3]	0.6113[2]	0.6110[1]
CMC	Average F-measure (higher the better)	0.5280[6]	0.5218[5]	0.5095[4]	0.5030[3]	0.4960[2]	0.4945[1]
	Average Entropy (lower the better)	1.4230[6]	1.4234[5]	1.4658[3]	1.4667[1]	1.4654[4]	1.4658[3]
	Average Purity (higher the better)	0.4781[6]	0.4773[5]	0.4624[4]	0.4572[3]	0.4524[2]	0.4508[1]

Table 7 The comparison of Seed-Detective and ModEx on all datasets.

Datasets	Average *F*-measure (higher the better)		Average Entropy (lower the better)		Average Purity (higher the better)	
	Seed-Detective	ModEx	Seed-Detective	ModEx	Seed-Detective	ModEx
SH	0.8046[2]	0.6484[1]	0.6538[2]	0.7300[1]	0.8007[2]	0.6437[1]
CA	0.7872[2]	0.7591[1]	0.7428[2]	0.7861[1]	0.7538[2]	0.7299[1]
GCA	0.8070[2]	0.8010[1]	0.8271[2]	0.8436[1]	0.7181[2]	0.7129[1]
CMC	0.5366[2]	0.5280[1]	1.4754[1]	1.4230[2]	0.4549[1]	0.4781[2]

Table 8 Clustering evaluation of Seed-Detective, PAM, SK, BFPH and NFPH on all datasets.

Datasets	Evaluation metric	Seed-Detective	PAM	SK	BFPH	NFPH
SH	Average *F*-measure	0.8046[5]	0.6805[4]	0.6509[3]	0.6365[2]	0.6349[1]
	Average Entropy	0.6538[5]	0.9248[3]	0.9194[4]	0.9342[2]	0.9527[1]
	Average Purity	0.8007[5]	0.6408[4]	0.6368[3]	0.6208[2]	0.6100[1]
CA	Average *F*-measure	0.7872[5]	0.6974[1]	0.7010[2]	0.7033[3]	0.7034[4]
	Average Entropy	0.7428[5]	0.9503[4]	0.9669[3]	0.9760[1]	0.9759[2]
	Average Purity	0.7538[5]	0.6006[4]	0.5761[3]	0.5651[1]	0.5655[2]
GCA	Average *F*-measure	0.8070[5]	0.8038[3]	0.8061[4]	0.8005[2]	0.7994[1]
	Average Entropy	0.8271[5]	0.8629[1]	0.8609[4]	0.8619[3]	0.8621[2]
	Average Purity	0.7181[5]	0.7086[3]	0.7097[4]	0.7084[2]	0.7080[1]
CMC	Average *F*-measure	0.5366[4]	0.5370[5]	0.5164[1]	0.5206[3]	0.5192[2]
	Average Entropy	1.4754[5]	1.4990[1]	1.4835[2]	1.4781[4]	1.4792[3]
	Average Purity	0.4549[5]	0.4359[1]	0.4423[4]	0.4395[3]	0.4364[2]

Table 9 The average SSE (lower the better) of Seed-Detective, PAM, SK, BFPH and NFPH on all datasets.

Datasets	Seed-Detective	PAM	SK	BFPH	NFPH
SH	16.0819[5]	16.9576[3]	16.5886[4]	17.1009[2]	17.3047[1]
CA	23.3226[5]	25.9202[4]	25.9430[3]	26.5553[1]	26.5530[2]
GCA	56.4133[5]	68.2010[3]	65.8282[4]	68.7982[2]	68.8912[1]
CMC	37.1672[2]	35.2871[3]	34.0954[5]	38.5754[1]	34.7343[4]

the techniques. Clearly the cluster quality of ModEx is better than the cluster qualities of the other techniques.

In the experiments of Seed-Detective, we use the same weight patterns that we used for the experiments on ModEx (see Table 4). For each individual weight pattern of the BA and BM category (see Table 4), we compare the quality of the clustering results obtained by Seed-Detective and ModEx in four datasets. In Table 7, we present the average values of *F*-measure, Entropy and Purity for each dataset. Clearly Seed-Detective performs better than ModEx in all cases except the two evaluation criteria in CMC.

We also compare the performance of Seed-Detective with PAM, SK, BFPH and NFPH in the same way as we do for ModEx. In Table 8 we present the average *F*-measure, Entropy and Purity values for four datasets. Seed-Detective performs the best in all three evaluation criteria for all four datasets.

Moreover we also compare the performance of the techniques by using one internal cluster evaluation criteria called Sum of Square Error (SSE) (Tan et al., 2005). Unlike external evaluation criteria such as *F*-measure, Purity and Entropy, the internal cluster evaluation criterion does not require any external information like the class values (labels) of the records. In Table 9 we present the average SSE (lower the better) values of the techniques. From Table 9, we see that Seed-Detective performs better than the existing techniques in all datasets except CMC.

We also calculate the average execution time required by each technique, as presented in Table 10. The experiment is carried out in a machine that has Intel (R) Core (TM) i5 CPU M430 @ 2.27GHZ and 4 GB of RAM. Both ModEx and Seed-Detective require higher execution time than K-Means and its variants (SK, BFPH, NFPH and PAM). This is to pay the price of finding high quality seeds that ultimately result in better quality clustering results. Additionally, the execution time required by ModEx, Seed-Detective and Ex-Detective (an existing technique) are very similar to each other.

Note that the proposed techniques are not suitable for a time critical application where a clustering solution is needed on the fly and the dataset is dynamic meaning that new data are being added regularly. On the other hand there are many non-time critical applications such as a research on a disease pattern extraction (from a patient dataset) for disease prediction, prevention and treatment where accuracy is more

Table 10 The execution times (in seconds) of the techniques on the datasets.

Datasets	Seed-Detective	ModEx	Ex-Detective	PAM	SK	BFPH	NFPH
CA	12.098	12.0138	12.0139	3.1034	0.4167	0.1047	0.1342
CMC	168.4991	168.1751	168.1949	52.0887	0.5182	0.5944	0.6602
SH	1.693	NA	NA	0.7410	0.20875	0.14637	0.1905
GCA	64.245	NA	NA	18.2110	3.5084	1.8189	1.937

important than speed. The proposed techniques are more suitable for such non-time critical applications.

Moreover, time complexity of our techniques can be reduced in many ways. For example, a suitable feature reduction technique can be used to heavily drop the number of attributes for the datasets that have a huge number of attributes. Since our techniques make use of decision trees for clustering the reduction of number of attributes can reduce the time complexity heavily. Note that decision tree algorithms such as C4.5 has a time complexity of $O(nm^2)$ (Su and Zhang, 2006), where n is the number of records and m is the number of attributes in a dataset.

4.1. Statistical analysis

For a parametric statistical significance test (such as t-test and confidence interval test) it is important that the results follow two conditions: 1. a normal distribution and 2. equal variance (Triola, 2001). We observe that the values (of the clustering results obtained in the experiments of this study) for the techniques do not follow a normal distribution and the variances of the values are different. Since the results do not satisfy the conditions of a parametric test, we carry out the non-parametric sign test on the clustering results (*F*-measure, Entropy and Purity) of the techniques in order to evaluate the statistical significance of the superiority of ModEx and Seed-Detective over the existing techniques.

The sign test (Mason et al., 1998; Triola, 2001) is carried out at the right tail considering significance level $\alpha = 0.10$ (i.e. 90% significance level). In Fig. 14, we present the z-values (test statistics values) while comparing ModEx with the existing techniques based on the CA and CMC datasets. The first five bars in Fig. 14 represent the z-values while comparing ModEx with the existing techniques and the 6th bar represents the z-ref value. If any z-value is greater than z-ref value then the results of ModEx is significantly better than the results

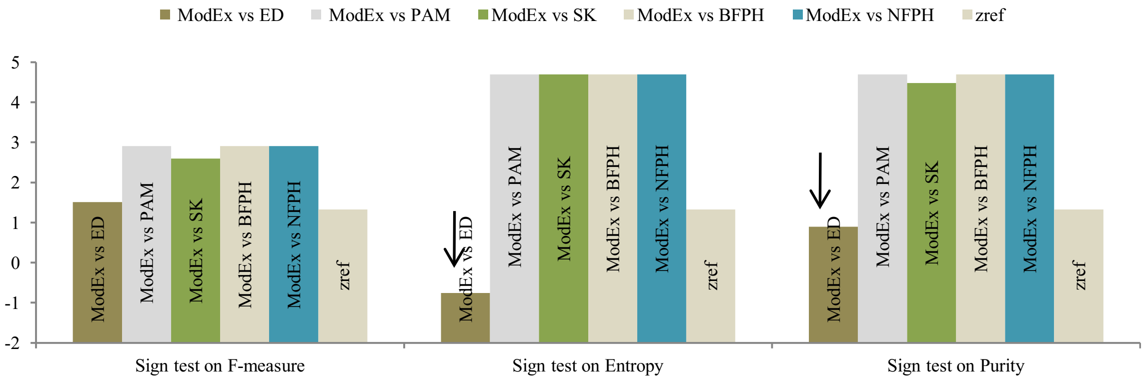


Figure 14 The sign test results of ModEx.

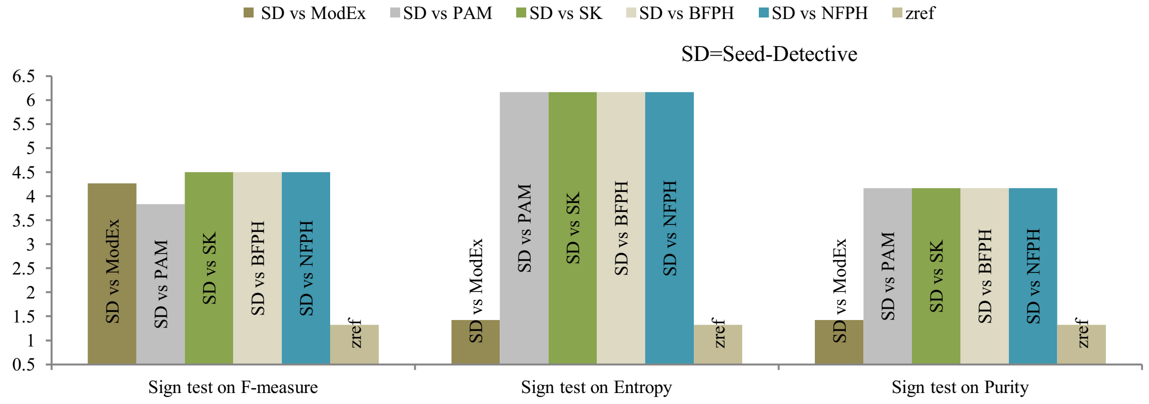


Figure 15 The sign test results of Seed-Detective.

of the existing technique. For $\alpha = 0.10$ and degree of freedom df = 19 (since we have 20 results for the two datasets), the *z*-ref value is 1.328 (Mason et al., 1998; Triola, 2001). From Fig. 14, we see that ModEx performs significantly better than the other existing techniques namely SK, PAM, BFPH and NFPH in terms of *F*-measure, Entropy and Purity. However, the performance of ModEx is not significantly better than Ex-Detective (ED) in terms of Entropy and Purity. The cases where the ModEx is not significantly superior have been marked with arrow signs on top in Fig. 14.

Similarly, we present the sign test results comparing Seed-Detective with the existing techniques in Fig. 15. For Seed-Detective, we carry out sign tests on the CA, CMC, SH and GCA datasets. From Fig. 15, we see that Seed-Detective performs significantly better than the existing techniques in terms of *F*-measure, Entropy and Purity. Therefore, there is no arrow sign for this result.

5. Conclusion

In this paper we present two clustering techniques called ModEx and Seed-Detective. ModEx is a modified version of an existing clustering technique called Ex-Detective(Islam, 2008; Islam and Brankovic, 2011). Seed-Detective is a combination of a modified version of ModEx and Simple K-Means. In Seed-Detective we use a modified version of ModEx to produce high quality initial seeds. The high quality initial seeds are then given as input to K-Means to produce final clusters. The expectation is that with high quality initial seeds K-Means will produce high quality clusters.

The performance of ModEx and Seed-Detective is compared with the performance of Ex-Detective, PAM, Simple K-Means (SK), Basic Farthest Point Heuristic (BFPH) and New Farthest Point Heuristic (NFPH). We use three cluster evaluation criteria namely *F*-measure, Purity and Entropy by using four natural datasets that we obtain from the UCI machine learning repository (Bache and Lichman, 2013). The experimental results strongly indicate a superiority of ModEx and Seed-Detective over the existing techniques. Moreover, Seed-Detective performs better than ModEx in general. Our sign test results further prove the superiority of Seed-Detective (and ModEx) over the existing techniques.

References

Ahmad, A., Dey, L., 2007. A k-mean clustering algorithm for mixed numeric and categorical data. Data Knowl. Eng. 63 (2), 503–527. http://dx.doi.org/10.1016/j.datak.2007.03.016.

Bache, K., Lichman, M., 2013. UCI Machine Learning Repository. Available from: <http://archive.ics.uci.edu/ml>.

Bagirov, A.M., 2008. Modified global-means algorithm for minimum sum-of-squares clustering problems. Pattern Recogn. 41 (10), 3192–3199. http://dx.doi.org/10.1016/j.patcog.2008.04.004.

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

Berzal, F., Cubero, J.-C., Marı́n, N., Sánchez, D., 2003. Numerical attributes in decision trees: a hierarchical approach. In: Berthold, M.R., Lenz, H.-J., Bradley, E., Kruse, R., Borgelt, C. (Eds.), . In: Advances in Intelligent Data Analysis V, vol. 2810. Springer, Berlin Heidelberg, pp. 198–207.

Berzal, F., Cubero, J.-C., Marı́n, N., Sánchez, D., 2004. Building multi-way decision trees with numerical attributes. Inf. Sci. 165 (1–2), 73–90. http://dx.doi.org/10.1016/j.ins.2003.09.018.

Chou, C.-H., Su, M.-C., Lai, E., 2004. A new cluster validity measure and its application to image compression. Pattern Anal. Appl. 7 (2), 205–220. http://dx.doi.org/10.1007/s10044-004-0218-1.

Chuan Tan, S., Ming Ting, K., Wei Teng, S., 2011. A general stochastic clustering method for automatic cluster discovery. Pattern Recogn. 44 (10–11), 2786–2799. http://dx.doi.org/10.1016/j.patcog.2011.04.001.

Chuang, K.-T., Chen, M.-S., 2004. Clustering categorical data by utilizing the correlated-force ensemble. In: Paper presented at the 4th SIAM International Conference on Data Mining (SDM 04), Lake Buena Vista, Florida.

Han, J., Kamber, M., 2006. Data Mining Concepts and Techniques, second ed. Morgan Kaufmann, San Francisco.

He, Z., 2006. Farthest-Point Heuristic Based Initialization Methods for K-Modes Clustering. CoRR, abs/cs/0610043.

Huang, Z., 1997. Clustering large data sets with mixed numeric and categorical values. In: Paper presented at the First Pacific-Asia Conference on Knowledge Discovery and Data Mining, Singapore.

Islam, M.Z., 2008. Privacy Preservation in Data Mining Through Noise Addition (Doctor of Philosophy), University of Newcastle, Callaghan, NSW, Australia.

Islam, M.Z., 2012. EXPLORE: a novel decision tree classification algorithm. Data Secur. Secur. Data 6121, 55–71. http://dx.doi.org/10.1007/978-3-642-25704-9_7.

Islam, M.Z., Brankovic, L., 2005. DETECTIVE: a decision tree based categorical value clustering and perturbation technique in privacy preserving data mining. In: Paper presented at the 3rd International IEEE Conference on Industrial Informatics (INDIN 2005), Perth, Australia.

Islam, M.Z., Brankovic, L., 2011. Privacy preserving data mining: a noise addition framework using a novel clustering technique. Knowl.-Based Syst. 24 (8), 1214–1223. http://dx.doi.org/10.1016/j.knosys.2011.05.011.

Jain, A.K., 2010. Data clustering: 50 years beyond K-means. Pattern Recogn. Lett. 31 (8), 651–666. http://dx.doi.org/10.1016/j.patrec.2009.09.011.

Khan, F., 2012. An initial seed selection algorithm for k-means clustering of georeferenced data to improve replicability of cluster assignments for mapping application. Appl. Soft Comput. 12 (11), 3698–3700, http://dx.doi.org/10.1016/j.asoc.2012.07.021.

Kim, S., Lee, J., Bae, J., 2006. Effect of data normalization on fuzzy clustering of DNA microarray data. BMC Bioinf. 7 (1), 1–14. http://dx.doi.org/10.1186/1471-2105-7-134.

Kurgan, L.A., Cios, K.J., 2004. CAIM discretization algorithm. IEEE Trans. Knowl. Data Eng. 16 (2), 145–153. http://dx.doi.org/10.1109/TKDE.2004.1269594.

Maitra, R., Peterson, A., Ghosh, A., 2010. A systematic evaluation of different methods for initializing the K-means clustering algorithm. IEEE Trans. Knowl. Data Eng.

Malik, F., Baharudin, B., 2013. Analysis of distance metrics in content-based image retrieval using statistical quantized histogram texture features in the DCT domain. J. King Saud University – Comput. Inf. Sci. 25 (2), 207–218, http://dx.doi.org/10.1016/j.jksuci.2012.11.004.

Maqbool, O., Babri, H.A., 2006. Automated software clustering: an insight using cluster labels. J. Syst. Softw. 79 (11), 1632–1648, http://dx.doi.org/10.1016/j.jss.2006.03.013.

Mason, R., Lind, D., Marchal, W., 1998. Statistics: An Introduction, fifth ed. Brooks/Cole Publishing Company.

Noordam, J.C., van den Broek, W.H.A.M., Buydens, L.M.C., 2002. Multivariate image segmentation with cluster size insensitive Fuzzy C-means. Chemom. Intell. Lab. Syst. 64 (1), 65–78, http://dx.doi.org/10.1016/S0169-7439(02)00052-7.

Quinlan, J.R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Quinlan, J.R., 1996. Improved use of continuous attributes in C4.5. J. Artif. Intell. Res. 4 (1), 77–90.

Su, J., Zhang, H., 2006. A fast decision tree learning algorithm. In: Paper presented at the Proceedings of the 21st National Conference on Artificial intelligence – vol. 1. Boston, Massachusetts.

Tan, P.-N., Steinbach, M., Kumar, V., 2005. Introduction to Data Mining, first ed. Pearson Addison Wesley.

Triola, M.F., 2001. Elementary Statistics, eighth ed. Addison Wesley Longman Inc..

Visalakshi, N.K., Thangavel, K., 2009. Impact of normalization in distributed K-Means clustering. Int. J. Soft Comput. 4 (4), 168–172.

Yang, Y., Webb, G., 2009. Discretization for naive-Bayes learning: managing discretization bias and variance. Mach. Learn. 74 (1), 39–74. http://dx.doi.org/10.1007/s10994-008-5083-5.