4. INCREMENTAL CLUSTERING APPROACH TO MIXED DATA

Data clustering is a crucial data mining method used in diverse domains for discovering patterns in the underlying data. Traditional clustering algorithms can be classified into numeric, categorical, and mixed categories according to the type of datasets they handle. Algorithms focusing on purely numeric data such as BIRCH, DBSCAN, CURE, etc. make use of the inherent geometric properties of the data to define distance measures like Euclidean distance between data points to cluster data. But many databases also contain categorical attributes like gender, religion, nationality etc. Euclidean distance cannot capture the similarity of data elements in case of categorical attributes. Hence numeric clustering algorithms become inappropriate for clustering categorical data. So another set of algorithms like COBWEB, ROCK, CACTUS etc. were designed for clustering categorical data. However algorithms focusing on only numeric data or only categorical data are not suitable to cluster datasets with mixed attributes directly.

Datasets with mixed types of attributes including categorical and numeric are very common in data mining applications like banking sector, health data and web-log data. Such domains maintain dynamically growing datasets described in terms of heterogeneous attributes and hence require incremental clustering algorithms that an handle all types of attributes. To the best of the knowledge of the author, existing incremental clustering algorithms including CFICA, developed by the author are limited to handle either numeric or categorical attributes but not their combinations. So the author proposes an incremental clustering algorithm called M-CFICA (Cluster Feature-Based Incremental Clustering

Approach to Mixed Data) that extends the basic ideas of CFICA approach. While designing M-CFICA, the Inverse Proximity Estimate (IPE) devised in chapter 3 was extended to deal with mixed distances that estimate the distance between two data points represented in terms of both numeric and categorical attributes.

4.1 Distance Estimation for Mixed type of data

An attribute of any type can be transformed into one of the basic data types namely numeric or categorical. For example, a binary attribute can be considered as a categorical attribute with two distinct values; an ordinal attribute with a small number of distinct values can be transformed into categorical attribute but if it has a large number of distinct values like ranking sequence it can be transformed into a numerical attribute [Han and Kamber 2001]. Hence data sets with mixed types of attributes can be represented in terms of numeric or categorical attributes either directly or after transformation. The distance between such data points is estimated separately in terms of numeric attributes and categorical attributes to arrive at Normalized Euclidean Distance (NED) and Dissimilarity (DS) respectively.

In this thesis, the author has devised a new approach to estimate the Dissimilarity (DS) between two data points based on their categorical attributes. The distinct values of categorical attributes are expressed as attribute-value pairs [Chen H.L et. al., 2005] and the prominence of various attribute-value pairs in a cluster of data points characterizes the cluster.

4.1.1. Specificity of an Attribute-value pair

An attribute-value pair is defined as $<$ attribute name, attribute value $>$. It is denoted by AV_r. An attribute-value pair, (AV_r) will help in avoiding the ambiguity that might arise due to identical attribute values for multiple attributes. For example, the same

city *Paris* can be the value of source as well as destination attributes which is represented as 2 attribute-value pairs < Source, Paris and Destination, Paris. AV_{ir} refers to the attribute value pair AV_r that occurs in the cluster C_i and $|AV_{ir}|$ refers to the frequency of AV_r in C_i .

Hung-Len Chen et.al [Chen H.L et al. 2009] suggested a weighting function to measure the distribution of attribute-value pairs among clusters as in Eq. 4.1

$$f(AV_r) = 1 - \frac{-1}{\log k} * \sum_{i=1}^{k} p(AV_{ir}) \log \left(p(AV_{ir})\right) \tag{4.1}$$

where

$$p(AV_{ir}) = \frac{|AV_{ir}|}{\Sigma_{z=1}^{k} |AV_{ir}|} \tag{4.2}$$

f (AV$_r$) is adopted to estimate the specificity of the attribute-value pair AV_r in discriminating the clusters as it resembles information theoretic entropy measure, E(X) which is expressed in Eq 4.3 as follows:

$$E(X) = -\sum_{i=1}^{n} p(x_i) \log\left(p(x_i)\right) \tag{4.3}$$

Entropy measures information requirement or uncertainty on a random variable (say X) with possible states x_1, x_2……. x_n and p (x_i) is the probability of the i^{th} state for X. The bindings of X to x_1, x_2……. x_n is analogous to occurrence of an attribute-value pair in clusters 1 to n. The entropy is maximal when X possesses minimum information indicated by equal possibility to assume any state. An attribute-value pair that occurs in all clusters uniformly has minimum discrimination power while predicting the cluster and hence has

maximum entropy. The maximum entropy value for k partitions is equal to log k. So the entropy value is divided by log k to normalize the weighting function into a scale of [0, 1].

Based on these concepts, the author has devised a specificity estimate of an attribute-value pair in terms of $f(AV_r)$ as in Eq. 4.4:

$$f(AV_r) = 1 - \frac{Entropy}{Log\ k} \tag{4.4}$$

It may be noted that $f(AV_r)$ reaches its maximum of one for those attribute-value pairs which are confined to one cluster and it gradually decreases as the attribute-value pair extends to other clusters. It reaches zero when it is uniformly distributed among all clusters.

4.1.2 Prominence of attribute-value pairs

Certain attribute-value pairs are shared by most of the members of a cluster while the other attribute-value pairs are not so common among the members of the cluster. The prominence of an attribute-value pair in a cluster increases proportionally with its frequency within the cluster. In addition, the prominence also increases proportionally with specificity of the attribute-value pair.

The prominence of each of an attribute-value pair AV_r in a cluster  is estimated as in Eq. 4.5.

$$w(c_i, AV_r) = \frac{|AV_{ir}|}{n_i} \times f(AV_r) \tag{4.5}$$

where

$w(c_i, AV_{ir})$ → Prominence of an attribute value pair [redacted] in cluster [redacted]

$|AV_{ir}|$ → Count or frequency of [redacted] in cluster [redacted]

n_i → number of data points in cluster [redacted]

$f(AV_r)$ → Specificity of the attribute-value pair

Thus the prominence of an attribute-value pair is estimated and the most prominent n_c (number of categorical attributes) number of attribute-value pairs were selected to represent the categorical component of the cluster referred to as 'Weight vector' (w_i) with n_c components of the form : $< AV_{ir}, w(c_i, AV_{ir}) >$. These n_c important attribute-value pairs that constitute w_i are referred to as characterizing attribute-value pairs of cluster i.

The Weight vector w_i represents the cluster prototype from the perspective of categorical attributes while the mean vector represents the prototype in the perspective of numerical attributes.

4.2 Mixed Distance

The proximity between two data points involving both numeric and categorical attributes is estimated by combining the distance estimated in terms of numeric attributes & dissimilarity estimated in terms of categorical attributes. In this process, the author proposes to transform the distance as well as dissimilarity into the same range, per se [0, 1] and combine them giving due importance based on the cardinalities of numeric versus categorical attributes.

1) **Numerical distance:**

The commonly used distance measure for computing the distance between the two data points with respect to its numerical attributes is Euclidean distance. The author has devised a

Normalized Euclidean Distance (NED) that is confined to a range [0, 1] by transforming the original Euclidean distance. Since all the numerical attributes were preprocessed and normalized into a range of [0, 1] individually, the maximum value of Euclidean distance for a given pair of data points is $\sqrt{n_n}$ and hence the Euclidean distance value is divided by $\sqrt{n_n}$ while defining Normalized Euclidean Distance as given below in Eq. 4.6

$$\text{NED}(\Delta y, C_i) = \frac{ED(\Delta y, \quad c_i)}{\sqrt{n_n}} \tag{4.6}$$

Where $ED(\Delta y, \quad c_i)$ is the Euclidean distance between centroid C_i and the incoming data point Δy and n_n is the number of numerical attributes.

2) Categorical dissimilarity:

Normally, the dissimilarity measure between two data points described in terms of '*m'* categorical attributes is estimated as the ratio of the number of mismatches to m. This conventional approach to estimating dissimilarity ignores the specificity of an attribute-value pair giving equal importance to all attribute-value pairs. Hence a data point with an essential characteristic represented by an attribute-value pair and another data point having a general characteristic found among the members of all clusters are considered equally close to a cluster. To circumvent this drawback the author suggests that the categorical nature of a cluster is better represented in terms of specificity and prominence of attribute-value pairs. Hence, the author has made use of *Resemblance metric* [Chen H.L et al. 2009] while defining the dissimilarity of a data point to a cluster/data point.

The dissimilarity (DS) of a data point to either a cluster prototype or another data point is confined to a range of [0, 1] based on the following definition

$$\text{DS}(\Delta y, Ci) = 1 - \frac{R(\Delta y, Ci)}{n_c} \tag{4.7}$$

where,

n_c → number of categorical attributes

DS(Δy, C_i) → Dissimilarity between centroid C_i and the incoming data point Δy.

R (Δy, C_i) → signifies resemblance of a point Δy to the i^{th} cluster

$$R\ (\Delta y, c_i) = \sum_{I_r \in \Delta y} w(\ c_i, AV_r\ ) \tag{4.8}$$

3) Weightage:

'α' represents the relative weightage given to numerical component compared to categorical component for estimating the mixed distance. The weightage is calculated based on the number of numerical as well as categorical attributes in the dataset as in the following Eq. 4.9

$$\alpha = \frac{n_n}{n_n + n_c} \tag{4.9}$$

where, n_n → Total number of numerical attributes in the dataset

n_c → Total number of categorical attributes in the dataset

M-CFICA defines mixed distance between a pair of data points described in terms of heterogeneous attributes. It is estimated as the weighted sum of Normalized Euclidean Distance and dissimilarity and is used while forming initial clusters for the static database.

$$MD\ (C_i, \Delta y) = [\alpha \times NED\ (C_i, \Delta y)] + [(1 - \alpha)\ DS\ (C_i, \Delta y)] \tag{4.10}$$

where

$MD\ (C_i, \Delta y)$ = mixed distance between each cluster centroid c_i and data points Δy .

$NED\ (C_i, \Delta y)$ = Normalized Euclidean distance between the data point Δy and cluster centroid c_i.

$DS\ (C_i, \Delta y)$ = Categorical dissimilarity is computed for the data point Δy and cluster centroid c_i using the dissimilarity measure.

α = weightage calculated based on the number of numerical as well as categorical attributes in the dataset

4.3 Initial cluster formation of the static database

Clustering both the static database as well as the incremental database is important in the case of the incremental clustering. Initial cluster formation is performed on the static database first. Like CFICA, the proposed algorithm M-CFICA also employs a partitional clustering algorithm which makes use of the mixed distance estimate defined in the previous section. In CFICA, the author has used the k-means clustering algorithm so that, k number of clusters are obtained. Since the *k*-means algorithm cannot cluster categorical objects the *k*-means algorithm has been modified to cater to the categorical component also in M-CFICA.

4.3.1 Modified k-means algorithm

Step 1: ***Initialization:***

'k' data points were randomly selected as cluster centroids for formation of initial clusters.

Step 2: ***Distance calculation:***

For each data point Δy, the mixed distance from the data point to each cluster centroid c_i is calculated as in Eq.4.10

Step 3: ***Assignment:***

A data point is assigned to its nearest cluster based on the mixed distance, in the similar way as described in chapter 3.

Step 4: ***Centroid recalculation:***

For each cluster, re-compute the cluster centroid and update the k centroids based on the new members assigned to k clusters.

- The average of the numerical attribute values of the members within a cluster characterizes the numerical component of the cluster centroid.
- For categorical attribute, the weight vector (w_i) formed by the most prominent n_c attribute-value pairs characterizes categorical component of the cluster centroid.

Step 5: ***Convergence condition:***

Step 2 to Step 4, were repeated until no data point moves from one cluster to another and all the clusters become stable.

4.4 Clustering of incremental database

A set of clusters, $C = \{C_1, C_2, \cdots, C_k\}; 1 \le i \le k$ are initially obtained using the modified k-means algorithm on the static database  . The concept of Cluster Feature (CF) discussed in chapter 3 section 3.3 is further extended as per the requirements of M-CFICA for clustering the incremental database. The Cluster Feature is so designed to represent numeric as well as categorical nature of the members of a cluster in a nutshell, while retaining all aspects that are essential for its incremental update. The clustering solution of  is represented in the form of Cluster Features.

4.5 Computation of Cluster Feature (CF)

The original structure of cluster feature used in CFICA has been slightly modified to accommodate categorical data also as CFICA caters to numerical data only.

In M-CFICA the Cluster Feature is denoted as,

$$CF_i = \{n_i, \overrightarrow{m_i}, \overrightarrow{w_i}, Q_i, \overrightarrow{ss_i}, \overrightarrow{m_i'}, \overrightarrow{AV_{ir}'}\} \qquad (4.11)$$

where

CF_i → Cluster feature for cluster i (C_i)

n_i → number of data points in cluster C_i

$\overrightarrow{m_i}$ → mean vector of numerical attributes in the cluster C_i with respect to which farthest points are calculated

$\overrightarrow{w_i}$ → weights of characterizing attribute-value pairs of cluster C_i

Q_i → List of p-farthest points of cluster C_i

$\overrightarrow{SS_i}$ → Squared sum vector that changes during incremental updates

$\overrightarrow{m_i'}$ → new mean vector of the cluster C_i including newly added points to the cluster C_i.

$\overrightarrow{AV_{ir}'}$ → updated count of all attribute value pairs occurred in i^{th} cluster

4.6 Proximity Estimation for a new data point

Developing an effective clustering algorithm to cater to mixed data in general, needs a very effective distance measure to identify the most relevant cluster. So, the author makes use of 'Mixed Distance' estimate presented in section.. as well as Inverse Proximity Estimate (IPE) proposed in chapter 3. The IPE considers the proximity of a data point to a cluster centroid as well as its proximity to a farthest point of the cluster in its vicinity to determine the membership of a data point in a cluster which is formally stated below in Eq. 4.12

$$IPE\ \square_{\Delta y}^{(i)} = MD\ (C_i, \Delta y) + [MD\ (q_i, \Delta y) * MD\ (C_i, q_i)] \quad (4.12)$$

where

$IPE\ \square_{\Delta y}^{(i)}$ → Inverse Proximity Estimate of incoming data point Δy to the i^{th} cluster C_i

$MD\ (C_i, \Delta y)$ → Mixed distance from the data point Δy to the centroid of cluster C_i which is given in Eq. 4.10

$$MD\ (C_i, \Delta y) = [\alpha \times NED\ (C_i, \Delta y)] + [(1 - \alpha)\ DS\ (C_i, \Delta y)] \quad (4.10)$$

$MD\ (q_i, \Delta y)$ → Mixed distance from farthest point q_i to the data point Δy

$$MD\ (q_i, \Delta y) = [\alpha * NED\ (q_i, \Delta y)] + [(1 - \alpha) * DS\ (q_i, \Delta y)] \quad (4.11)$$

$MD\ (C_i, q_i)$ → Mixed distance from cluster C_i to the farthest point q_i

$$MD\ (C_i, q_i) = [\alpha * NED\ (C_i, q_i)] + [(1 - \alpha) * DS\ (C_i, q_i)] \quad (4.12)$$

Though, the mixed distance measures employed to estimate the $IPE\ \square_{\Delta y}^{(i)}$ are in the range of [0, 1] the inverse proximity estimate may exceed 1.0.

4.7 Insertion of a new data point

After initial clustering of the static database, the clustering solution is converted into the form of Cluster Features (CFs). When there is an incoming data point Δy to be inserted into one of the existing 'k' clusters decisions regarding the inclusion of new data points involves estimation of $IPE\ \square_{\Delta y}^{(i)}$ for each cluster i. Though the Cluster Feature maintains $\overrightarrow{m_i'}$ and $\overrightarrow{AV_{ir}'}$ for keeping track of possible *concept – drift* associated with the cluster, estimation of inverse proximity estimate only considers (m_i , w_i) referred to as its prototype. If the incoming data point Δy cannot be included into any of the existing 'k' clusters then a new singleton cluster will be formed with this data point Δy In such a case the number of clusters is increased by one and the Cluster Feature is constructed for the newly added cluster.

4.8 Finding p-farthest points of a cluster

The procedure for finding the list of p-farthest points (Q_i) of the cluster c_i is the same as discussed in chapter 3 section 3.3. The only difference is that now mixed distance $MD\ (\Delta x, C_i)$ is used for calculating the distance between every data point Δx present in

the cluster c_i to the cluster prototype of the relevant cluster c_i instead of simple Euclidean distance using the formula:

$$MD\ (\Delta x, C_i) = [\alpha * NED\ (\Delta x, C_i)] + [(1 - \alpha) * DS\ (\Delta x, C_i,)] \quad (4.12)$$

where

$NED\ (\Delta x, C_i)$ = Euclidean distance between points Δx and C_i from Eq.4.6

$$\sqrt{n_n}$$

$$DS\ (\Delta x, c_i) = 1 - \frac{1}{n_c} \sum_{AV_r \in \Delta x} w(\ c_i, AV_r\ ) \quad (4.13)$$

n_n = no. of numerical attributes

n_c = number of categorical attributes

$w(c_i, AV_{ir})$ → Weight of attribute value pair AV_r in cluster c_i

Based on the mixed distance between every data point Δx of cluster c_i to its prototype $(\ \overrightarrow{m_i}, \overrightarrow{w_i}\ )$ p- farthest points are identified and maintained in Q_i.

4.9 Finding farthest point in the vicinity of Δy

Mixed distance between the newly arrived data point Δy and each of the p-farthest points (Q_i) is computed and one point that has the minimum distance is chosen as the farthest point, q_i in the vicinity of Δy for that cluster.

$$q_i = argmin_{q_j \in Q_i} \{MD\ (\Delta y, q_j)\} \quad (4.14)$$

where

$MD\ (\Delta y, q_j) = [\alpha * NED(\Delta y, q_j)] + [(1-\alpha) DS\ (\Delta y, q_j)]$ as in Eq.4.11

$NED(\Delta y, q_j)$ = Euclidean distance between points q_j, Δy from Eq.4.6

$$\sqrt{n_n}$$

$$DS\ (\Delta y, q_j) = 1 - \frac{1}{n_c} \sum_{r \in (\Delta y \cap q_j)} f\ (AV_r) \qquad (4.15)$$

n_n = no. of numerical attributes

n_c = number of categorical attributes

$f(AV_r)$ → Specificity of the attribute-value pair

It may be noted that estimating dissimilarity between a pair of points (Eq. 4.15) is slightly different from estimating dissimilarity of a point to a cluster (Eq. 4.13) represented by its weight vector.

4.10 Updating of Cluster Feature

Inclusion of a data point Δy into the already existing cluster solution requires updating of the Cluster Feature as Δy may be included either into any of the existing clusters or it may form a new cluster.

Case 1: New data point forms a separate cluster

When the new data point form a separate cluster, then the Cluster Feature will be updated as follows:

(1) $n_i = 1$

$\longrightarrow \quad \longrightarrow$

(2) $m_i = m_i'$

(3) w_i = Calculated based on AV_{ir}'

(4) $Q_i = \Delta y$

(5) SS_i = squared sum components of Δy

(6) $AV_{ir}' \rightarrow$ count of all attribute value pairs in Δy

Case 2: New data point included in existing cluster i

Whenever a data point is included in an existing cluster i, the CF_i has to be updated as given below:

(1) $n_i = n_i + 1$

(2) $m_i' = m_i' +$ *numerical component of* Δy

(3) (m_i', w_i) *remains the same*

(4) Q_i represent the p-farthest points

(5) $SS_i = SS_i +$ squares of numerical components of Δy

(6) $AV_{ir}' \rightarrow AV_{ir}' +$ characterizing attribute value pairs in Δy

Updating of the Cluster Feature upon inclusion of a data point into an existing cluster or formation of a separate cluster is iteratively performed for the whole chunk of data points in the incremental database ΔS_D.

4.11 Merging of closest cluster pair

After processing the incremental database ΔS_D with M-CFICA, a merging strategy as discussed in chapter 3 section 3.9 is used to maintain reasonable number of clusters. Since CFICA deals with numerical attributes only ensuring minimum increase in variance while merging was performed. But now the categorical component of the clusters has also to be dealt with. For this purpose, the dissimilarity measure is used.

Merging involves two clusters with' n' data points and combines them so that the resultant cluster formed is of better quality. Although merging could be attempted to limit the number of clusters to k, on all possible cluster pairs whenever new clusters are created it would lead to redundancy and become costly. Instead, only the closest cluster pairs are considered for merging. A cluster pair is considered closest if only the Mixed distance between the centroids of the pair of clusters is smaller than user defined merging threshold (θ).

The procedure used for the merging process is described below:

Step 1: Estimate the Euclidean distance between every pair of clusters based on $\overrightarrow{m_i}$.

Step 2 : For those cluster pairs whose Euclidean distance, ED less than the merging threshold value, θ (ED $\ll \theta$) ,

a) Find increase in variance (σ^2) as described in section 3.2.1 for numerical attributes in every cluster pair.

b) Sort the cluster pair based on the ascending order of increase in variance and accordingly prepare a ranking list of cluster pairs.

c) Similarly, find increase in dissimilarity (DS) for the categorical attributes present in every cluster pair as follows:

For each cluster, C_i, the dissimilarity is calculated as in Eq. 4.16

$$DS_i = 1 - \frac{Avg\ \{R\ (\ AV_i',\ C_i)\ \}}{n_c} \tag{4.16}$$

where, $Avg\ \{R\ (\ AV_i, C_i)\} = \frac{\sum_{r\ \in AV_i} w\ (\ AV_{ir}, C_i)\ *\ AV_i}{n_i}$ (4.17)

DS_i is the average dissimilarity of the data points of cluster C_i to its W_i.

$w\ (\ AV_{ir}, C_i)$ contains the weights of attribute-value pairs.

Similarly increase in dissimilarity for j[th] cluster and merged cluster of i and j are calculated after estimating $f(AV_{ij})$ of the newly formed cluster.

d) Accordingly prepare a ranking list of cluster pairs with minimum increase in dissimilarity in ascending order.

Step 3 : Identify the cluster pairs (C_i, C_j) with better ranking in both the lists.

Step 4 : Merge C_i and C_j to form the new cluster, C_k and compute the Cluster

Feature for C_k and delete C_i and C_j .

Step 5 : Repeat steps 1 to 4 until no cluster pair is mergable.

4.12 Cluster Refresh

When there is a difference in the growth pattern clusters between the current and the last clustering snapshot i.e some of the clusters may have variation in cluster prototypes due to inclusion of new data points. Hence, it requires to update the cluster prototype ($\overrightarrow{m_i}$, $\overrightarrow{w_i}$) and also the list of p-farthest points Q_i. $\overrightarrow{m_i}$ represents the updated mean of the cluster C_i which is checked for its deviation (as discussed in chapter 3 section 3.10) from $\overrightarrow{m_i}$ and if the deviation in mean is greater than δ, which is user defined then the prototype of the cluster ($\overrightarrow{m_i}$, $\overrightarrow{w_i}$) and obviously the Cluster Feature will be updated as follows:

(1) n_i = updated number of data points in cluster C_i

(2) $\overrightarrow{m_i}$ = $\overrightarrow{m_i}'$

(3) $\overrightarrow{w_i}$ = Recalculated based on $\overrightarrow{AV_{ir}}'$

(4) Q_i = Updated list of p-farthest points of cluster C_i with respect to the new prototype

(5) $\overrightarrow{SS_i}$ = Updated squared sum vector as it changes due to incremental update

(6) $\overrightarrow{AV_{ir}}'$ → updated count of all attribute value pairs occurred in i^{th} cluster

Cluster refresh happens based on deviation in mean which is continuously monitored whenever a new chunk of data points are added to the existing cluster solution. Though it is monitored globally, refresh happens only on selected clusters based on deviation of their prototypes.

4.13 Hypothetical dataset

As bench mark datasets involve mostly entities with uniformly distributed clusters, the author has created a hypothetical dataset to see how the proposed algorithms would perform on entities with non-uniformly distributed clusters.

The hypothetical dataset contains entities which give the geographical location of a habitat. It is well known that all the land mass available is not suitable for habitation because some geographical and political conditions like mountain areas, canals, forest areas, defense areas obstruct habitat. Also all areas may not be amicable for proper habitation. This example would help us to identify where habitats are densely populated and where they are sparsely populated and find out where a new habitat would be likely to come - in the dense or sparse areas. Generally development of a new habitat is to be encouraged in the sparser areas and discouraged in the denser areas.

Hypothetical dataset has roughly 192 entities with 4 attributes which are as follows:

i) The first two attributes are the co-ordinates of the habitat in terms of their latitude and longitude which are normalized to a scale of 0 to 1.

ii) Type of use – whether it is used for commercial or domestic purposes. This attribute has two distinct values hence it results in two attribute-value pairs namely, COM and DOM.

iii) Type of housing – whether it is an independent house or an apartment. This attribute also has two distinct values hence it results in two attribute-value pairs namely, APT and IND.

The points are as follows:

{0.33, 0.11, COM, APT}, {0.36, 0.16, COM,APT},{0.38, 0.13, COM,APT}
{0.40, 0.18, COM,APT}, {0.41, 0.11, DOM, APT},{0.45, 0.21, COM,APT}
{0.45, 0.15, COM,APT}, {0.47, 0.10, DOM, APT},{0.47, 0.19, COM, IND}

{0.49, 0.23, COM,APT}, {0.49, 0.13, COM,APT}, {0.5, 0.22, DOM, APT}
{0.5, 0.15, DOM, IND}, {0.5, 0.2, COM, IND}, {0.51, 0.24, DOM, APT}
{0.52, 0.12, COM,APT}, {0.52, 0.15, COM,APT},{0.52, 0.17, COM,APT}
{0.52, 0.21, COM,APT}, {0.52, 0.22, COM,APT}, {0.52, 0.25, COM,APT}
{0.52, 0.19, DOM, IND},{0.53, 0.12, COM, IND},{0.54, 0.23, DOM, APT}
{0.54, 0.22, DOM, APT},{0.55, 0.12, COM, IND},{0.55, 0.15, COM,APT}
{0.55, 0.17, COM,APT},{0.55, 0.19, COM,APT}, {0.55, 0.2, COM,APT}
{0.55, 0.21, COM,APT}, {0.55, 0.25, COM, IND},{0.55, 0.13, COM, IND}
{0.55, 0.17, DOM, APT},{0.56, 0.17, DOM, APT},{0.56, 0.2, COM,APT}
{0.56, 0.21, COM,APT},{0.57, 0.23, DOM, APT},{0.57, 0.12, COM,APT}
{0.57, 0.17, COM,APT},{0.57, 0.18, COM,APT},{0.57, 0.2, COM,APT}
{0.57, 0.22, DOM, IND},{0.58, 0.15, DOM, IND},{0.58, 0.20, DOM, IND}
{0.58, 0.18, DOM, IND},{0.6, 0.2, DOM, IND},{0.65,0.32,COM,APT}
{0.65,0.34,COM,APT},{0.65,0.37,COM,APT},{0.66,0.41,COM,APT}
{0.67,0.31,DOM, APT},{0.67,0.32,COM,APT},{0.67,0.35,COM,APT}
{0.67,0.36,DOM, APT},{0.67,0.37,DOM, APT},{0.68,0.44,COM,APT}
{0.68,0.41,COM,APT},{0.68,0.38,COM,APT},{0.69,0.34,DOM, APT}
{0.7,0.3,DOM, IND},{0.7,0.32,COM, IND},{0.7,0.35,DOM, APT}
{0.7,0.37,DOM, APT},{0.71,0.36,DOM, APT},{0.71,457,DOM, APT}
{0.71,0.3,DOM, APT},{0.72,0.43,DOM, APT} ,{0.72,0.3,DOM, APT}
{0.72,0.32,DOM, APT},{0.72,0.37,DOM, APT} ,{0.73,0.34,DOM, APT}
{0.73,0.41,DOM, APT},{0.74,0.36,COM, IND},{0.74,0.45,COM, IND}
{0.75,0.3,COM, IND},{0.75,0.32,COM, IND},{0.75,0.33,COM, IND}
{0.75,0.35,COM,APT},{0.75,0.4,COM,APT},{0.75,0.42,COM,APT}
{0.75,0.48,COM,APT},{0.75,0.5,DOM, APT},{0.76,0.36,DOM, APT}
{0.76,0.42,DOM, APT},{0.77,0.43,DOM, APT},{0.77,0.32,DOM, APT}

{0.77,0.35,DOM, APT},{0.77,0.32,DOM, APT},{0.77,0.37,DOM, APT}
{0.77,0.42,DOM, APT},{0.79,0.35,DOM, APT},{0.79,0.52,DOM, APT}
{0.79,0.39,DOM, APT},{0.8,0.47,DOM, APT},{0.8,0.49,DOM, APT}
{0.81,0.5,DOM, APT},{0.82,0.52,DOM, APT},{0.82,0.42,DOM, APT}
{0.84,0.48,DOM, APT},{0.87,0.47,DOM, APT},{0.3,0.4,DOM, IND}
{0.30,0.39,COM,APT},{0.31,0.43,COM,APT},{0.32,0.36,COM,APT}
{0.32,0.4,DOM, IND },{0.32,0.47,COM,APT},{0.33,0.38,COM,APT}
{0.33,0.52,COM,APT},{0.33,0.41,DOM,APT},{0.34,0.56,DOM,IND}
{0.34,0.35,COM,APT},{0.34,0.43,DOM,IND},{0.35,0.35,COM,APT}
{0.35,0.36,COM,APT},{0.35,0.37,COM,APT},{0.35,0.4,DOM, IND}
{0.36,0.38,COM,APT},{0.36,0.41,DOM,IND},{0.37,0.43,DOM,APT}
{0.37,0.35,COM,APT},{0.37,0.37,COM,APT},{0.37,0.4,DOM, IND}
{0.37,0.42,DOM,APT},{0.37,0.47,DOM,APT},{0.38,0.38,COM,APT}
{0.39,0.41,DOM,IND},{0.4,0.35,COM,APT}, {0.4,0.37,COM,APT}
{0.4,0.4,DOM, IND},{0.4,0.41,DOM, IND}, {0.4,0.42,DOM, APT}
{0.40,0.43,DOM,APT}, {0.41,0.36,COM,APT},{0.41,0.38,COM,APT}
{0.41,0.4,DOM, IND},{0.42,0.37,COM,APT}, {0.42,0.41,DOM,APT}
{0.42,0.38,COM,APT},{0.43,0.4,COM,APT},{0.43,0.42,DOM,APT}
{0.43,0.46,DOM,IND},{0.44,0.49,DOM,IND},{0.45,0.4,DOM,IND}
{0.45,0.42,DOM,APT}, {0.45,0.52,DOM,IND} ,{0.2,0.69,COM,APT }
{0.20,0.72,COM,APT },{0.21,0.67,COM,APT},{0.21,0.7,DOM, IND }
{0.22,0.75,COM,APT},{0.23,0.70,COM,APT},{0.23,0.77,COM,APT}
{0.23,0.66,DOM,APT},{0.24,0.68,DOM,APT},{0.25,0.65,DOM,APT}
{0.25,0.66,DOM,IND},{0.25,0.7,COM,APT },{0.25,0.72,COM,APT}
{0.25,0.72,COM,APT},{0.26,0.66,DOM,IND},{0.26,0.26,COM,APT}
{0.27,0.65,DOM,IND},{0.27,0.68,DOM,APT},{0.27,0.71,COM,APT}

{0.27,0.74,COM,APT},{0.28,0.66,DOM,IND},{0.28,0.77,DOM,APT}

{0.29,0.78,DOM,APT},{0.3,0.65,DOM, IND},{0.3,0.68,COM,APT }

{0.3,0.7,DOM, APT },{0.3,0.75,COM,APT},{0.30,0.69,COM,APT}

{0.31,0.66,DOM,APT},{0.32,0.74,DOM,IND},{0.32,0.68,DOM,IND}

{0.32,0.7,DOM, IND },{0.33,0.69,DOM,IND},{0.33,0.73,DOM,IND}

The graphical representation of the dataset in terms of numerical attributes is shown in Figure 8.

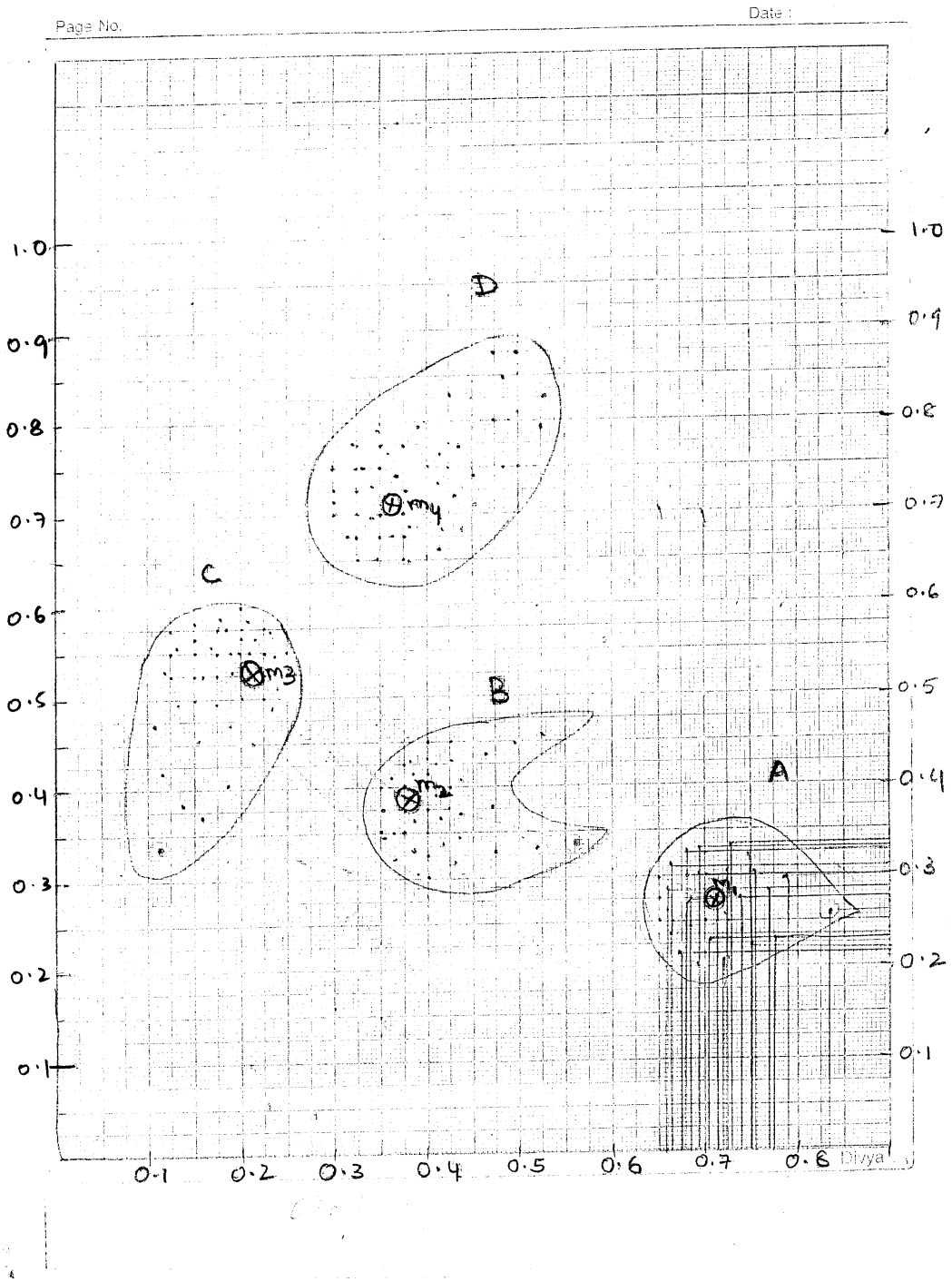
Page No.
Date :
D
C
B
A
m_4
m_3
m_2
m_1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Divya

Fig. 8

From the above, it can be seen that the first attribute is numerical where as the remaining attributes are of categorical nature. The hypothetical dataset is divided into 4 chunks namely hypo1 containing 100 data points, hypo2, hypo3 containing 30 data points each and hypo4 containing the remaining 32 points to check the incremental nature of the algorithms proposed.

M-CFICA is run with the first chunk of 100 data points and the number of clusters (k) is given as 4. After initial clustering of the data points, 4 clusters are formed and the cluster features CF_1 to CF_4 are constructed for each cluster. The clustering solution is represented in the form of Cluster Features as shown below:

Initial clustering solution in terms of CF_1, CF_2, CF_3 and CF_4 on hypo1 dataset....

CF_1

No.of.DataPoints : 35

Mean : {x=0.5206668, y=0.18175}

Weights : {COM=0.49153352, APT=0.571335, DOM=0.23014836, IND=0.18545161}

Farthest : [Numerical Att1 = 0.335 : Numerical Att2 = 0.115 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.368 : Numerical Att2 = 0.16 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.382 : Numerical Att2 = 0.137 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.405 : Numerical Att2 = 0.185 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.45 : Numerical Att2 = 0.215 : Categorical Att1 = COM : Categorical Att2 = APT ||]

No.of.NewDataPoints : 0

New Mean: {}

New Points : No

CF_2

No.of.DataPoints : 20

Mean : {x=0.7400727, y=0.39123636}

Weights : {COM=0.45665053, APT=0.54261553, DOM=0.267809, IND=0.21579823}

Farthest : [Numerical Att1 = 0.66 : Numerical Att2 = 0.415 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.65 : Numerical Att2 = 0.375 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.68 : Numerical Att2 = 0.44 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.681 : Numerical Att2 = 0.418 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.65 : Numerical Att2 = 0.347 : Categorical Att1 = COM : Categorical Att2 = APT ||]
No.of.NewDataPoints : 0

New Mean: {}

New Points : No

CF_3

No.of.DataPoints : 30

Mean : {x=0.3797778, y=0.41146663}

Weights : {DOM=0.37642044, IND=0.23078424, COM=0.33825964, APT=0.51980287}

Farthest : [Numerical Att1 = 0.4 : Numerical Att2 = 0.35 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.41 : Numerical Att2 = 0.362 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.375 : Numerical Att2 = 0.35 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.425 : Numerical

Att2 = 0.375 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.428 : Numerical Att2 = 0.38 : Categorical Att1 = COM : Categorical Att2 = APT ||]

No.of.NewDataPoints : 0

New Mean: {}

New Points : No

CF_4

No.of.DataPoints : 15

Mean : {x=0.27020591, y=0.68964714}

Weights : {COM=0.33577245, APT=0.5219116, DOM=0.36823738, IND=0.21817838}

Farthest : [Numerical Att1 = 0.26 : Numerical Att2 = 0.26 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.3 : Numerical Att2 = 0.65 : Categorical Att1 = DOM : Categorical Att2 = IND ||Numerical Att1 = 0.275 : Numerical Att2 = 0.65 : Categorical Att1 = DOM : Categorical Att2 = IND ||Numerical Att1 = 0.25 : Numerical Att2 = 0.65 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.285 : Numerical Att2 = 0.66 : Categorical Att1 = DOM : Categorical Att2 = IND ||]

No.of.NewDataPoints : 0

New Mean: {}

New Points : No

Now the next chunk (hypo2) is added. These data points are incorporated into one of the existing 4 clusters by looking at the information present in the Cluster Feature. Since

some of the clusters may have variation in data points due to inclusion of data points, the Cluster Feature is updated.

Similarly the remaining chunks of data points (hypo3 and hypo4) are processed and the output set of Cluster Features are as follows:

CF_1, CF_2, CF_3 and CF_4 after adding hypo2, hypo3 and hypo4 datasets .....

CF_1

No.of.DataPoints : 35

Mean : {x=0.5206668, y=0.18175}

Weights : {COM=0.49153352, APT=0.571335, DOM=0.23014836, IND=0.18545161}

Farthest : [Numerical Att1 = 0.335 : Numerical Att2 = 0.115 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.368 : Numerical Att2 = 0.16 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.382 : Numerical Att2 = 0.137 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.405 : Numerical Att2 = 0.185 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.45 : Numerical Att2 = 0.215 : Categorical Att1 = COM : Categorical Att2 = APT ||]

No.of.NewDataPoints : 56

New Mean: {x=0.93316674, y=0.3275}

New Points : [Numerical Att1 = 0.335 : Numerical Att2 = 0.115 : Categorical Att1 = COM : Categorical Att2 = APT || Numerical Att1 = 0.368 : Numerical Att2 = 0.16 : Categorical Att1 = COM : Categorical Att2 = APT || Numerical Att1 = 0.382 : Numerical Att2 = 0.137 : Categorical Att1 = COM : Categorical Att2 = APT || Numerical Att1 = 0.405 : Numerical Att2 = 0.185 : Categorical Att1 = COM : Categorical Att2 = APT || Numerical Att1 = 0.415 :

Numerical Att2 = 0.115 : Categorical Att1 = DOM : Categorical Att2 = APT || Numerical Att1 = 0.455 : Numerical Att2 = 0.155 : Categorical Att1 = COM : Categorical Att2 = APT || Numerical Att1 = 0.47 : Numerical Att2 = 0.109 : Categorical Att1 = DOM : Categorical Att2 = APT || Numerical Att1 = 0.47 : Numerical Att2 = 0.19 : Categorical Att1 = COM : Categorical Att2 = IND ||]

CF_2

No.of.DataPoints : 20

Mean : {x=0.7400727, y=0.39123636}

Weights : {COM=0.45665053, APT=0.54261553, DOM=0.267809, IND=0.21579823}

Farthest : [Numerical Att1 = 0.66 : Numerical Att2 = 0.415 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.65 : Numerical Att2 = 0.375 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.68 : Numerical Att2 = 0.44 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.681 : Numerical Att2 = 0.418 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.65 : Numerical Att2 = 0.347 : Categorical Att1 = COM : Categorical Att2 = APT ||]
No.of.NewDataPoints : 55

New Mean: {x=0.864321, y=0.436218}

New Points : [Numerical Att1 = 0.3 : Numerical Att2 = 0.65 : Categorical Att1 = DOM : Categorical Att2 = IND ||Numerical Att1 = 0.275 : Numerical Att2 = 0.65 : Categorical Att1 = DOM : Categorical Att2 = IND ||Numerical Att1 = 0.25 : Numerical Att2 = 0.65 : Categorical Att1 = COM : Categorical Att2 = APT || Numerical Att1 = 0.382 : Numerical Att2 = 0.137 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.405 : Numerical Att2 = 0.185 : Categorical Att1 = COM : Categorical Att2 = APT ||]

CF_3

No.of.DataPoints : 45

Mean : {x=0.3797778, y=0.41146663}

Weights : {DOM=0.37642044, IND=0.23078424, COM=0.33825964, APT=0.51980287}

Farthest : [Numerical Att1 = 0.4 : Numerical Att2 = 0.35 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.41 : Numerical Att2 = 0.362 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.375 : Numerical Att2 = 0.35 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.425 : Numerical Att2 = 0.375 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.428 : Numerical Att2 = 0.38 : Categorical Att1 = COM : Categorical Att2 = APT ||]

No.of.NewDataPoints : 47

New Mean: {x=0.8497778, y=0.6364666}

New Points : [Numerical Att1 = 0.45 : Numerical Att2 = 0.215 : Categorical Att1 = COM : Categorical Att2 = APT || Numerical Att1 = 0.49 : Numerical Att2 = 0.235 : Categorical Att1 = COM : Categorical Att2 = APT ||]

----------CF_4

No.of.DataPoints : 15

Mean : {x=0.27020591, y=0.68964714}

Weights : {COM=0.33577245, APT=0.5219116, DOM=0.36823738, IND=0.21817838}

Farthest : [Numerical Att1 = 0.26 : Numerical Att2 = 0.26 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.3 : Numerical Att2 = 0.65 : Categorical Att1 = DOM : Categorical Att2 = IND ||Numerical Att1 = 0.275 : Numerical Att2 = 0.65 : Categorical Att1 = DOM : Categorical Att2 = IND ||Numerical Att1 = 0.25 : Numerical Att2 = 0.65 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.285 : Numerical Att2 = 0.66 : Categorical Att1 = DOM : Categorical Att2 = IND ||]

No.of.NewDataPoints : 34

New Mean: {x=368925, y=724532}

New Points : [Numerical Att1 = 0.41 : Numerical Att2 = 0.362 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.375 : Numerical Att2 = 0.35 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.425 : Numerical Att2 = 0.375 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.428 : Numerical Att2 = 0.38 : Categorical Att1 = COM : Categorical Att2 = APT || Numerical Att1 = 0.425 : Numerical Att2 = 0.375 : Categorical Att1 = COM : Categorical Att2 = APT ||Numerical Att1 = 0.428 : Numerical Att2 = 0.38 : Categorical Att1 = COM : Categorical Att2 = APT ||]

It can be noted from above, that CF_1 and CF_3 have significant deviation in mean from their prototypes and hence require cluster refresh.

The M-CFICA algorithm can handle datasets with uniformly distributed clusters similar to conventional incremental clustering algorithms and can specifically deal with datasets with non-uniformly distributed clusters as seen from the above example.