CORE Metadata, citation and similar papers at core.ac.uk
Provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 70 (2015) 226 – 237
Procedia Computer Science

4thInternational Conference on Eco-friendly Computing and Communication Systems (ICECCS)

K-Medoid Clustering for Heterogeneous DataSets

Sandhya Harikumar[a], Surya PV[b]

[a]Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, Kollam, 690525, India
[b]Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham , Kollam, 690525, India

Abstract

Recent years have explored various clustering strategies to partition datasets comprising of heterogeneous domains or types such as categorical, numerical and binary. Clustering algorithms seek to identify homogeneous groups of objects based on the values of their attributes. These algorithms either assume the attributes to be of homogeneous types or are converted into homogeneous types. However, datasets with heterogeneous data types are common in real life applications, which if converted, can lead to loss of information. This paper proposes a new similarity measure in the form of triplet to find the distance between two data objects with heterogeneous attribute types. A new k-medoid type of clustering algorithm is proposed by leveraging the similarity measure in the form of a vector. The proposed k-medoid type of clustering algorithm is compared with traditional clustering algorithms, based on cluster validation using Purity Index and Davies Bouldin index. Results show that the new clustering algorithm with new similarity measure outperforms the k-means clustering for mixed datasets.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Organizing Committee of ICECCS 2015

Keywords: Clustering; Heterogeneous datasets; L_1 norm; K-Medoid; Probabilistic Computation;

1. Introduction

Data analysis is one of the critical phases of Knowledge Discovery in Databases [1] because different approaches such as exploratory, statistical, predictive, etc analyzes data with different perspectives and thus the information obtained is visualized and interpreted in different forms [2]. Real-world datasets are often heterogeneous, represented by a set of mixed attribute data types like numerical, categorical and binary. Banks, financial sectors, insurance policies, stock markets, medical domains and biological domains have a strong urge for data clustering which is a common technique used in data analysis and is used in many fields including statistics, data mining, and image analysis. One of the main requirements of any clustering algorithm is a good similarity measure to know the distance between the objects in order to group them together. Lots of research have been done on the distance measures between objects of homogeneous data types. However for two objects having dissimilar or mixed types of attributes, still a gap persists as

*Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: sandhyaharikumar@am.amrita.edu(Sandhya Harikumar), surivijay26@gmail.com(Surya PV)

1877-0509 © 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Organizing Committee of ICECCS 2015
doi:10.1016/j.procs.2015.10.077

to how to compare the two objects for similarity. In case of big data, the volume of data is too huge and the structure is quite unordered. For analytical purposes, a structured format is given to the data. But all the data types cannot be converted into homogeneous types.

Table 1. An instance of Bank Dataset.

Id	Duration	Job	Marital	Education	Housing	c
1	261	Management	Married	Tertiary	Yes	c1
2	151	Technician	Single	Secondary	Yes	c1
3	968	Technicain	Married	Secondary	Yes	c1
4	375	Technician	Single	Tertiary	No	c2
5	355	Services	Divorced	Primary	Yes	c2

Consider the sample of a bank dataset as shown in Table 1. Here duration is one of the numerical attributes. It has a natural ordering, for instance, 261 > 151. Discretizing such numeric values may assign 151 and 256 the same categorical value. Thus, such an attempt leads to loss of information. $Marital\ status$ is another attribute which is categorical. There is no ordering between the values $Married$, $Single$ and $Divorced$. So it is very difficult to convert the categorical values like $Marital\ status$ or Job to a numeric value. $Housing$ is a binary attribute which has only two possible values yes and no. These can be just checked for equality and no other ordering like $Yes < No$ or $Yes > No$ is possible. Moreover, binary attributes follow a bernoulli distribution but categorical attributes follow a discrete probability distribution.[1]. Thus each of the numeric,categorical and binary attribute types has disparate characteristics and hence should be treated separately. So, if the distance between two data points say row 1 and row 2 is to be computed, it is not possible until a unified model of distance measure for such data exists.A methodology to cluster objects having different types of attributes is addressed in this paper. This type of work is still in modest phase and various strategies employing this type of measure need to be devised for effective and valid clustering.
The main contribution of this paper is a modified version of k-medoid clustering for enabling clustering of mixed as well as pure numeric, categorical and binary datasets. For this, we introduce a similarity measure in the form of a triplet. The proposed work adopts the probability based similarity measure for categorical pairs, together with the L_1 norm[3] for numeric pairs and hamming distance measure for binary pair of attributes. The correctness of this distance measure is established through experimental results. The objective function of K-medoid[8] is to find a non-overlapping set of clusters such that each cluster has a most representative point, called medoids. Medoids are most centrally located with respect to some distance measure. However, finding a better medoid is a computationally expensive task that requires trying all points which are currently not medoids. So, apart from a new distance measure, this paper also employs a greedy strategy in the initial phase of clustering to find the most potential medoids which can form good representatives for clusters in the final phase.

2. Related Work

Any clustering algorithm requires a generalized cost function that works for mixed as well as numeric or categorical datasets. The distance function for numeric datasets is not applicable for categorical data and vice versa because there is no natural distance between two categorical data points. Also, traditional hierarchical clustering algorithms are not scalable to very large databases because of their high computational cost[7]. Various strategies have been employed such as assigning numerical values to categorical data and applying numerical distance measures or discretization of numerical values into categorical form and then applying categorical distance measure. But in high dimensional datasets, employing such a method is inefficient as explained in the previous section. Therefore, work on clustering mixed categorical and numeric datasets is directed towards a different cost measure. Huang's cost function[5] is one such attempt where for representing cluster centers of categorical attributes, mode value is considered. As a result, only one attribute acts as the cluster center. Hamming distance measure is considered for clustering binary data

[1] http://en.wikipedia.org/wiki/Categorical_distribution

points. $\delta(p, q) = 0$ if p = q, which can be considered meaningful. But $\delta(p, q) = 1$ if p≠q. $\delta(p, q)$ is different for different data points. It is dependent on the relative frequency of data points within a class[6]. So for computing $\delta(p, q)$, a co-occurrence based approach is considered. For categorical attribute values, user-defined weight value is assigned. Incorrect assignment of values leads to incorrect clustering results. Amir et.al[6] proposed a generalized cost function for clustering datasets by modifying the Huangs cost function. The work alleviates the short-comings of[5]. For categorical data points, the contribution of attributes towards cluster formation is calculated by a co-occurrence based probabilistic approach.

The idea of defining cluster center for categorical datasets by finding the dominant data items is discussed in[7]. A new distance calculation based on entropy which replaces the Manhattan distance calculation is used. Similar to numerical datasets, categorical datasets also suffer from the curse of high dimensionality. The approach used in K-meansCD[11] is FBC(Frequency Based Center) and a new distance function to perform K-means clustering on categorical datasets is proposed. Work proposed in[7] utilizes FBC concept of[11] to find new center for categorical datasets.

In most of the clustering algorithms, for clustering high dimensional datasets, the initialization of cluster centers is computationally complex. Work adopted in[8] considers all the data points for center initialization. The work presented in[9] solves this problem by selecting random points using greedy approach. This idea of initialization of cluster centers is employed in[10]. The distance measure for clustering numeric data set is Manhattan distance as opposed to Euclidean distance. The work presented in[11] discusses various variants of K-means clustering algorithm for clustering binary dataset. The paper provides efficient distance computation for sparse binary vectors, sparse matrix operations and summary table of clustering results. K-medoid is more flexible and robust to outliers than the K-means approach and has been demonstrated in various work such as[18,19,20]. However the core part is to choose the medoids in a dataset which can form the representatives of the clusters. Though different approaches are described in[12,13], no approach focuses on heterogeneous datasets.

3. Proposed Distance Measure

We propose a generalized distance function in the form of a triplet which works for mixed datasets for enabling Clustering. This triplet consists of three different distance measures for numeric, categorical and binary data types. In our proposed approach we use L_1 norm for distance calculation and medoid selection of numeric attributes, since L_2 norm is sensitive to outliers[3]. A small perturbation in the dataset may change the result drastically if L_2 norm is used. For binary attributes, Hamming distance is used. For categorical attributes, a probabilistic approach based on Modified Huang's cost function[6] is used. Categorical attributes are treated separately from binary attributes due to the variations in the probability distributions of the two data types. Moreover each binary attribute has only two possible outcomes, but categorical attributes have at least 3 possible outcomes and hence the complex computations of computing the probability of each attribute value can be avoided if binary attributes are treated separately. Before applying the distance measure for similarity amongst the data points, the numeric attributes are normalized as in (1).

$$x_{new} = (x - x_{min})/(x_{max} - x_{min}) \tag{1}$$

The proposed similarity measure has three distinct components, one for handling numeric attributes, second for handling categorical attributes and third for handling binary attributes. For each component, lower distance value indicates higher similarity and the distance between any two attribute values are in the same range.

3.1. Distance Measure for numeric attributes

In most of the clustering algorithms the distance between two records, with numerical attributes, is calculated with the help of norms[1]. For a record r_i and a record r_j, with m_r numeric attributes, L_p (p-norm distance) is defined as in (2).

$$L_p = \left(\sum_{k=1}^{mr} \left|r_{i_k} - r_{j_k}\right|^p\right)^{\frac{1}{p}} \tag{2}$$

When $p = 1$, it is L_1 norm, when $p = 2$ it is L_2 norm and so on. The L_1 norm is the absolute difference between each attribute of *two* records. L_1 norm is flexible, robust and resistant to outliers. Also it is computationally efficient in high dimensional data due to the inherent sparsity in high dimensional data.

3.2. Distance Measure for Categorical attributes

Probabilistic Approach: Let x and y be two categorical values of attribute A_i. Inorder to find distance between x and y a co-occurrence based approach [6] is used. Here, probability of occurrence of the two attribute values with other categorical attribute values of the dataset is computed. The following two probabilities are computed:

1) The probability of occurrence of x of A_i with a particular set of attributes w of A_j.
2) The probability of occurrence of y of A_i with a particular set of attributes $\neg w$ of A_j.
So, the distance between the pair of values x and y of A_i with respect to the attribute A_j and a particular subset w, is defined as:

$$\delta^{ij}(x, y) = P_i(w/x) + P_i(\neg w/y) \tag{3}$$

where x is the subset w of values of A_i that maximizes the quantity $P_i(w/x)+P_i(\neg w/y)$. To restrict $P_i(w/x)+P_i(\neg w/y)$ between zero and one $\delta^{ij}(x, y)$ is modified as in (4)

$$\delta^{ij}(x, y) = P_i(w/x) + P_i(\neg w/y) - 1 \tag{4}$$

3.3. Distance Measure for Binary attributes

For binary attribute, Hamming distance is taken into consideration. Binary distance between two boolean attribute values x and y is taken as $\delta(x, y) = 0$ for x=y and $\delta(x, y) = 1$ for x $\neq$ y.

3.4. Vector based distance measure

Let $R = (r_1, r_2, ..., r_N)$ be the set of datapoints with each r_i being described by $A = (a_1, a_2...., a_m)$ set of m attributes. In the context of database, each point (feature vector) of a cluster is in fact a record r_i and each dimension (feature) is an attribute a_i. Let m_r be the number of numeric attributes, m_c be the number of categorical attributes and m_b be the number of binary attributes.

Let r_i and r_j be two objects of R. The distance between the two objects is represented as $< \tilde{dn}, \tilde{dc}, \tilde{db} >$ where $\tilde{dn}$ represents the numeric distance, $\tilde{dc}$ represents the categorical distance, $\tilde{db}$ represents the binary distance. A single measure for the distance between r_i and r_j is given as

$$\vartheta(r_i, r_j) = \Sigma_{k=1}^{m_r} \left| r_{i_k} - r_{j_k} \right| + \Sigma_{t=1}^{m_c} \delta_c(r_{it}^c, r_{jt}^c) + \Sigma_{t=1}^{m_b} \delta_b(r_{it}^b, r_{jt}^b) \tag{5}$$

where m_r is the number of numeric attributes, m_c is the number of categorical attributes and m_b is the number of binary attributes (ie, Total attributes $m = m_r + m_c + m_b$)

4. Distance Computation between a pair of attribute values

The previous approaches for clustering categorical data points were based on Hamming distance where $\delta(p, q)$ is taken as 1 when p$\neq$q. However, for categorical attributes, the distance is a function of distribution of values. So we can conclude that employing Hamming distance measure for computing distance between two categorical data points is inappropriate.
A co-occurrence based approach is devised similar to[6] where the distance between any two pairs of categorical points is computed based on overall distribution of values in the dataset.

4.1. Co-occurrences of Categorical data points

Consider the rows of bank dataset shown in table 1. Here Housing is binary attribute. Job, Marital and Education attributes are categorical. Duration attribute is numeric. Row id is explicitly included by us to distinguish between the data points. For computing the distance between any pair of categorical attribute values, conditional probabilities of the corresponding categorical attributes values with other attribute columns needs to be computed. The probability table for computing distance between job attribute is given in table 2:

Table 2. Probability Table.

Co-occurrence of categorical attribute values	
P(married/management) = 1	P(tertiary/management) = 1
P(single/technician) = 2/3	P(secondary/technician)= 2/3
P(married/technician) = 1/3	P(tertiary/technician) = 1/3
P(divorced/services) = 1	P(primary/services) = 1
P(management/married)=1/2	P(tertiary/married) = 1/2
P(technician/single) = 1	P(secondary/single) = 1/2
P(technician/married) = 1	P(secondary/married) = 1/2
P(services/divorced) = 1	P(tertiary/single) = 1/2
P(primary/divorced) = 1	

Table 3. Normalised Bank Dataset.

Id	Duration (t)	Job (t)	Marital (t)	Education (t)	Housing (t)	c (t)
1	0.13	Management	Married	Tertiary	Yes	c1
2	0	Technician	Single	Secondary	Yes	c1
3	1	Technicain	Married	Secondary	Yes	c1
4	0.27	Technician	Single	Tertiary	No	c2
5	0.24	Services	Divorced	Primary	Yes	c2

4.2. Similarity Measure of two Categorical values Using Probabilistic Approach

Here probabilistic distance computation for categorical pair of attribute values is employed. The algorithm as given in[6] is used to find the distance between two categorical values of an attribute. We now illustrate how distance between two attributes values is computed using the conditional probabilities and normalized Bank dataset presented in table 2 and 3 respectively. Consider $Management$ and $Technician$ values of Job attribute.

1. Compute distance between $Management$ and $Technician$ with respect to Marital attribute.
   $\delta^{(Job,Marital)}(Management, Technician)$ =P(married/management) + P(single/technician) - 1 = 1 + 2/3 - 1 = 2/3
2. Compute distance between $Management$ and $Technician$ with respect to Education attribute.
   $\delta^{(Job,Education)}(Management, Technician)$ = P(tertiary/management) + P(secondary/technician) - 1 = 1 + 2/3 - 1 = 2/3
3. Using Probabilistic Distance Function, $\delta_c(management, technician)$ = 2/3 + 2/3 - 1 = 1/3

4.3. Similarity measure of two Binary values

For distance computation between a pair of binary data points, Hamming distance is taken into consideration. Though binary attribute is a special type of categorical attribute, hamming distance is employed here since it follows a bernoulli distribution and hence the distance between two binary values will always be either 1 or 0. Moreover, the number of distinct categorical values in a categorical attribute may change but the number of distinct values in a binary attribute will always remain 2 and hence the distribution will always be bernoulli.

In Bank data set given in Table 3, Housing is a Binary attribute. Distance between data points 1 and 2 w.r.t Housing attribute is:
$\delta(Yes, Yes) = 0$.
Where as, distance between data points 1 and 4 w.r.t Housing attribute is
$\delta(Yes, No) = 1$

4.4. Similarity measure of two Numeric values

For distance computation between a pair of numeric data points, the numeric attribute values should be normalised first. In Bank data set, Duration is a Numeric attribute. Bank dataset after normalization of Numeric attribute is illustrated in table 3

Distance between data points 1 and 2 w.r.t Duration attribute is $|0.13 - 0| = 0.13$. Distance between data points 1 and 4 is $|0.13 - 0.27| = 0.14$.

Thus two objects r_i and r_j are said to be farthest from each other if value of $\tilde{d}n$ is higher and the probabilistic measure $\tilde{d}c$ is higher as well as the hamming distance $\tilde{d}b$ is higher.

5. K-medoid clustering for mixed datasets

The phases involved in the clustering algorithm are as follows:
Step 1 : Initialization Phase
Step 2 : Iterative Phase

- Assign Points
- Evaluate Clusters

Step 3 : Outlier Detection

5.1. Initialization Phase

This phase is geared towards finding a potential set of medoids by a greedy approach. There are several efficient clustering algorithms like K-means, Expectation Maximization(EM) clustering etc. that randomly chooses K centers for forming K clusters. But this method of random initialization of cluster centers may lead to a lengthy convergence time and hence computationally expensive in case of high dimensional data sets. Therefore, we have used an approach as proposed in[9], to find potential medoids which can become representatives of the clusters. The initialization phase of the algorithm to find K clusters, proceeds as follows:

1. Choose a random sample of points of size equal to $S = A.K$, where A denotes a large number.
2. Apply greedy technique to S to obtain a smaller subset of points of size equal to $B.K$, where B denotes a small integer such that $B << A$.

Thus if K clusters are required to be formed, select $B.K$ medoids from the sample set of original records where B is an integer constant. The reduction to the sample set significantly reduces the running time of the *Initialization phase*. We then improve the quality of clusters, using these Medoids, in the Iterative Phase.

5.2. Iterative Phase

In this phase, the quality of clusters is improved by applying Hill Climbing technique. We iteratively improve the quality of clusters in this phase by replacing bad medoids. In this phase, data points are assigned to their respective cluster centers and cluster evaluation is done. For evaluating the quality of clustering we have considered Davies Bouldin Index(DBI)[16]. The Davies Bouldin criterion is based on a ratio of within cluster and between cluster distances. In our K-medoid formulation, the compactness of the corresponding clusters and the separation between them are the principal parameters that distinguish one cluster from the other. Davies-Bouldin index is one such measure and hence we have chosen that for cluster evaluation. DBI is defined as in (6):

$$DB = 1/K \sum_{j=1}^{K} max_{j \neq i} D_{i,j} \tag{6}$$

where D_{ij} is the within to between cluster distance ratio for the i^{th} and j^{th} clusters as given in (7).

$$D_{i,j} = (\bar{d}_i + \bar{d}_j)/d_{ij} \tag{7}$$

$\bar{d}_i$ is the average distance between each point in the i^{th} cluster and the centroid of the i^{th} cluster. $\bar{d}_j$ is the average distance between each point in the i^{th} cluster and the centroid of the j^{th} cluster. d_{ij} is the distance between the centroids of the i^{th} and j^{th} clusters. The maximum value of d_{ij} represents the worst case within to between cluster ratio for cluster i. The optimal clustering solution has the smallest Davies-Bouldin index value. If the value returned by this evaluation metric is greater than a threshold(taken as 0.4), such medoid is termed as bad medoid and we replace the medoid with a new one from the available list and repeat the iterative phase until convergence. This phase continues until bad medoids are detected.

5.3. Outlier Detection

The final phase of the algorithm takes care of the outliers. The Outlier detection algorithm proposed by us is based on farthest nearest approach. Let D be the set of data, K be the number of clusters and N_k be the number of datapoints in k^{th} cluster. For detecting outliers in a cluster, find the farthest (N_k/K)*0.1 points from medoid m_k. This implies that for each of the clusters, data points that are far from the corresponding medoids are computed. For each of those points, find the locality with respect to the smallest distance from the remaining points. If the locality of a chosen data point (say O_i) contain less than c number of data points in it, then the data point is considered as an outlier, where c is a threshold value. The locality of O_i is defined as the space within distance δ_o, where δ_o is the minimum distance of O_i to the farthest data points chosen. Since our medoid selection is based on distance based approach, it is likely that an Outlier may be chosen as a medoid during the Initialization phase. A medoid in such cases, is considered as bad, if it contains less than $(N/K) * 0.1$ points in it. Since outliers clusters with only minimum number of data points we can find the bad medoids effectively using this criteria. We replace the bad medoids with new points from the medoid list, and again perform the assignment of the points to the medoids. This phase terminates until bad medoids are reported by the algorithm.

5.4. Purity Evaluation of Clusters

The purity measure[21] is an external evaluation criterion that evaluates the quality of the clusters according to the labeled samples available. A cluster is considered pure if it contains labeled objects from one and only one class. Inversely, a cluster is considered as impure if it contains labeled objects from many different classes. Purity is computed as shown in (8):

$$purity(\Omega, C) = 1/N \sum_{k} max_j |C_k \cap w_j| \tag{8}$$

where $\Omega = w_1, w_2.....w_j$ is the set of classes and $C = c_1, c_2, ..., c_k$ is the set of clusters. Bad clusterings have purity values close to 0, a perfect clustering has a purity of 1. This measure is applicable to only labeled samples where the class or labeling of each datapoint is available.

5.5. Main Algorithms

The outline of two main algorithms implemented is presented in this section. Algorithm 1 is K-medoid clustering that includes initialization, iterative and outlier detection phases. The algorithms for Evaluate Clusters is based on DBI measure and AssignPoints is based on the similarity of each datapoint with the medoids. Algorithm 2 is the one for distance computation between two heterogeneous data objects as discussed in the earlier subsection.

Algorithm 1 K-medoid for clustering

```
procedure K-MEDOID CLUSTERING(K,D,A,B)
    Input- K : Number of Clusters, D : Set of Data Points, A : A constant value, B : A small constant value
    Output- Clusters C_1, C_2, ..., C_K
    Begin
    {1. Initialization Phase}
    S = random sample of size A.K
    {M = Set of potential medoids of size B.K {m_1, m_2, ...} computed from S by a greedy strategy}
    M = Greedy(S, B.K)

    {2. Iterative Phase}
    BestObjective = ∞
    M_current = Choose randomly {m_1, m_2,... m_k} ⊂ M
    repeat
        {Assign each datapoint to a medoid in M_current based on the similarity measure}
        C = AssignPoints(M_current, D)
        where { C = {C_1, C_2, ..., C_K} is the set of clusters}


        ObjectiveFunction = EvaluateClusters(C_1,C_2...C_K)
        if (ObjectiveFunction < BestObjective) then
            BestObjective = ObjectiveFunction
            M_best = M_current
            Compute bad medoids in M_best
            if (ObjectiveFunction >= threshold) then
                M_current = M_best ∪ m where m ∈ Mand m ∉ M_current
            end if
        end if
    until termination condition
    return M_best
    End
end procedure
```

6. Experimental Analysis

For experimental purpose, four real datasets have been taken. One is mixed, second is purely numeric, third is purely categorical and the fourth one is purely binary. The cluster evaluation is based on purity index and davies bouldin index.

6.1. Cluster Evaluation using Australian Credit Dataset

This is a mixed dataset having 690 data points defined by 14 attributes of which 6 attributes are numeric, 3 are binary and 5 are categorical. The data is about credit card applicants who were given approval and who were not. Thus the data belong to two different classes namely, positive(309 data points) and negative (381 data points). Cluster evaluation for Australian Credit Card Dataset with proposed algorithm and K-means algorithm is as shown in Table 4 and 5 respectively. The best value of DBI obtained for Credit dataset using proposed algorithm is 0.38 for $K = 2$. The value of DBI started increasing when value of K was increased. So it indicates that K = 2 is the optimum number of clusters for Credit dataset. The purity value of the clusters formed using mixed K-means algorithm is 0.882 but with our proposed approach is 0.902 which indicates the increased cluster compactness.

Algorithm 2 Distance between two data objects

```
procedure DistanceComputation(r_i,r_j)
    Input- Two data points each with m_c categorical attributes, m_b binary attributes and m_r numeric attributes
    Output- Distance between two data points
    Begin
    Initialize δ(x, y),δ(a, b),δ(u, v) to 0
    for each attribute A_i do
        if A_i is categorical attribute then
            Initialize catSum to 0
            for pair of categorical values(x,y) of A_i do
                for every categorical attribute A_j ≠ A_i do
                    Compute δ^{ij}(x, y)
                    catSum = catSum +δ^{ij}(x, y)
                end for
                δ(x, y) = δ(x, y)+catSum − 1
            end for
        end if
        if A_i is numerical attribute then
            for pair of numeric values(a,b) of A_i do
                d(a, b) = Compute L_1(a, b)
                δ(a, b) = δ(a, b) + d(a, b)
            end for
        end if
        if A_i is binary attribute then
            for pair of binary attribute values(u,v) of A_i do
                d(u,v)=1,if u≠v else d(u,v)= 0
                δ(u, v) = δ(u, v) + d(u, v)
            end for
        end if
    end for
    return Sum = δ(x, y) + δ(a, b) + δ(u, v)
    End
end procedure
```

6.2. Cluster Evaluation using Zoo Dataset

This is a pure binary dataset having 101 data points defined by 16 attributes of which 41 data points belong to true class and 60 belongs to false class. Cluster evaluation for Zoo dataset with proposed algorithm and mixed K-means is as shown in Table 6 and Table 7 respectively.

Table 4. Cluster evaluation for Australian Credit Card Dataset with proposed algorithm.

Cluster No.	Credit(Positive) (*t*)	Credit(Negative) (*t*)
1	296	54
2	13	327
	Purity = 0.9028	**DBI = 0.38**

Table 5. Cluster evaluation for Australian Credit Card Dataset with mixed K-means algorithm [6].

Cluster No.	Credit(Positive) (*t*)	Credit(Negative) (*t*)
1	288	62
2	19	321
	Purity = 0.882	

Table 6. Cluster evaluation for Zoo Dataset with proposed algorithm .

Cluster No.	True (*t*)	False (*t*)
1	41	0
2	0	60
	Purity = 1	**DBI = 0.447**

Table 7. Cluster evaluation for Zoo with mixed K-means algorithm [6].

Cluster No.	True (*t*)	False (*t*)
1	33	8
2	8	52
	Purity = 0.841	

K-means algorithm for Mixed datasets treats binary attributes as categorical unlike our proposed approach where Hamming distance measure is employed for measuring similarity. Using proposed approach a purity value of 1 is obtained which indicates high quality of clustering. Purity value of 1 shows there is no misclassification of data-points. Best DBI value obtained is 0.447. DBI value started increasing when K value was increased from 2. Using K-means algorithm the purity value obtained is 0.841 which is worse than our proposed K-medoid algorithm that has considered binary as a different data type than categorical.

6.3. Cluster Evaluation for Bank dataset

Bank dataset consists of 45210 records and 17 attributes. In this dataset, eight are numeric, four are binary and five are categorical. The corresponding numeric dataset after preprocessing is also considered for evaluation.

For Bank dataset, the Purity value for Heterogeneous and Numeric datasets using K-means remained constant. But in our proposed approach, the heterogeneous dataset if clustered gave better purity and dbi than the corresponding numeric dataset. We sampled and replicated Bank dataset for time analysis of the algorithm as shown in Figure 1a. The purity evaluation of the clusters formed from Bank dataset revealed that K = 3 gives the pure clusters as per the labeled samples of the bank dataset as shown in Figure 1b

6.4. Cluster Evaluation using Vote Dataset

This is a pure categorical dataset having 435 data points defined by 16 attributes. The elements belong to two different classes namely, Republican(170 data points) and Democrats(265 data points). The cluster evaluation result

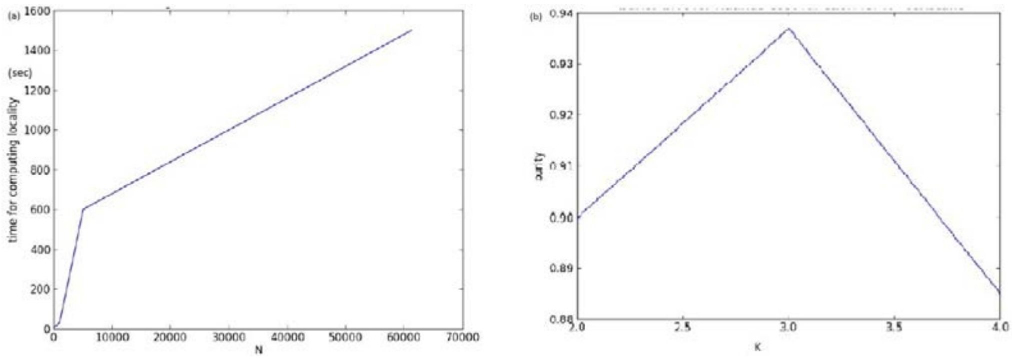

Fig. 1. (a) : Time analysis with varying N and K=2 ; (b)Purity values for different values of K with N=5000 records of bank dataset using proposed K-medoid algorithm

of Vote dataset with $K = 2$ using our proposed approach and Mixed K-means Clustering [6] is in Table 8 and Table 9 respectively.

The comparison of the proposed algorithm and mixed K-means algorithm on the basis of purity evaluation is shown in Figure 2a and the consolidated results of purity and dbi measures of various datasets using the proposed K-medoid algorithm is shown in Figure 2b. It reveals that mixed Bank dataset has better clustering than the preprocessed numeric Bank dataset, since there is a loss of information while data is preprocessed which may affect the quality of clusters formed.

Table 8. Cluster evaluation for Vote Dataset with proposed algorithm after Outlier removal.

Cluster No.	Republican (*t*)	Democrat (*t*)
1	157	20
2	9	225
	Purity = 0.93	**DBI = 0.181**

Table 9. Cluster evaluation for Vote Dataset with mixed K-means algorithm [6] .

Cluster No.	Republican (*t*)	Democrat (*t*)
1	141	25
2	6	200
	Purity = 0.91	

7. Conclusion

In this paper, we proposed a variant of K-medoid clustering for heterogeneous datasets with varied data types. A distance measure to compute the similarity between two objects with varied data types is formulated and this measure has been employed to devise a new algorithm for k-medoid clustering. K-medoid clustering algorithm for heterogeneous datasets has relevance in various commercial, financial and medical sectors. The performance of the algorithm has been improved and good clusters have been formed due to the improvised initialization phase, DBI based evaluation and new outlier detection. The purity and DBI index values computed on different datasets show that our algorithm outperforms K-means algorithm for mixed dataset.

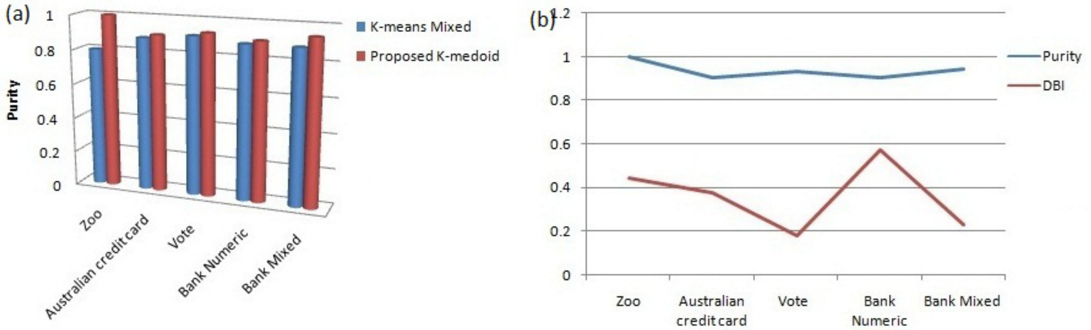

Fig. 2. (a): Comparison of Purity evaluation of various datasets using Proposed K-medoids and mixed K-means [6] algorithm; (b): Purity and DBI evaluation of clusters formed by the Proposed K-medoid algorithm

8. Acknowledgement

We are highly grateful to Dr. M.R. Kaimal, Chairperson, Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham. His valuable suggestions and support through out the work has helped us in improving the work and standard of the paper.

References

1. Usama Fayyad,Gregory, Smyth,"Knowledge Discovery and Data Mining: Towards a Unifying Framework",AAAI Press, pp : 82-88, 1996
2. Trevor Hastie, Robert Tibshirani,Jerome Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer,2009
3. John Hopcroft, Ravindran Kannan,"Foundations of Data Science", 2013.
4. A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review", ACM journal of Computing Surveys, Vol 31, pp: 264-323, NewYork, 1999.
5. Z. Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values", Journal of Data Mining and Knowledge Discovery, Vol 2, pp:283-304, ACM, 1998
6. Ahmad A, Dey L, "A k-means clustering algorithm for mixed numeric and categorical data", Data and Knowledge Engineering Vol. 63, pp:503-527, Elsevier 2007
7. Minho Kim, R.S. Ramakrishna,"Projected clustering for categorical datasets", Pattern Recognition Letters, Vol 27, Elsevier, 2006
8. Hae-Sang Park, Chi-Hyuck Jun, "A simple and fast algorithm for K-medoids clustering", Expert Systems with Applications, Vol 36, Elsevier,2008
9. T. Gonzalez, "Clustering to minimize the maximum intercluster distance", Theoretical Computer Science, Vol. 38, pp. 293-366, 1985.
10. Charu C. Aggarwal, Cecilia Procopiuc, "Fast Algorithms for Projected Clustering", Proceedings of the ACM SIGMOD International Conference on Management of Data, Vol 28, pp: 61-72, ACM, 1999
11. Carlos Ordonez, "Clustering binary data streams with K-means", Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery,pp:12-19, 2003
12. Qian Li and Xiyu Liu, "A K-medoids Clustering Algorithm with Initial Centers Optimized by a P System", Human Centered Computing - First International Conference, HCC 2014,pages:488–500, 2014
13. Monica Sood,Shilpi Bansal, "K-Medoids Clustering Technique using Bat Algorithm", International Journal of Applied Information Systems (IJAIS) ISSN : 2249-0868, Vol. 5 - No 8.
14. Robson L. F. Cordeiro, Agma J. M. Traina, Christos Faloutsos, Caetano Traina Jr., "Finding Clusters in Subspaces of Very Large, Multidimensional Datasets", ICDE Conference, 2010
15. Man Lung Yiu, Nikos Mamoulis. "Iterative Projected Clustering by Subspace Mining", IEEE Transactions on Knowledge and Data Engineering, Vol 17, 2005
16. Legany,Csaba,Juhasz,Sandor and Babos,Attila, "Cluster Validity Measurement Techniques", In proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, pp:388-393, WSEAS,2006
17. M.Deepa, P. Revathy, "Validation of Document Clustering based on Purity and Entropy measures", International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 3, May 2012
18. Velmurugan T., "Evaluation of k-Medoids and Fuzzy C-Means clustering algorithms for clustering telecommunication data",International Conference on Emerging Trends in Science, Engineering and Technology (INCOSET), Dec 2012, pp :115-120
19. He Hu and Xiaoyong Du, "Semi-Automatic Online Tagging with K-Medoid Clustering", International Journal of Software Engineering and Knowledge Engineering,Vol 24,No. 8, pp: 1115–1130
20. Rakesh Chandra Balabantaray,Chandrali Sarma and Monica Jha, "Document Clustering using K-Means and K-Medoids", CoRR, 2015.
21. Manning C.D., Raghavan P. Schutze H, "Introduction to Information Retrieval",Cambridge University Press 2008