Pattern Recognition Letters 32 (2011) 1062–1069

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A *k*-means type clustering algorithm for subspace clustering of mixed numeric and categorical datasets

Amir Ahmad [a,*], Lipika Dey [b]

[a] *Faculty of Computing and Information Technology, King Abdulaziz University, Rabigh, Saudi Arabia*
[b] *Innovation Labs, Tata Consultancy Services, New Delhi, India*

ARTICLE INFO

Article history:
Received 3 July 2010
Available online 21 February 2011
Communicated by Debasis Chaudhuri

Keywords:
Clustering
Subspace clustering
Mixed data
Categorical data

ABSTRACT

Almost all subspace clustering algorithms proposed so far are designed for numeric datasets. In this paper, we present a *k*-means type clustering algorithm that finds clusters in data subspaces in mixed numeric and categorical datasets. In this method, we compute attributes contribution to different clusters. We propose a new cost function for a *k*-means type algorithm. One of the advantages of this algorithm is its complexity which is linear with respect to the number of the data points. This algorithm is also useful in describing the cluster formation in terms of attributes contribution to different clusters. The algorithm is tested on various synthetic and real datasets to show its effectiveness. The clustering results are explained by using attributes weights in the clusters. The clustering results are also compared with published results.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The problem of clustering in general deals with partitioning a data set consisting of n points embedded in m-dimensional space into k distinct set of clusters, such that the data points within the same cluster are more similar to each other than to data points in other clusters (Jain and Dubes, 1988). Conventional clustering methods do not work well for high dimensional sparse data as in the high dimensions the distances between any two data objects become same (Agrawal et al., 1998). The subspace clustering is a solution proposed for the high dimensional sparse datasets. The aim of the subspace clustering is to find clusters from the subspaces of the data instead of the entire data space. There are two types of subspace clustering algorithms. First type of subspace clustering algorithms try to find out the exact subspaces of different clusters (Goil et al., 1999; Sequeira and Zaki, 2004; Aggarwal and Yu, 2000). The second type of algorithms cluster data points in the entire data space but the different attributes contribute differently to the different clusters (Friedman and Meulman, 2004; Jing et al., 2007).

Almost all the subspace clustering algorithms proposed so far are designed for numeric data (exception is CLICKS (Zaki et al., 2007)). In this paper, we propose a *k*-means type clustering algorithm for subspace clustering of mixed numeric and categorical dataset. This is a second type of subspace clustering algorithm. Another advantage of this algorithm is its complexity which is linear with respect to the number of the data points. Ahmad and Dey (2007b) propose a *k*-means type clustering algorithm for mixed data. Jing et al. (2007) propose an entropy weighting *k*-means clustering of numeric high dimensional sparse dataset. They propose a formula for computing the contributions of different attributes to different clusters. We combine these two approaches to develop a subspace clustering algorithm for mixed data. A new step has been added to the algorithm proposed by Ahmad and Dey (2007b) to compute the contributions of different attributes (attributes weights) to different clusters. In Section 2, we present related work. In Section 3, we describe our proposed algorithm. In Section 4, we present results and discussion. Section 5 has conclusion and future work.

2. Related work

There are a large number of conventional clustering algorithms for the numeric data (Jain and Dubes, 1988). Many clustering algorithms have been proposed for pure categorical datasets. *k*-modes (Huang, 1997), ROCK (Guha et al., 1999), COOLCAT (Barbara et al., 2002), CACTUS (Ganti et al., 1999) are clustering algorithms for pure categorical datasets. As the clustering algorithms have high complexity, Chen et al. (2008) proposed a sampling-based algorithm to improve the efficiency of these algorithms. They proposed a mechanism to allocate each unlabeled data point into the corresponding appropriate cluster based on the novel categorical clus-

[*] Corresponding author. Tel.: +91 9670296804; fax: +91 1123953449.
E-mail addresses: amirahmad01@yahoo.co.in, amirahmad01@gmail.com (A. Ahmad), lipikadey@gmail.com (L. Dey).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.02.017

tering representative, which represents clusters by the importance of the combinations of attribute values.

With the advent of very large databases containing mixed set of attributes, the data mining community has been on the look-out for good cost function for handling mixed data. Li and Biswas (2002) present a Similarity Based Agglomerative Clustering (SBAC) algorithm based on Goodall similarity measure (Goodall, 1966). The clustering process is hierarchical. This algorithm works well with mixed numeric and categorical attributes, though is computationally expensive. Huang (1997) propose a cost function that considers numeric and categorical attributes separately. The cost function is used in conjunction with a partitional clustering algorithm. The cost function handles mixed datasets and computes the distance between a data point and a centre of cluster in terms of two distance values – one for the numeric attributes and the other for the categorical attributes. Since it can be used with a partitional algorithm, it is cost-effective. Huang et al. (2005) propose a k prototypes clustering method for handling mixed data. In this method, attribute weights are automatically calculated based on the current partition of data. These attributes weights are same for all the clusters. Ahmad and Dey (2007b) propose a k-means type clustering algorithm that overcomes the weaknesses of cost function proposed by Huang (1997).

Many subspace clustering algorithms have been proposed for the numeric datasets. CLIQUE (Agrawal et al, 1998), MAFIA (Goil et al., 1999), and SCHSIS (Sequeira and Zaki, 2004) are the examples of the first type of subspace clustering. Methods like CLIQUE, MAFIA and SCHISM that discretize the numeric data to find out the dense regions rely on the dense region to be ordered, whereas the categorical data has no ordering. Generally, researchers utilize a density threshold to discover the dense regions in all subspaces, Chu et al. (2010) proposed a method that use different density thresholds to discover the clusters in different subspace cardinalities. Chu et al. (2009) suggested a method that avoids redundant clusters. Moise et al. (2009) carried out detailed experimental study of different subspace clustering algorithms. They recommended that techniques with low number of parameters should be preferred.

All these subspace clustering algorithms can handle numeric data only. Zaki et al. (2007) proposed a algorithm CLICKS, that finds subspace clusters in categorical datasets based on a search for k-partite maximal cliques but it is computationally expensive because mining all maximal cliques is an exponential time algorithm in the worst case

Many algorithms have been proposed which fall in the second type of subspace clustering (Friedman and Meulman, 2004; Jing et al., 2007; Deng et al., 2010). In these algorithms different clusters have different sets of contributions of different attributes in a cluster formation. Recently, Jing et al. (2007) proposed a subspace clustering algorithm that computes the contributions of different attributes in formation of different clusters. The experimental results presented in their paper show the superiority of their approach over the other subspace clustering algorithms. The complexity of this algorithm is linear with respect to the number of the data points. That makes it a good candidate for large scale datasets. In the next section, we discuss this algorithm in detail.

2.1. Jing et al. (2007) algorithm

Jing et al. (2007) proposed an entropy weighting k-means algorithm for subspace clustering of high-dimensional numeric data. In their proposed algorithm they have added an extra step in normal k-mean clustering algorithm (MacQuuen, 1967). In this step, they calculate the contribution of each attribute to each cluster. These attribute weights are used to compute the cluster membership of the data points.

In their proposed algorithm, they consider the attribute weight as the probability of contribution of that attribute in forming a cluster. The entropy of contribution of the attributes represents the certainty of contribution of attributes in forming a cluster. They add a new weight entropy term to the k-means clustering algorithm cost function. They include attribute weights to the normal k-mean distance function. They simultaneously minimize the within cluster dispersion and maximize the negative weight entropy. The objective of the first minimization is similar to the k-means clustering algorithm objective whereas the objective of maximizing the negative weight entropy is to induce more attributes to contribute in formation of the clusters. Their cost function is presented in Eq. (2.1).

$$\zeta = \sum_{j=1}^{k} \left[\sum_{i=1}^{n} \sum_{t=1}^{m} \delta_{ji} \lambda_{jt} ((d_{it} - C_{jt}))^2 + \gamma \sum_{t=1}^{m} \lambda_{jt} \log \lambda_{jt} \right] \tag{2.1}$$

subject to

$$\left\{ \begin{array}{l} \sum_{j=1}^{k} \delta_{ji} = 1, \quad 1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant k, \quad \delta_{ji} \in \{0,1\} \\ \\ \sum_{t=1}^{m} \lambda_{jt} = 1, \quad 1 \leqslant j \leqslant k, \quad 1 \leqslant t \leqslant m, \quad 0 \leqslant \lambda_{jt} \leqslant 1 \end{array} \right\}$$

where

d_{it} is the value of tth attribute value of ith data object.
C_{jt} is the value of tth attribute value of jth cluster center.
δ_{ji} is the degree of the membership of the ith data object in jth clusters,
λ_{jt} is the weight of the tth attribute to jth cluster.

The first term is the sum of the within cluster dispersion and the second term represents the negative weight entropy. The positive γ controls the strength of the incentive for clustering on more dimensions. They add one step in the standard k-means clustering algorithm to compute attributes weights. The formula for computing λ_{lt} is given by,

$$\lambda_{lt} = \exp(-\mathrm{D}_{lt}/\gamma) / \sum_{i=1}^{m} \exp(-\mathrm{D}_{li}/\gamma), \tag{2.2}$$

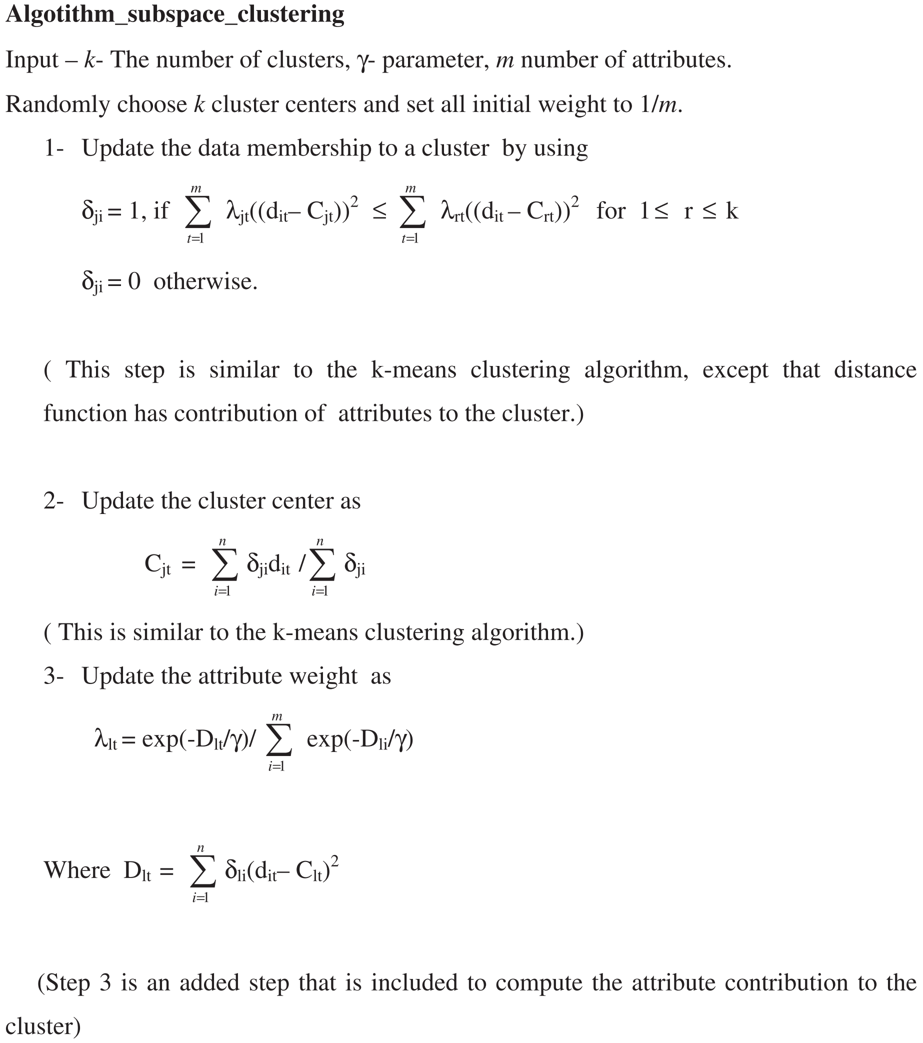
$$\text{where} \quad \mathrm{D}_{lt} = \sum_{i=1}^{n} \delta_{li} (d_{it} - C_{lt})^2.$$

Their proposed algorithm is presented in Fig. 1.

3. The proposed subspace clustering algorithm for mixed datasets

Ahmad and Dey (2007b) proposed a clustering algorithm-based on the k-means paradigm that works well for data with mixed numeric and categorical features. They propose a new cost function and distance measure based on co-occurrence of attribute values. The measure also takes into account the significances of the numeric attributes in the dataset. Ahmad and Dey (2007b) define a cost function for clustering mixed datasets. The proposed cost function with n data objects and m attributes (m_r numeric attributes, m_c categorical attributes, $m = m_r + m_c$ (r or c in subscript or superscript show that the attribute is numeric (r) or categorical(c)) is presented in Eq. (3.1), which is to be minimized. The cost function has two distinct components, one for handling numeric attributes and another for handling categorical attributes.

$$\zeta = \sum_{i=1}^{n} \vartheta(d_i, C_j), \tag{3.1}$$



```
Algotithm_subspace_clustering
Input – $k$- The number of clusters, $\gamma$- parameter, $m$ number of attributes.
Randomly choose $k$ cluster centers and set all initial weight to $1/m$.
    1- Update the data membership to a cluster by using
        $\delta_{ji} = 1$, if $\sum_{t=1}^{m} \lambda_{jt}((d_{it}– C_{jt}))^2 \le \sum_{t=1}^{m} \lambda_{rt}((d_{it} – C_{rt}))^2$ for $1 \le r \le k$
        $\delta_{ji} = 0$ otherwise.
    ( This step is similar to the k-means clustering algorithm, except that distance
    function has contribution of attributes to the cluster.)
    2- Update the cluster center as
        $C_{jt} = \sum_{i=1}^{n} \delta_{ji} d_{it} / \sum_{i=1}^{n} \delta_{ji}$
    ( This is similar to the k-means clustering algorithm.)
    3- Update the attribute weight as
        $\lambda_{lt} = \exp(-D_{lt}/\gamma) / \sum_{i=1}^{m} \exp(-D_{li}/\gamma)$
    Where $D_{lt} = \sum_{i=1}^{n} \delta_{li}(d_{it}– C_{lt})^2$
    (Step 3 is an added step that is included to compute the attribute contribution to the
cluster)
```

Fig. 1. Subspace clustering algorithm for numeric data proposed by Jing et al. (2007).

where $\vartheta(d_i, C_j)$ is the distance of a data object d_i from the closest cluster center C_j.$\vartheta(d_i, C_j)$is defined as

$$\vartheta(d_i, C_j) = \sum_{t=1}^{m_r}(w_t(d_{it}^r - C_{jt}^r))^2 + \sum_{t=1}^{m_c}\Omega(d_{it}^c, C_{jt}^c)^2, \tag{3.2}$$

where $\sum_{t=1}^{m_r}(w_t(d_{it}^r - C_{jt}^r))^2$ denotes the distance of data point d_i from its closest cluster center C_j for numeric attributes, w_t denotes the significance of the tth numeric attribute, which is to be computed from the data set, $\sum_{t=1}^{m_c}\Omega(d_{it}^c, C_{jt}^c)^2$ denotes the distance between data point d_i and its closest cluster center C_j for categorical attributes.

The distance function for categorical attributes does not assume a binary or a discrete measure between two distinct values. Ahmad and Dey (2007a) proposed a method to compute distance between two attribute values of same attribute for unsupervised learning. This approach is based on the fact that similarity of two attribute values is dependent on their relationship with other attributes. The dissimilarity between two categorical values is computed with respect to every other attribute of data set, the average value of distances will give the distance between two categorical values in that data set. This distance measure is used to compute the distance between two categorical values. The other important point in this algorithm is the significance of the numeric attributes. As discussed in (Ahmad and Dey, 2007a, b) an attribute plays a significant role in clustering, provided any pair of its attribute values are well separated against all attributes i.e. have an overall high value of distance for all pairs of attribute values. They exploited this property to compute the significance of an numeric feature. One may discretize numeric features and compute the distance between each pairs. The average value of these distances will be a good indicative of significance of features. However, while calculating the prototype of a cluster and the distance of a data point from the cluster prototype distance, they use non-discretized numeric variables as discretization lead to the loss of information. In other words, in a discretized dataset, all the points in a bin will be similar. One may argue that as they are using discretized dataset for computing a significance of attribute, there is an information loss in that step. We agree that there will be an information loss. However, this step is only to compute a significance of attribute. Hence, two different points of an attribute will give different distances as desired. This algorithm has 3 main steps:

1. It computes the distance between every pair of attribute values of categorical attributes by using co-occurrence based approach proposed by Ahmad and Dey (2007a, b). The distance between the pair of values x and y of attribute A_i with respect to the attribute A_j, for a particular subset w of attribute A_j values, is defined as follows:

$$\delta^{ij}_{w}(x,y) = p(w/x) + p(\sim w/y) - 1. \qquad (3.3)$$

The distance between attribute values x and y for A_i with respect to attribute A_j is denoted by $\delta^{ij}(x,y)$, and is given by $\delta^{ij}(x,y) = p(\omega/x) + p(\sim\omega/y) - 1$, where ω is the subset ***w*** of values of A_j that maximizes the quantity $p(w/x) + p(\sim w/y)$. The distance between x and y is computed with respect to every other attribute. The average value of distances will be the distance $\delta(x,y)$ between x and y in the dataset.

2. It computes the significance of each numeric attribute in the dataset. To compute the significance of a numeric attribute, it follows following steps;
   (i) Discretize the numeric attribute.
   (ii) Compute the distance between every pair of categorical values, using the same method already discussed.
   (iii)
   The average value of these distances is taken as the significance of the attribute.
3. It computes the distance between the data points and the modified centers. In the proposed definition of cluster center, though the central value of a numeric attribute is represented by its mean, they use a different representation for categorical attributes. The centre for a cluster **Ω** for categorical attributes is represented as

$$1/\mathbf{N}_{\Omega}\langle(N_{1,1,\Omega}, N_{1,2,\Omega}, \ldots, N_{1,p1,\Omega}), \ldots (N_{i,1,\Omega}, N_{i,2,\Omega}, \ldots N_{i,k,\Omega}, \ldots, N_{m,pi,\Omega} \ldots (N_{m,1,\Omega}, N_{m,2,\Omega}, \ldots, N_{m,pm,\Omega})\rangle, \qquad (3.4)$$

where $\boldsymbol{N}_{\Omega}$ is the number of data points in the cluster Ω, $N_{i,k,,\Omega}$ denotes the number of the data points in the cluster Ω, which have the kth attribute value for the ith attribute, assuming that ith attribute has pi different values. Thus the cluster center represents the proportional distribution of each categorical value in the cluster. The distance between a data point having ith categorical attribute value X in the ith dimension (ith categorical attribute) and the center of cluster C is defined as

$$\Omega(X,C) = (N_{i,1,c}/N_c) * \delta(X, A_{i,1}) + (N_{i,2,c}/N_c) * \delta(X, A_{i,2}) \cdots + (N_{i,,t,c}/N_c) * \delta(X, A_{i,t}) + (N_{i,pi,c}/N_c) * \delta(X, A_{i,pi}), \qquad (3.5)$$

where $A_{i,t}$ is the tth attribute value of ith categorical attribute

3.1. The proposed algorithm

Motivated by the subspace clustering algorithm proposed by Jing et al. (2007), we propose a modification in the distance measure proposed by Ahmad and Dey (2007b). We include attributes weights to the distance, our proposed distance measure is,

$$\vartheta(d_i, C_j) = \sum_{t=1}^{m_r} \lambda_{jt}(w_t(d^r_{it} - C^r_{jt}))^2 + \sum_{t=1}^{m_c} \lambda_{jt}\Omega(d^c_{it}, C^c_{jt})^2. \qquad (3.6)$$

To compute attributes weights (λ_{jt}), similar to the method proposed by Jing et al. (2007), we add a step in the algorithm proposed by Ahmad and Dey (2007b). Our proposed algorithm for the subspace clustering for the mixed data is presented in Fig. 2.

3.2. The complexity of the proposed algorithm

The computational complexity of Ahmad and Dey (2007b) algorithm is $O(m^2n + m^2S^3 + pn(km_r + km_cS))$ which is linear with respect to the number of the data elements. where n is the total number of the data elements, m is the total number of attributes, m_r is the number of the numeric attributes, m_c is the number of the categorical attributes, and S is the average number of distinct categorical values, k is the number of clusters and p is the number of iterations. As we add one more step (the computation of the attribute contribution to the clusters) that needs the computation of distances from the center, the complexity of this step is $pn(km_r + km_cS)$ so the total complexity is $O(m^2n + m^2S^3 + 2pn(km_r + km_cS))$ which is linear with respect to the number of data elements.

4. Experiments

In this section, we present the clustering results obtained by our approach on some standard datasets. These datasets are taken from the UCI repository (http://www.sgi.com/tech/mlc/db). To judge the quality of clustering, we assume that we are given preclassified data and measure the "overlap" between an achieved clustering and the ground truth classification.

Clustering error = The number of data points not in desired clusters/The number of data points.

The class information is suppressed during clustering. Experiments with low dimensional datasets give better visualization. However, the algorithm has a positive parameter, γ, that controls the strength of the incentive for clustering on more dimensions. There is no rule (Jing et al., 2007) that suggests exactly how many features will take part in cluster formation. Hence, one has to do trail and error method to look for the value of γ that force the desired attributes to to be part of the the cluster. If clusters are using all the features it will become similar to the algorithm (Ahmad and Dey, 2007b). As the purpose of the algorithm is to find out clusters in subspaces, we presented the results on the synthetic data to show that the proposed is able to find out the clusters with large number of irrelevant features. *In these experiments, the clustering error was the main performance criterion; however, the proposed algorithm also gives the insight about the cluster formation. Hence, we also studied the weights of different attributes in different clusters.* All numeric attributes were normalized where the normalized value y of x (x belongs to ith attribute) was obtained as $y = (x - x_{i,\min})/(x_{i,\max} - x_{i,\min})$. The significance of numeric attributes are extracted as discussed in Section 3.2. We used the equal width discretization method for this purpose. We have two parameters one is γ and another is k (number of clusters). We studied the performance of our proposed algorithm different datasets on different values of γ, we found that there was a cluster merging problem in some of datasets for small value of γ ($\gamma < 5$). We got consistent results with $\gamma = 20$. We carried out all the experiments with $\gamma = 20$. For small datasets, we kept $k = 20$ (except for the synthetic dataset), whereas for large data set (mushroom) $k = 40$ was chosen. The large value of k was chosen for better study of clusters formation. For every dataset the clustering algorithm was run 100 times and average results are presented.

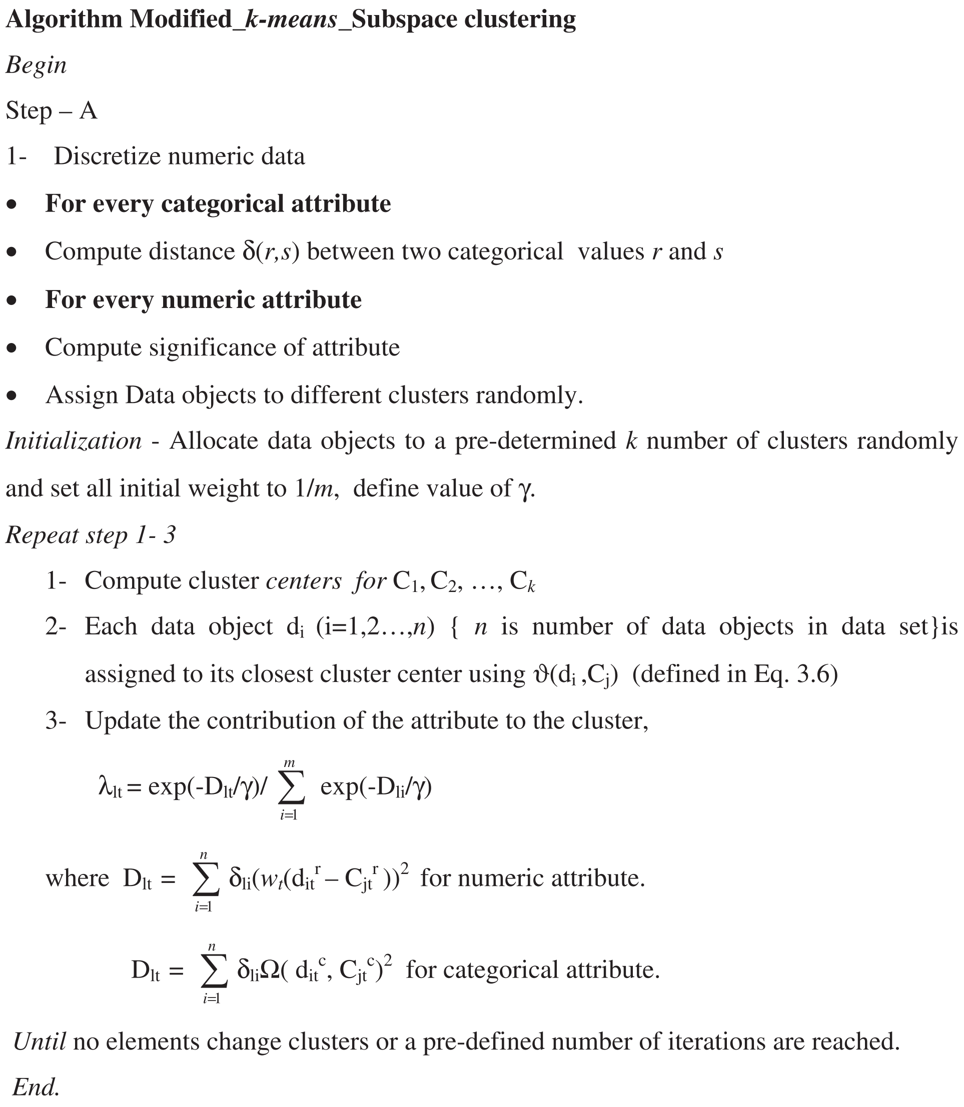
4.1. Synthetic data

We carried out controlled experiment with the Breast Cancer dataset. It has 699 data points described by 9 categorical attributes. It has 2 clusters; Benign (458 data points) and Malignant (241). We did the experiments by adding some irrelevant attributes. These irrelevant attributes values were number 1–10 generated randomly. These values were treated as categorical values. We studied how the performance of proposed algorithm was affected by subspace ratios.

The subspace ratio s is defined as

$$s = \sum_{i=1}^{k} m_i / km,$$

where m_i is the number of relevant dimensions in the ith cluster, m is the total number of the attributes.



```
Algorithm Modified_k-means_Subspace clustering
Begin
Step – A
1-   Discretize numeric data
•  For every categorical attribute
•  Compute distance $\delta(r,s)$ between two categorical values $r$ and $s$
•  For every numeric attribute
•  Compute significance of attribute
•  Assign Data objects to different clusters randomly.
Initialization - Allocate data objects to a pre-determined $k$ number of clusters randomly
and set all initial weight to $1/m$, define value of $\gamma$.
Repeat step 1- 3
    1-  Compute cluster centers for $C_1, C_2, \ldots, C_k$
    2-  Each data object $d_i$ (i=1,2…,$n$) { $n$ is number of data objects in data set}is
        assigned to its closest cluster center using $\vartheta(d_i, C_j)$ (defined in Eq. 3.6)
    3-  Update the contribution of the attribute to the cluster,
            $\lambda_{lt} = \exp(-D_{lt}/\gamma) / \sum_{i=1}^{m} \exp(-D_{li}/\gamma)$
        where $D_{lt} = \sum_{i=1}^{n} \delta_{li}(w_t(d_{it}^{r} - C_{jt}^{r}))^2$ for numeric attribute.
              $D_{lt} = \sum_{i=1}^{n} \delta_{li}\Omega(d_{it}^{c}, C_{jt}^{c})^2$ for categorical attribute.
Until no elements change clusters or a pre-defined number of iterations are reached.
End.
```

Fig. 2. Our proposed algorithm.

As we know clusters are defined by at most 9 attributes (original attributes) the m_1 and m_2 are at most 9, so $s <= 18/2(9 + \text{new added attributes})$. We carried out the experiments on different subspace ratios when $k = 2$. Average clustering results are presented in Table 1. Table 1 shows the effect of the subspace ratios on clustering errors. There is a slight decrease in the average clustering accuracy as we decrease the subspace ratios. However, our proposed algorithm is capable of finding subspace clusters.

Table 1
Clustering results for different subspace ratios for the synthetic dataset.

Subspace ratio (Atmost)	Clustering error in%
1	5.4
0.5	5.8
0.3	6.1
0.1	6.3

4.2. Vote dataset

This is a pure categorical data set with 435 data points described by 16 attributes. These data points belong to two classes; Republican (168 data points) and Democrats (267 data points). With $k = 20$, we got 14 clusters. The clustering results are presented in Table. 2. *We got* 21 *data points not in desired clusters. In other words, the clustering error is* 4.8%. Zaki et al. (2007) *got the*

Table 2
The confusion matrix for the Vote dataset.

Cluster No.	1	2	3	4	5	6	7	8
Republican	5	0	62	0	6	0	73	0
Democrat	1	7	4	147	1	27	2	2
Cluster No.	9	10	11	12	13	14		
Republican	2	8	0	3	7	0		
Democrat	1	2	10	13	45	5		

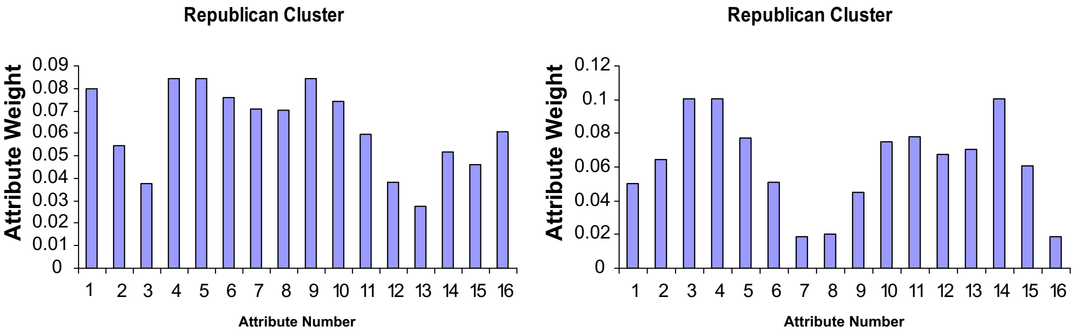


Fig. 3. Attribute weights for cluster no.3 (the left graph) and cluster no.7 (the right graph).

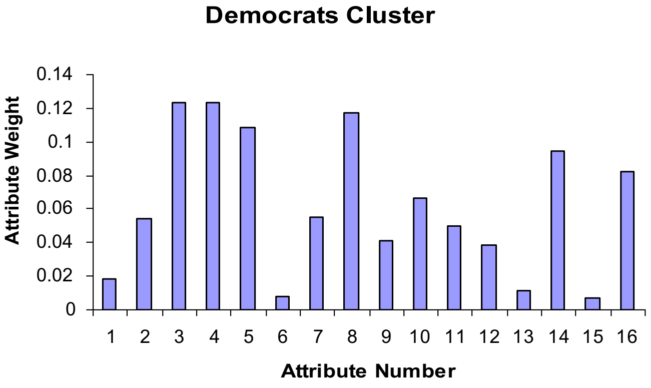


Fig. 4. Attribute weights for the cluster 4 (the Democrat cluster).

clustering error 3.8% *whereas their algorithm was not able to cluster* 0.9% *data points.*

Weight distributions of different attributes in three big clusters are shown in Figs. 3 and 4. The results (Fig. 3) suggest that attribute 1, attribute 4, attribute 5 and attribute 9 are the main contributors to cluster 3 (Republican cluster). The original Republican class has many data points which are represented by attribute 1 (y), attribute 4 (n), attribute 5 (n) and attribute 9 (y) values. In the same way, the Republican class has many data points (cluster No. 7) which are represented by attribute 3 (n), attribute 4 (y), and attribute 14 (y) values whereas Fig. 4. shows that attribute 3, attribute 4 and attribute 8 are the main contributors for cluster 4 (Democrats). The original Democrats class has many data points, which are represented by attribute 3 (y), attribute 4 (n), and attribute 8 (y) values. These results show that the proposed algorithm give good information cluster formation that is consistent with the given information (the original class).

4.3. Mushroom dataset

This is a pure categorical dataset, it contains 8124 data points. These data points are described by 22 attributes and divided into two groups; edible (4208 points) and poisonous (3916 points). As it was a large dataset, we took the number of desired clusters = 40. However, with $k = 40$ we got only 11 clusters. We got 40 clusters with $k = 200$. The results are presented in Table 3. Results suggest that only one of the clusters is not pure. *We got* 335 *data points* (4.1%) *not in the desired clusters. For this dataset* Zaki et al. (2007) *got* 0.3% *data points not in desired clusters.* We also studied the attributes weights for the cluster 6 and the cluster 22 (edible); the clus-

Table 4
The confusion matrix for the DNA dataset.

Cluster No.	1	2	3	4	5	6	7	8
None	191	132	33	91	87	104	63	70
IE	0	0	3	12	63	1	8	1
EI	0	0	122	1	10	1	6	280

Cluster No.	9	10	11	12	13	14	16	17	18	19	20
None	20	21	161	26	40	80	33	160	53	32	111
IE	107	232	0	65	0	0	5	0	175	96	0
EI	70	4	6	20	86	2	140	0	10	8	0

Table 5
The confusion matrix for the Credit dataset.

Cluster	1	2	3	4	5	6	7	8	9	10
Negative	3	16	26	45	2	62	32	26	1	6
Positive	2	0	10	2	6	5	1	0	0	9

Cluster	11	12	13	14	15	16	17	18	19
Negative	6	79	55	0	15	3	0	4	2
Positive	1	5	239	4	2	3	2	0	15

Table 3
The confusion matrix for the Mushroom dataset.

Cluster No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Edible	4	48	12	800	32	768	60	24	144	48	16	36	96	4
Poisonous	0	0	0	335	0	0	0	0	0	0	0	0	0	0

Cluster No.	15	16	17	18	19	20	21	22	23	24	25	26	27
Edible	12	70	8	48	36	192	32	1726	0	0	0	0	0
Poisonous	0	0	0	0	0	0	0	0	36	3	128	4	8
Cluster No.	28	29	30	31	32	33	34	35	36	37	38	39	40
Edible	0	0	0	0	0	0	0	0	0	0	0	0	0
Poisonous	432	383	39	91	63	143	71	135	15	16	31	191	1800

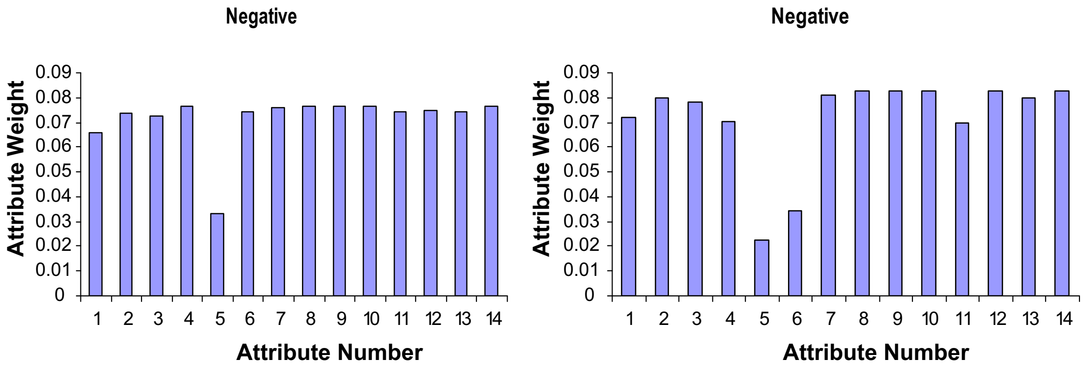


Fig. 5. Attribute weights for the cluster 6 (the left graph) and the cluster 12 (the right graph).

ter 28 and the cluster 40 (poisonous). The cluster 6 and the cluster 22 belong to the edible group but contributions of attributes no. 12, 13, 14, 15 are quite different in these clusters. Similarly for the cluster 28 and the cluster 40, the study suggests that attributes {1, 2, 5, 9, 12–15, 22} contribute differently. This suggests that with our proposed algorithm, we can get interesting information about the cluster formation, which may be useful for decision making.

4.4. DNA dataset

This data consists of 3175 data points each represented by 60 categorical attributes. This data has three classes EI (765), IE (762) and None (1648). Clustering results with $k = 20$ are presented in Table 4. There are 540 data points (17.0%) which are not in the desired clusters. Cluster 9, cluster 10, cluster12, cluster 18 and cluster 19 represent the IE class. Cluster 3, cluster 8, cluster 13 and cluster 16 represent the EI class whereas the rest of the clusters belong to the None class. We also studied how different attributes contribute to different types of clusters. The study suggests that for EI and IE clusters, attributes 20 to 40 contribute more, whereas for None cluster contribution of all attributes are similar. This is in line with the characteristic of the data. (http://www.niaad.liacc.up.pt/old/statlog/datasets/dna/dna.descri.html)

4.5. Australian credit data

This dataset contains data from credit card organization, where customers are divided into two classes. It is a mixed dataset with eight categorical and six numeric features. It contains 690 data points belonging to two classes – Negative (383) and Positive (307). Table 5 presents the results of cluster recovery using our algorithm. We got 13.9% data points not in desired clusters. Huang et al. (2005) achieve 15% clustering error (only 2 out of 2000 runs) for this dataset with full data space clustering, whereas Ahmad and Dey (2007b) achieved 12% clustering error. We also studied the contributions of different attributes to the different clusters. We studied the attribute weights for the cluster 6 (Negative), the cluster 12 (Negative) and the cluster 13 (Positive). We observe that attribute contributions for the cluster 6 and the cluster 12 (Negative clusters) are similar whereas attribute weights for cluster 13 (Positive cluster), are different. The attributes weights for the cluster 6 and the cluster 12 are presented in Fig. 5. The attributes weights for the cluster 13 are presented in Fig. 6. For the cluster 13 (Positive cluster), attribute 8 and attribute 12 are very important. The attribute 8 has two values 0 and 1. The majority of data points with *the attribute* 8 *value* = 0 belong to negative class, whereas the majority of data points with *the attribute* 8 *value* = 1 belong to the negative class. This explains the importance of this attribute in the cluster formation. The low attribute weight value of the attribute 5 is due to almost similar distribution of attribute values in both the clusters (positive and negative).

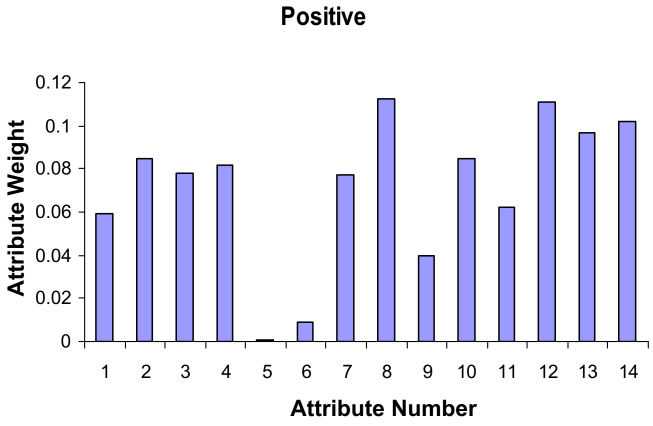


Fig. 6. Attribute weights for the cluster 13.

5. Conclusion

In this paper, we presented a k-means type clustering algorithm for subspace clustering for mixed numeric and categorical data. This algorithm is linear with respect to the number of the data points. We used the method proposed by Jing et al. (2007) to compute the attribute weights in the clusters formation. We presented results on pure categorical and mixed datasets. The results suggest that we are getting good cluster recovery. We explained these clusters with the attribute weights. That suggests that we can use this algorithm for better understanding of cluster formation. In future we will work to understand the relationship between γ (user defined value) and k (number of desired clusters) in a dataset, as we can achieve better results by optimizing these values.

References

Aggarwal, C.C., Yu, P.S., 2000. Finding generalized projected clusters in high dimensional spaces, In: Proc. ACM SIGMOD.

Agrawal, R., Gehrke, Gunopulos, J. D., Raghavan, P. 1998. Automatic subspace clustering of high dimensional data for data mining applications. In: Proc. 1998 ACM SIGMOD, 94–105.

Ahmad, A., Dey, L., 2007a. A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set. Pattern Recognit. Lett. 28 (1), 110–118.

Ahmad, A., Dey, L., 2007b. A k-mean clustering algorithm for mixed numeric and categorical data. Data Knowl. Eng. 63 (2), 503–527.

Barbara, D., Li, Y., Couto, J. 2002. COOLCAT: An entropy-based algorithm for categorical clustering, In: CIKM Conf., 582–589.

Chen, H.L., Chuang, K.T., Chen, M.S., 2008. On data labeling for clustering categorical data. IEEE Trans. Knowl. Data Eng. 20 (11), 1458–1472.

Chu, Y.H., Huang, J.W., Chuang, K.T., Yang, D.N., 2010. Density conscious Subspace clustering for high-dimensional data. IEEE Trans. Knowl. Data Eng. 22 (1), 16–30.

Chu, Y.H., Chen, Y.J., Yang, D.H., Chen, M.S., 2009. Reducing redundancy in subspace clustering. IEEE Trans. Knowl. Data Eng. 21 (10), 1432–1446.

Deng, Z., Choi, K.S., Chung, F.L., Wang, S., 2010. Enhanced soft subspace clustering integrating within-cluster and between-cluster information. Pattern Recognit. 43, 767–778.

Friedman, J.H., Meulman, J.J., 2004. Clustering objects on subsets of attributes. J. Roy. Statist. Soc. Ser. B 66, 1–25.

Ganti, V., Gekhre, J.E., Ramakrishnan, R. 1999. CACTUS-Clustering categorical data using summaries, In: Proc. 2nd Internat. Conf.on Knowledge Discovery and Data Mining (KDD), 311–314.

Guha, S., Rastogi, R., Kyuseok, S. 1999. ROCK: A Robust Clustering Algorithm for Categorical Attributes. In: Proc. 15th Internat. Conf. on Data Engineering, Sydney, Australia, 512–521.

Goil, S., Nagesh, H., Choudhary, A., 1999. Mafia: Efficient and scalable subspace clustering for very large data sets. Technical Report CPDC-TR-9906-010, Northwestern University, 2145 Sheridan Road, Evanston IL 60208.

Goodall, D.W., 1966. A new similarity index based on probability. Biometric 22, 882–907.

Huang, Z. 1997. Clustering Large Data sets with Mixed Numeric and Categorical Values. In: Proceedings of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining, Singapore, World Scientific, 21–34.

Huang, Z. 1997. A fast clustering algorithm to cluster very large categorical data sets in data mining. In: Proc. SIGMOD DMKD Workshop, 1–8.

Huang, J.Z., Ng, M.K., Rong, H., Li, Z., 2005. Automated variable weighting in *k*-mean type clustering. IEEE Trans. Pattern Anal. Machine Intell. 27 (5), 657–668.

Jain, A.K., Dubes, R.C. 1988. Algorithms for Clustering Data. Prentice Hall, Englewood Cliff, New Jersey.

Jing, L., Ng, M.K., Huang, J.Z., 2007. An entropy weighting *k*-means algorithm for subspace clustering of high-dimensional sparse data. IEEE Trans. Knowl. Data Eng. 19 (8), 1026–1041.

Li, C., Biswas, G., 2002. Unsupervised learning with mixed numeric and nominal data. IEEE Trans. Knowl. Data Eng. 14 (4), 673–690.

MacQuuen, J. B. 1967. Some methods for classification and analysis of multivariate observation. In: Proc. 5th Berkley Symposium on Mathematical Statistics and Probability, 281–297.

Moise, G., Zimek, A., Kröger, P., Kriegel, H.P., Sander, J., 2009. Subspace and projected clustering: Experimental evaluation and analysis. Knowledge And Information Systems 21 (3), 299–326.

Sequeira, K., Zaki, M. 2004. SCHISM: A new approach for interesting subspace mining, IEEE Internat. Conf. on Data Mining, 186–193.

Zaki, M.J., Peters, M., Assent, T., Seidl, T., 2007. CLICKS: An effective algorithm for mining subspace clusters in categorical datasets. Data Knowl. Eng. 60 (1), 51–70.