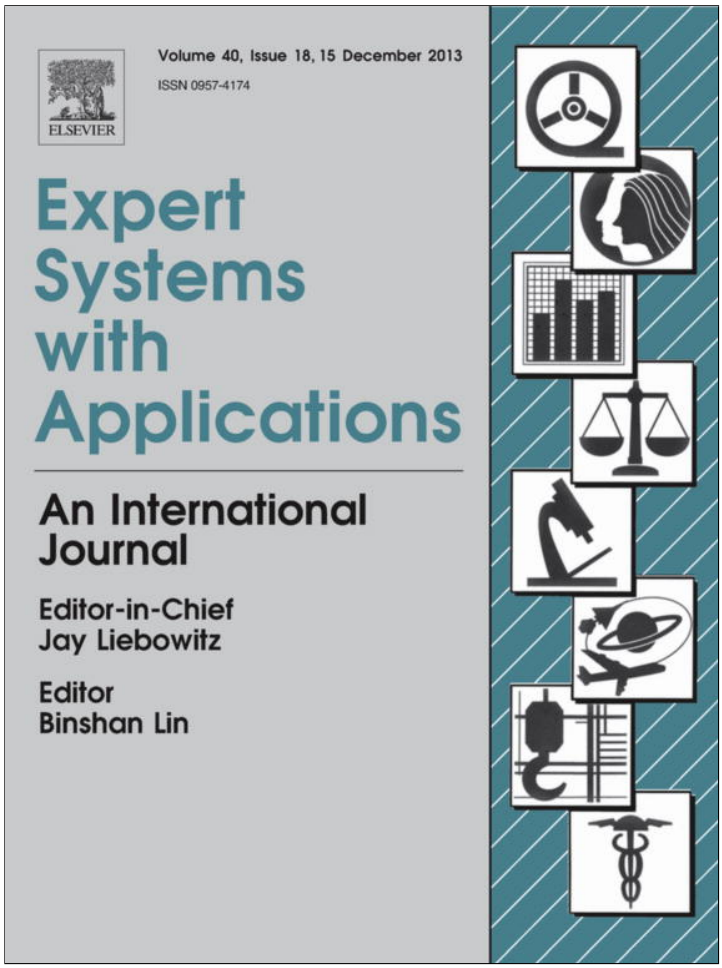

Expert Systems with Applications 40 (2013) 7444–7456

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Cluster center initialization algorithm for K-modes clustering

Shehroz S. Khan [a,*], Amir Ahmad [b]

[a] David R. Cheriton School of Computer Science, University of Waterloo, Canada
[b] Faculty of Computing and Information Technology, King Abdulaziz University, Rabigh, Saudi Arabia

ARTICLE INFO

Keywords:
K-modes clustering
Cluster center initialization
Prominent attributes
Significant attributes

ABSTRACT

Partitional clustering of categorical data is normally performed by using K-modes clustering algorithm, which works well for large datasets. Even though the design and implementation of K-modes algorithm is simple and efficient, it has the pitfall of randomly choosing the initial cluster centers for invoking every new execution that may lead to non-repeatable clustering results. This paper addresses the randomized center initialization problem of K-modes algorithm by proposing a cluster center initialization algorithm. The proposed algorithm performs multiple clustering of the data based on attribute values in different attributes and yields deterministic modes that are to be used as initial cluster centers. In the paper, we propose a new method for selecting the most relevant attributes, namely *Prominent* attributes, compare it with another existing method to find *Significant* attributes for unsupervised learning, and perform multiple clustering of data to find initial cluster centers. The proposed algorithm ensures fixed initial cluster centers and thus repeatable clustering results. The worst-case time complexity of the proposed algorithm is log-linear to the number of data objects. We evaluate the proposed algorithm on several categorical datasets and compared it against random initialization and two other initialization methods, and show that the proposed method performs better in terms of accuracy and time complexity. The initial cluster centers computed by the proposed approach are close to the actual cluster centers of the different data we tested, which leads to faster convergence of K-modes clustering algorithm in conjunction to better clustering results.

1. Introduction

Cluster analysis is a form of unsupervised learning that is aimed at finding underlying structures in the unlabeled data. The objective of a clustering algorithm is to partition a multi-attribute dataset into homogeneous groups (or clusters) such that the data objects in one cluster are more similar to each other (based on some similarity measure) than those in other clusters. Clustering is an active research topic in pattern recognition, data mining, statistics and machine learning with diverse application such as in image analysis (Matas & Kittler, 1995), medical applications (Petrakis & Faloutsos, 1997) and web documentation (Boley et al., 1999).

The partitional clustering algorithms such as K-means (Anderberg, 1973) are very efficient for processing large numeric datasets. Data mining applications require handling and exploration of data that contains numeric, categorical or both types attributes. The K-means clustering algorithm fails to handle datasets with categorical attributes because it minimizes the cost function by calculating *means* and distances. The traditional way to treat categorical attributes as numeric does not always produce meaningful results because generally categorical domains are not ordered. Several approaches have been reported for clustering categorical datasets that are based on K-means paradigm. Ralambondrainy (1995) presents an approach by using K-means algorithm to cluster categorical data by converting multiple category attributes into binary attributes (using 0 and 1 to represent either a category absent or present) and treats the binary attributes as numeric in the K-means algorithm. Gowda and Diday (1991) use a similarity coefficient and other dissimilarity measures to process data with categorical attributes. CLARA (Clustering LARge Application) (Kaufman & Rousseeuw, 1990) is a combination of a sampling procedure and the clustering program Partitioning Around Medoids (PAM). Guha, Rastogi, and Shim (1999) present a robust hierarchical clustering algorithm, ROCK, that uses links to measure the similarity/proximity between a pair of data objects with categorical attributes that are used to merge clusters. However, this algorithm has worst-case quadratic time complexity.

Huang (1997) presents the K-modes clustering algorithm by introducing a new dissimilarity measure to cluster categorical data. The algorithm replaces *means* of clusters with *modes* (most

* Corresponding author.
E-mail addresses: s255khan@uwaterloo.ca (S.S. Khan), amirahmad01@gmail.com (A. Ahmad).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.07.002

frequent attribute value in a attribute), and uses a frequency based method to update *modes* in the clustering process to minimize the cost function. The algorithm is shown to achieve convergence with linear time complexity with respect to the number of data objects. Huang (1998) also points out that in general, the K-modes algorithm is faster than the K-means algorithm because it needs less iterations to converge. In principle, K-modes clustering algorithm functions similar to K-means clustering algorithm except for the cost function it minimizes, and hence suffers from the same drawbacks. Similar to K-means clustering algorithm, the K-modes clustering algorithm assumes that the number of clusters, *K*, is known in advance. Fixed number of *K* clusters can make it difficult to predict the actual number of clusters in the data that may mislead the interpretations of the results. The K-means/K-modes clustering algorithm falls into problems when clusters are of differing sizes, density and non-globular shapes. The K-means clustering algorithm does not guarantee unique clustering due to random choice of initial cluster centers that may yield different groupings for different runs (Jain & Dubes, 1988). Similarly, the K-modes algorithm is also very sensitive to the choice of initial cluster centers and an improper choice may result in highly undesirable cluster structures. Random initialization is widely used as a seed for K-modes algorithm for its simplicity, however, this may lead to non-repeatable clustering results. Machine learning practitioners find it difficult to rely on the results thus obtained and several re-runs of K-modes algorithm may be required to arrive at a meaningful conclusion.

In this paper, we extend the work of Khan and Ahmad (2012) and present a multiple clustering approach that infers cluster structure information from multiple attributes by using the attribute values present in the data for computing initial cluster centers. This approach focus only on *Prominent* attributes (discussed in Section 4.2) that are important for finding cluster structures. We also use another unsupervised learning method to find *Significant* attributes (Ahmad & Dey, 2007a, 2007b) and compare it with the proposed approach. The proposed algorithm performs multiple clustering based on distinct attribute values present in different attributes to generate multiple clustering views of the data that are utilized to obtain fixed initial cluster centers (modes) for K-modes clustering algorithm. The proposed algorithm has worst-case log-linear time complexity with respect to the number of data objects. The present paper extends the previous work in terms of:

- Using a unsupervised method to compute significant attributes and compare their clustering performance and quality of initial cluster centers against the centers computed by prominent attributes.
- Comparing the quality of initial cluster centers by using all attributes and prominent attributes.
- Analyzing the closeness of initial cluster centers to the actual centers by using prominent, significant and all attributes.
- Performing comprehensive experiments, presentation of results, time scalability analysis, inclusion of more datasets, and extended discussions on the multiple clustering challenges from the perspective of the proposed approach.

The rest of the paper is organized as follows. In Section 2, we present a short survey of the research work on cluster center initialization for K-modes algorithm. Section 3 briefly discusses the K-modes clustering algorithm. In Section 4, we present the proposed multiple attribute clustering approach to compute initial cluster centers along with three different approaches to choose different number of attributes to generate multiple clustering views. Section 5 shows the detailed experimental analysis of the proposed method on various categorical datasets and compare it with other cluster center initialization methods. Section 6 concludes the paper with pointers to future work.

2. Related work

The K-modes algorithm (Huang, 1997) extends the K-means paradigm to cluster categorical data and requires random selection of initial cluster centers or modes. As discussed earlier, a random choice of initial cluster centers leads to non-repeatable clustering results that may be difficult to comprehend. The random initialization of cluster centers may only work well when one or more randomly selected initial cluster centers are similar to the actual cluster centers present in the data. In the most trivial case, the K-modes algorithm keeps no control over the choice of initial cluster centers and therefore repeatability of clustering results is difficult to achieve. Moreover, an inappropriate choice of initial cluster centers can lead to undesirable clustering results. The results of partitional clustering algorithms are better when the initial partitions are close to the final solution (Jain & Dubes, 1988). Hence, it is important to invoke K-modes clustering with fixed initial cluster centers that are similar to the true representative centers of the actual clusters to get better results.

There are several research papers reported for computing initial cluster centers for K-modes algorithm, however, most of these methods suffer from either of the following two drawbacks:

(a) The initial cluster center computation methods are quadratic in time complexity with respect to the number of data objects – these type of methods mitigate the advantage of linear time complexity of K-modes algorithm and are not scalable for large datasets.
(b) The initial cluster centers are not fixed and have randomness in the computation steps – these type of methods fare as good as random initialization methods.

We present a short review of the research work done to compute initial cluster centers for K-modes clustering algorithm and discuss their associated problems.

Khan and Ahmad (2003) use density-based multiscale data condensation (Mitra, Murthy, & Pal, 2002) approach with Hamming distance to extract *K* initial cluster centers from the datasets, however, their method has quadratic complexity with respect to the number of data objects. Huang (1998) proposes two approaches for initializing the cluster centers for K-modes algorithm. In the first method, the first *K* distinct data objects are chosen as initial K-modes, whereas the second method calculates the frequencies of all categories for all attributes and assign the most frequent categories equally to the initial K-modes. The first method may only work if the top *K* data objects come from disjoint *K* clusters. The second method is aimed at choosing diverse cluster center that may improve clustering results, however a uniform criteria for selecting K-initial cluster centers is not provided.

Sun, Zhu, and Chen (2002) present an experimental study on applying Bradley and Fayyad's iterative initial-point refinement algorithm (Bradley & Fayyad, 1998) to the K-modes clustering to improve the accuracy and repetitiveness of the clustering results. Their experiments show that the K-modes clustering algorithm using refined initial cluster centers leads to higher precision results that are more reliable than the random selection method without refinement. This method is dependent on the number of cases with refinements and the accuracy value varies. Khan and Kant (2007) propose a method that is based on the idea of evidence accumulation for combining the results of multiple clusterings (Fred & Jain, 2002) and only focus on those data objects that are less vulnerable to the choice of random selection of modes and to choose the most

diverse set of modes among them. Their experiments suggest that the computed initial cluster centers outperform the random choice, however the method does not guarantee fixed choice of initial cluster centers. He (2006) presents two farthest point heuristic for computing initial cluster centers for K-modes algorithm. The first heuristic is equivalent to random selection of initial cluster centers and the second uses a deterministic method based on a scoring function that sums the frequency count of attribute values of all data objects. This heuristic does not explain how to choose a point when several data objects have same scores, and if it randomly breaks ties, then fixed centers cannot be guaranteed. The method only considers the distance between the data points, due to which outliers can be selected as cluster centers.

Wu, Jiang, and Huang (2007) develop a density based method to compute the K initial cluster centers which has quadratic complexity. To reduce the worst case complexity to $O(n^{1.5})$, they randomly select square root of the total points as a sub-sample of the data, however, this step introduces randomness in the final results and repeatability of clustering results may not be achieved. Cao, Liang, and Bai (2009) present an initialization method that consider distance between objects and the density of the objects. Their method selects the object with the maximum average density as the first initial cluster center. For computing other cluster centers, the distance between the object and the already known clusters, and the average density of the object are considered. A shortcoming of this method is that a boundary point may be selected as the first center that can affect the quality of selection of subsequent initial cluster centers. Bai, Liang, Dang, and Cao (2012) propose a method to compute initial cluster centers on the basis of a density function (defined by using the average distance of all the other points from a point) and a distance function. The first cluster center is decided by the density function. The remaining cluster centers are computed by using the density function and the distance between the already calculated cluster centers and the probable new cluster center. In order to calculate the density of a point they calculate the summary of all the other points. Hence, there is information loss that may lead to improper density calculation, which can affect the results. A major problem with this research paper lies in the evaluation of results. For at least two datasets, the accuracy, precision and recall values are computed incorrectly. From the confusion matrix presented in the paper the accuracy, precision and recall values for

- Dermatology data should be 0.6584, 0.6969, 0.6841 and not 0.7760, 0.8527, 0.7482
- Zoo data should be 0.7425, 0.7703, 0.8654 and not 0.9208, 0.8985 and 0.8143. The confusion matrix mis-classify almost half of the data objects of first cluster and thus accuracy cannot reach the value indicated in the paper.

In comparison to the above stated research works, the proposed algorithm (see Section 4 for details) for finding initial clusters centers for categorical datasets circumvents both the drawbacks discussed earlier i.e. its worst-case time complexity is log-linear in the number of data objects and it provides deterministic (fixed) initial cluster centers.

3. K-modes algorithm for clustering categorical data

Due to the limitation of the dissimilarity measure used by traditional K-means algorithm, it cannot be used to cluster categorical dataset. The K-modes clustering algorithm is based on K-means paradigm, but removes the numeric data limitation whilst preserving its efficiency. The K-modes algorithm (Huang, 1998) extends the K-means paradigm to cluster categorical data by removing the barrier imposed by K-means through following modifications:

1. Using a simple matching dissimilarity measure or the Hamming distance for categorical data objects.
2. Replacing means of clusters by their modes (cluster centers).

The simple matching dissimilarity measure (Hamming distance) can be defined as following. Let X and Y be two categorical data objects described by m categorical attributes. The dissimilarity measure $d(X,Y)$ between X and Y can be defined by the total mismatches of the corresponding attribute categories of two objects. Smaller the number of mismatches, more similar the two objects are. Mathematically, we can say

$$d(X,Y)=\sum_{j=1}^{m}\delta(x_j,y_j) \tag{1}$$

where $\delta(x_j,y_j)=\begin{cases}0 & (x_j=y_j)\\ 1 & (x_j\neq y_j)\end{cases}$, and $d(X,Y)$ gives equal importance to each category of an attribute.

Let N be a set of n categorical data objects described by m categorical attributes, $M_1,M_2,\ldots,M_m$. When the distance function defined in Eq. (1) is used as the dissimilarity measure for categorical data objects, the cost function becomes

$$C(Q)=\sum_{i=1}^{n}d(N_i,Q_i) \tag{2}$$

where N_i is the ith element and Q_i is the nearest cluster center of N_i. The K-modes algorithm minimizes the cost function defined in Eq. (2).

The K-modes algorithm assumes that the knowledge of number of natural grouping of data (i.e. K) is available and consists of the following steps (taken from Huang (1997)):

1. Select K initial cluster centers, one for each of the cluster.
2. Allocate data objects to the cluster whose cluster center is nearest to it according to Eq. (2). Update the K clusters based on allocation of data objects and compute K new modes of all clusters.
3. Retest the dissimilarity of objects against the current modes. If an object is found such that its nearest mode belongs to another cluster rather than its current one, reallocate the object to that cluster and update the modes of both clusters.
4. Repeat step 3 until no data object has changed cluster membership.

4. Proposed approach for computing initial cluster centers using multiple attribute clustering

Khan and Ahmad (2004) show that for partitional clustering algorithms, such as K-Means,

- Some of the data objects are very similar to each other, that is why they share same cluster membership irrespective of the choice of initial cluster centers, and
- An individual attribute may also provide some information about initial cluster centers

He, Xu, and Deng (2005) present a unified view on categorical data clustering and cluster ensemble for the creation of new clustering algorithms for categorical data. Their intuition is that the attributes present in a categorical data contribute to the final cluster structure. They consider the distinct attribute values of an

attribute as cluster labels giving "best clustering" without considering other attributes and create a cluster ensemble.

Müller, Günnemann, Färber, and Seidl (2010) defines *multiple clusterings* as setting up multiple set of clusters for every data object in a dataset with respect to multiple views on the data. The basic objective of multiple clustering is to represent different perspectives on the data and utilize the variation among the clustering results to gain additional knowledge about the structure in the data. They discuss several challenges that arise due to multiple clustering of data and merging of their results. One of the major challenges is related to the detection of different clusterings revealed by multiple views on the data. This problem of multiple views has been studied in the original data space (Caruana, Elhawary, Nguyen, & Smith, 2006), orthogonal space (Davidson & Qi, 2008) and subspace projections (Agrawal, Gehrke, Gunopulos, & Raghavan, 1998). Other challenges include given knowledge about known clusterings, processing schemes for clustering, the number of multiple clusterings and flexibility.

We take motivation from these research works and propose a new cluster initialization algorithm for categorical datasets that perform multiple clustering on different attributes (in the original data space) and uses distinct attribute values in an attribute as cluster labels. These multiple views provide new insights into the hidden structures of the data that serve as a cue to find consistent cluster structure and aid in computing better initial cluster centers. In the following subsections, we will present three approaches to select different attribute spaces that can help in generating different clustering views from the data. It is to be noted that all the proposed approaches assume that the desired number of clusters, K, is known in advance.

4.1. Vanilla approach

A *Vanilla* approach is to consider all the attributes (m) present in the data and generate M clustering views that can be used for further analysis (see details for the follow up steps in Section 4.4).

4.2. Prominent attributes

Khan and Ahmad (2012) present that only few attributes may be useful to generate multiple clustering views that can help in computing initial cluster centers for K-modes algorithm. These relevant attributes are extracted based on the following experimental observations:

1. There may be some attributes in the dataset whose number of attribute values are less than or equal to K. Due to fewer attribute values per cluster, these attributes possess higher discriminatory power and will play a significant role in deciding the initial cluster centers as well as the cluster structures. The set of these relevant attributes are called *Prominent* attributes (P).
2. For the other attributes in the dataset whose number of attribute values are greater than K, the numerous attribute values in these attributes will be spread out per cluster. These attributes add little to determine proper cluster structure and contribute less in deciding the initial representative modes of the clusters.

Algorithm 1 shows the steps to compute *Prominent* attributes from a dataset. The number of attributes in the set P is defined as $p = |P|$. In the algorithm, $p = 0$ refers to a situation when there are no prominent attributes in the data and $p = m$ means that all attributes are prominent attributes. In both of these scenarios, all the attributes in the data are considered prominent or else a reduced set, P, of prominent attributes (equals to p) is chosen.

Algorithm 1. Computation of *Prominent* attributes

```
Input: N = data objects, M = Set of attributes in the data,
   m = |M| = Number of attributes in the data, p = 0
Output: P = Set of Prominent attributes
P = φ
for i = 1 → m do
  if Number of distinct attribute values in M_i > 1 && M_i ⩽ K
  then
    Add M_i to P
    increment p
  end if
end for
if p = 0 || p = m then
  use all attribute and call computeInitialModes(Attributes M)
else
  use reduced prominent attributes and call
  computeInitialModes(Attributes P)
end if
```

4.3. Significant attributes

As discussed in the previous section, we select prominent attributes as we expect that these attributes play important role in clustering. The following section is taken from the work of Ahmad and Dey (2007a) that discusses an approach to rank important attributes in a dataset. We use their method to find significant attributes from the dataset.

Ahmad and Dey (2007a, 2007b) propose an unsupervised learning method to compute the significance of attributes. On the basis of their significance, important attributes can be selected. In this method the most important step is to find out the distance between any two categorical values of an attribute. The distance between two distinct attribute values is computed as a function of their overall distribution and co-occurrence with other attributes. The distance between the pair of values x and y of attribute M_i with respect to the attribute M_j, for a particular subset w of attribute M_j values, is defined as follows:

$$\phi_{ij}^{w}(x,y) = p(w|x) + p(\sim w|y) - 1 \tag{3}$$

where $p(w|x)$ denotes the probability that elements of the dataset with attribute M_i equal to x have attribute M_j value such that it is contained in w and, $p(\sim w|y)$ denotes the probability that elements of the dataset with attribute M_i equal to y have attribute M_j value such that it is not contained in w.

The distance between attribute values x and y for M_i with respect to attribute M_j is denoted by $\phi_j(x,y)$ and is given by

$$\phi_j(x,y) = p(W|x) + p(\sim W|y) - 1 \tag{4}$$

where W is the subset of values of M_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 $\phi_j(x,y)$ between x and y in the dataset. The average value of all the attribute values pair distances is taken as the signif-

icance of the attribute. Algorithm 2 shows the steps to compute the significance of attributes in the data.

Algorithm 2. Computation of significance of attributes

Input: D = Categorical Dataset, N = data objects, M = Set of Attributes in the data, $m = |M|$ = Number of attributes in the data
Output: S = Set of attributes sorted in order of their significance
for every attribute M_i **do**
  for every pair of categorical attribute values (x, y) **do**
    $Sum = 0$
    for every other attributes M_j **do**
      $\phi^j(x,y) = max(p(w|x) + p_i^y(w|y) - 1$
      where w is subset of jth attribute values
      $Sum = Sum + \phi^j(x,y)$
    end for
    Distance $\phi(x,y)$ between categorical values $(x,y) = \frac{Sum}{(m-1)}$
  end for
  The average value of all the pair distances is taken as the significance of the attribute.
end for

We provide an example below to illustrate Algorithm 2. Consider a pure categorical dataset with three attributes M_1, M_2 and M_3 as shown in Table 1. We compute the significance of attribute M_1 by calculating the distance of each pair of attribute value with respect to every other attribute. In this case there is only one pair (L,T), therefore;

The distance between L and T with respect to M_2 is:

$$max(p(W|L)+p(\sim W|T)-1) = p(C|L)+p(\sim C|T)-1 = 1+\frac{2}{3}-1 = \frac{2}{3}$$

where W is the subset of values of M_2

Similarly, the distance between L and T with respect to M_3 is:

$$max(p(W|L)+p(\sim W|T)-1) = p(E|L)+p(\sim E|T)-1 = 1+\frac{1}{2}-1 = \frac{1}{2}$$

The average distance between L and T is:

$$\phi(L,T) = \frac{1}{2}\left(\frac{2}{3}+\frac{1}{2}\right) = 0.58$$

As there is only one pair of values in the attribute M_1, the significance of attribute M_1 (i.e. the average of distances of all pairs) = 0.58.

This method to compute significance of attribute has been used in various K-means type clustering algorithms for mixed numeric and categorical datasets (Ahmad & Dey, 2007a, 2007b, 2011). Generally the cost function of K-means type algorithm give equal importance to all the attributes. Ahmad and Dey (2007a, 2007b, 2011) show that with incorporating these significance of attributes in the cost function, better clustering results can be achieved. Ji, Pang, Zhou, Han, and Wang (2012) show that this approach is also useful for fuzzy clustering of categorical datasets. In this paper, we will use this approach to select the significant attributes from the datasets.

4.4. Computation of initial cluster centers

In the preceding sections, we discussed three methods of choosing attributes that can be used for computation of initial cluster centers. The *Vanilla* approach chooses all the attributes whereas the prominent attribute approach (see Section 4.2) has the ability to choose fewer number of attributes depending upon the distribution of attribute values in different attributes in the data. We will discuss the potential problem of choosing all the attributes in Section 5.3. The method to compute significant attributes (see Section 4.3) provides a ranking of all the attributes in order of their significance in the dataset. However, there is no straight-forward way to choose the most significant attributes in the data except to use a arbitrary cut-off value. For the experimentation, we choose the number of significant attributes to be the same as prominent attributes and discard rest of them, if all the attributes in the dataset are prominent then all of them are considered significant.

The main idea of the proposed algorithm is to partition the data into clusters that corresponds to the number of distinct attribute values for *Vanilla*/*Prominent*/*Significant* attributes, and generate a cluster label for every data object present in the dataset. This cluster labeling is essentially a clustering view of the original data in the original space. Repeating this process for all the *Vanilla*/*Prominent*/*Significant* attributes yield a number of cluster labels that represent multiple partition views of every data object. The cluster labels that are assigned to a data object over these multiple clusterings is termed as *cluster string* and the number of total *cluster strings* is equal to the number of data objects present in the dataset. As noted in Section 4, some data objects will not be affected by choosing different initial cluster centers and their cluster strings will remain same. The distinct number of cluster strings represents the number of distinguishable clusters in the data. The algorithm assumes that the knowledge of the natural clusters in the data i.e. K is available and if the number of distinct cluster strings are more than K then it merges them into K clusters of cluster strings, such that the cluster strings within a cluster are more similar than others. Lastly, the cluster strings within each K clusters are replaced by their corresponding data objects and modes of every K cluster is computed that serves as the initial cluster centers for the K-modes algorithm. In summary, the proposed method finds the dense localized regions in the dataset in the form of distinguishable clusters. If their count is greater than K then it merges them to K clusters (and has the ability to ignore the infrequent clusters) and finds their group modes to be used as initial cluster centers. This process helps in avoiding the outliers contributing to the computation of initial cluster centers.

Table 1
Categorical dataset.

Attributes		
M_1	M_2	M_3
L	C	E
L	C	F
T	C	F
T	K	F
T	D	F

Table 2
Cluster strings of different data objects.

Data point	Cluster string
D_1	1-1-3-2
D_2	2-2-1-1
D_3	1-1-3-2
D_4	2-2-1-1
D_5	1-2-4-2
D_6	2-1-4-1
D_7	2-2-2-1
D_8	1-1-3-2
D_9	2-2-2-1
D_{10}	1-2-3-1

Algorithm 3. *computeInitialModes*(*Attributes A*)

Input- Dataset $N, n = |N|$ = the number of data objects, and A is the set of categorical attributes with $a = |A|$ = number of attributes. If all the attributes are considered, then $A = M$ and $a = m$, If prominent/significant attributes are considered, $a \leqslant m$ i.e. $a = p$. $|a_i|$ is the cardinality of the a_i attribute and K is a user defined number that represent the number of clusters in the data.

Output- K cluster centers

Generation of cluster strings

for i = 1…a **do**

1. Divide the dataset into $|a_i|$ clusters on the basis of these $|a_i|$ attribute values such that data objects with different values (of this attribute a_i) fall into different clusters. Compute cluster centers of these $|a_i|$ clusters.
2. Partition the data by performing K-modes clustering that uses the cluster centers computed in above step as initial cluster centers.
3. Assign cluster label to every data object. S_{ti} defines the cluster label of tth data object computed by using a_i attribute, where $t = 1,2\ldots n$.

end for

The cluster labels assigned to a data object is considered as a cluster string, resulting in the generation of n clustering strings.

4. Find distinct cluster strings from n strings, count their frequency, and sort them in descending order. Their count, K', is the number of distinguishable clusters.
5. **if K′ = K** Get the data objects corresponding to these K cluster strings, and compute cluster centers of these K clusters. These will be the required initial cluster centers.
6. **if K′ > K** Merge similar distinct cluster string of K' strings into K clusters (more details in Section 4.4.1) and compute the cluster centers. These cluster centers will be the required cluster centers.
7. **if K′ < K** Reduce the value of K and repeat the complete process.

The steps to find initial cluster centers by using the proposed approach are presented in Algorithm 3. The computational efficiency of step 4 of the proposed algorithm can be improved by using other approaches such as on-line suffix trees (Ukkonen, 1995) that can perform string comparisons in time linear in the length of the string.

To illustrate Algorithm 3, we present a descriptive example. Suppose we have 10 data objects $D_1, D_2, \ldots, D_{10}$, defined by 4 categorical attributes with $K = 2$. Let the cardinality of M_1, M_2, M_3 and M_4 are 2, 2, 4, 2. For the *Vanilla* approach, we consider all the attributes and first divide the data objects on the basis of attribute M_1 and calculate 2 cluster centers because cardinality of M_1 is 2. We run K-modes algorithm by using these initial cluster centers. Every data object is assigned a cluster label (either 1 or 2) and the same process is repeated to all other attributes. As there are 4 attributes, each data object will have a cluster string that consists of 4 labels. For example data object D_1 has 1-2-2-1 as the cluster string. This means that in the first run (using M_1 to create initial clusters) the data object D_1 is placed in cluster 1, in the second run (using M_2 to create initial clusters) the data object D_1 is placed in cluster 2 and so on. We will get 10 different cluster strings corresponding to every data object. Suppose we get the following clustering strings for different data objects as shown in Table 2. We calculate the frequency of all the distinct strings as shown in Table 3.

Table 3
Example of frequency computation of distinct cluster strings.

String	Frequency	Data objects
1-1-3-2	3	D_1,D_3,D_8
2-2-1-1	2	D_2,D_4
2-2-2-1	2	D_7,D_9
1-2-4-2	1	D_5
2-1-4-1	1	D_6
1-2-3-1	1	D_{10}

We take $10^{0.5} \approx 3$ most frequent cluster strings (details in Section 4.4.1 on this step) and cluster them by using hierarchical clustering with $K = 2$. The similar strings 2-2-1-1 and 2-2-2-1 are merged in one cluster. This leads to two clusters containing the cluster strings 1-1-3-2 and 2-2-1-1, 2-2-2-1 with their corresponding data objects, i.e.

$$Cluster1 = \{D_1, D_3, D_8\}$$
$$Cluster2 = \{D_2, D_4, D_7, D_9\}$$

The data objects belonging to these clusters are to be used to compute the required 2 cluster centers as $K = 2$. The other infrequent cluster strings and their corresponding data objects are assumed to be outliers that do not contribute in computing the initial cluster centers. The centers of these clusters serve as the initial cluster centers for the K-Modes algorithm. For prominent features approach, the attributes with attribute values less than or equal to the number of clusters are selected. In the example shown, attributes M_1, M_2 and M_4 attributes are selected and the same procedure is followed with these three attributes to compute the initial cluster center. In the significant attributes approach, firstly the significant attributes are calculated, then they are used to calculate the initial cluster centers. For example if M_1, M_2 and M_3 are the most significant attributes, then they are used to calculate the initial cluster centers following the above procedure.

Algorithm 3 may give rise to an obscure case where the number of distinct cluster strings are less than the chosen K (assumed to represent the natural clusters in the data). This case can happen when the partitions created based on the attribute values of A attributes group the data in almost the same clusters every time. Another possible scenario is when the attribute values of all attributes follow almost same distribution, which is normally not the case in real data. This case also suggests that probably the chosen K does not resemble with the natural grouping and it should be changed to a different value. The role of attributes with attribute values greater than K has to be investigated in this case. Generally, in K-modes clustering, the number of desired clusters (K) is selected without the knowledge of natural clusters in the data. The number of natural clusters may be less than the number of the desired clusters. If the number of the cluster strings (K') obtained is less than K, a viable solution is to reduce the value of K and then apply the proposed algorithm to calculate the initial cluster centers. However, this particular case is out of the scope of the present paper.

4.4.1. Merging clusters

As discussed in step 6 of Algorithm 3, there may arise a case when $K' > K$, which means that the number of distinguishable clusters obtained by the algorithm are more than the desired number of clusters in the data. Therefore, K' clusters must be merged to arrive at K clusters. As these K' clusters represent distinguishable clusters, a trivial approach could be to sort them in order of cluster string frequency and pick the top K cluster strings. A problem with this method is that it cannot be ensured that the top K most frequent cluster strings are representative of K clusters. If more than

one cluster string comes from same cluster then the K-modes algorithm will give undesirable clustering results. This fact is also verified experimentally and holds to be true.

Keeping this issue in mind, we propose to use the hierarchical clustering method (Hall et al., 2009) to merge K' distinct cluster strings into K clusters. The hierarchical clustering generates more informative cluster structures than the unstructured set of clusters returned by non-hierarchical clustering methods (Jain & Dubes, 1988). Most hierarchical clustering algorithms are deterministic and stable in comparison to their partitional counterparts. However, hierarchical clustering has the disadvantage of having quadratic time complexity with respect to the number of data objects. In general, K' cluster strings will be less than n. However, to avoid extreme case such as when $K' \approx n$, we only choose the most frequent $n^{0.5}$ distinct cluster strings. This will make the hierarchical algorithm log-linear with the number of data objects (K' or $n^{0.5}$ distinct cluster strings here). The Hamming distance (defined in Section 3) is used to compare the cluster strings. The proposed algorithm is based on the observation that some data objects always belong to same clusters irrespective of the initial cluster centers. The proposed algorithm attempts to capture those data objects that are represented by most frequent strings. The infrequent cluster strings can be considered as outliers or boundary cases and their exclusion does not affect the computation of initial cluster centers. In the best case, when $K' \ll n^{0.5}$, the time complexity effect of log-linear hierarchical clustering will be minimal. This process generates K M-dimensional modes that are to be used as initial cluster centers for K-modes clustering algorithm. For merging cluster strings (in Section 5), we use 'single-linkage' hierarchical clustering, however other options such as average-linkage, complete-linkage, etc. can also be used.

Continuing with the example shown in Section 4.4, we start with $n^{0.5}$ strings as the lowest level of the tree for the bottom up approach of hierarchical clustering. Similar strings are merged up to the level where the number of clusters is equal to the number of desired cluster, K. The data objects belonging to strings in a cluster are used to compute initial cluster center. In the example shown in the previous section, there are three strings, 1-1-3-2, 2-2-1-1 and 2-2-2-1, that are to be used for computing initial cluster centers. The number of designated clusters is 2. The similar strings 2-2-1-1 and 2-2-2-1 are merged, resulting in two clusters. All the data objects corresponding to these strings within a cluster is used to compute the cluster centers.

4.4.2. Choice of attributes

The proposed algorithm starts with the assumption that there exists prominent attributes in the data that can help in obtaining distinguishable cluster structures that can either be used as is or be merged to obtain initial cluster centers. In the absence of any prominent attributes (or if all attributes are prominent), the *Vanilla* approach, all the attributes are selected to find initial cluster centers. Since attributes other than prominent attributes contain attribute values more than K, a possible repercussion is the increased number of distinct cluster strings due to the availability of more cluster allotment labels. This implies an overall reduction in the individual count of distinct cluster strings and many small clusters may be generated. In our formulation, the hierarchical clusterer imposes a limit of $n^{0.5}$ on the top cluster strings to be merged, therefore some relevant clusters could lay outside the bound during merging. This may lead to some loss of information while computing the initial cluster centers. The best case occurs when the number of distinct cluster strings are less than or equal to $n^{0.5}$.

4.4.3. Evaluating time complexity

The proposed algorithm to compute initial cluster centers has three parts,

1. Compute *Vanilla*/*Prominent*/*Significant* attributes
2. Compute initial cluster centers
3. If needed, merge clusters

The time complexity of computation of *Vanilla*/*Prominent* attributes is $O(nm)$, whereas for computing significant attributes is $O(nm^2T^2)$ (Ahmad & Dey, 2007a), where T is the average number of distinct attribute values per attribute and $T \ll n$. Computation of initial cluster centers (from Algorithm 3) needs the basic K-modes algorithm to run P times (in the worst-case m times). As the K-modes algorithm is linear with respect to the size of the dataset (Huang, 1997), the worst-case time complexity will be $O(rKm^2n)$, where r is the number of iterations needed for convergence and $r \ll n$. For merging the distinct cluster strings into K clusters, *computeInitialModes* (*Attributes A*) uses hierarchical clustering. The worst-case complexity of the hierarchical clustering is $O(n^2 \log n)$, however the proposed approach chooses only $n^{0.5}$ most frequent distinct cluster string (see Section 4.4.1), therefore the worst-case complexity for merging cluster strings become $O(n \log n)$. Combining all the parts together, we get two worst-case time complexities:

- Using All/*Prominent* attributes – $O(nm + rKm^2n + n \log n)$
- Using *Significant* attributes – $O(nm^2T^2 + rKm^2n + n \log n)$

It is to be noted that the worst-case time complexity using significant attributes is higher than using all/prominent attributes due to the additional computation time spent in finding out the significance of attributes. However, the worst-case time complexity of using both the methods is log-linear in the number of data objects. With prominent attributes approach, the proposed method is advantageous for the datasets when $n \gg rKm^2$. For significant attributes approach, the method may be useful for those datasets when $n \gg m^2T^2 + rKm^2$.

5. Experimental analysis

5.1. Datasets

To evaluate the performance of the proposed initialization method, we use several pure categorical datasets from the UCI Machine Learning Repository (Batche & Lichman, 2013). A short description for each dataset is given below.

Soybean Small. This dataset consists of 47 cases of soybean disease each characterized by 35 multi-valued categorical variables. These cases are drawn from four populations, each one of them representing one of the following soybean diseases: D1-Diaporthe stem canker, D2-Charcoat rot, D3-Rhizoctonia root rot and D4-Phytophthorat rot. Ideally, a clustering algorithm should partition these given cases into four groups (clusters) corresponding to the diseases. The clustering results on Soybean Small data are shown in Table 5.

Breast Cancer Data. This data has 699 instances with 9 attributes. Each data object is labeled as benign (458% or 65.5%) or malignant (241% or 34.5%). There are 9 instances in attribute 6 and 9 that contain a missing (i.e. unavailable) attribute value. The clustering results of breast cancer data are shown in Table 6.

Zoo Data. It has 101 instances described by 16 attributes and distributed into 7 categories. The first attribute contains a unique animal name for each instance and is removed because it is non-informative. All other characteristics attributes are Boolean except for the character attribute corresponds to the number of legs that lies in the set 0, 2, 4, 5, 6, 8. The clustering results of Zoo data are shown in Table 7.

Lung Cancer Data. This dataset contains 32 instances described by 56 attributes distributed over 3 classes with missing values in attributes 5 and 39. The clustering results for lung cancer data are shown in Table 8.

Mushroom Data. Mushroom dataset consists of 8124 data objects described by 22 categorical attributes distributed over 2 classes. The two classes are edible (4208 objects) and poisonous (3916 objects). It has missing values in attribute 11. The clustering results for mushroom data are shown in Table 9.

Congressional Vote Data. This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes. Each of the votes can either be a yes, no or an unknown disposition. The data has 2 classes with 267 democrats and 168 republicans instances. The clustering results for Vote data are shown in Table 10.

Dermatology Data. This dataset contains six types of skin diseases for 366 patients that are evaluated using 34 clinical attributes, 33 of them are categorical and one is numerical. The categorical attribute values signify degrees in terms of whether the feature is present, contain largest possible amount or relative intermediate values. In our experiment, we discretize the numerical attribute (representing the age of the patient) to contain 10 categories. The clustering results for Dermatology data are presented in Table 11.

We used the WEKA framework (Hall et al., 2009) for the data pre-processing and implementing the proposed algorithm.[1]

5.2. Comparison and performance evaluation metric

To evaluate the quality of clustering results and their fair comparison, we used the performance metrics used by Wu et al. (2007) that are derived from information retrieval. Assuming that a dataset contains K classes, for any given clustering method, let e_i be the number of data objects that are correctly assigned to class C_i, let b_i be the number of data objects that are incorrectly assigned to class C_i, and let c_i be the data objects that are incorrectly rejected from class C_i, then precision, recall and accuracy are defined as follows:

$$PR = \frac{\sum_{i=1}^{K}\left(\frac{e_i}{e_i+b_i}\right)}{K} \tag{5}$$

$$RE = \frac{\sum_{i=1}^{K}\left(\frac{e_i}{e_i+c_i}\right)}{K} \tag{6}$$

$$AC = \frac{\sum_{i=1}^{K} e_i}{N} \tag{7}$$

Jain and Dubes (1988) noted that the results of partitional clustering algorithms improve when the initial cluster centers are close to the actual cluster centers. To measure the closeness between initial cluster centers computed by the proposed method and the actual modes of the clusters in the data, we define a match metric,

$$matchMetric = \frac{1}{K*m}\sum_{i=1}^{K}\sum_{j=1}^{n}\delta(initial_{ij}, actual_{ij}) \tag{8}$$

where $initial_{ij}$ is the jth value of the initial mode for the ith cluster, $actual_{ij}$ is the corresponding jth value of the actual mode for the ith cluster and δ is defined same as in Section 3. The *matchMetric* will give degree of closeness between initial and actual modes with a value of 0 means no match and 1 means exact match between them.

[1] The Java source code is publicly available at http://www.cs.uwaterloo.ca/~s255khan/code/kmodes-init.zip.

Table 4
Effect of choosing different number of attributes.

Dataset	Vanilla		*Prominent*		*Significant*		$n^{0.5}$
	m	CS_A	p	CS_P	s	CS_S	
Soybean	35	25	20	21	20	23	7
Mushroom	22	683	5	16	5	44	91
Dermatology	34	357	33	352	33	357	20
Lung-Cancer	56	32	54	32	54	32	6
Zoo	16	7	16	7	16	7	11
Vote	16	126	16	126	16	126	21
Breast-Cancer	9	355	9	355	9	355	27

5.3. Effect of number of attributes

In Section 4.4.2, we discussed that choosing all the attributes can lead to the generation of large number of cluster strings, specially if the attributes have many attribute values. To test this intuition, we performed a comparative analysis of the effect of the number of selected attributes on the number of distinct cluster strings (generated in Step 6 of Algorithm 3). In Table 4, m is the total number of attributes in the data, p is the number of prominent attributes, s is the number of significant attributes ($s = |S|$ and $s = p$), CS_M, CS_P and CS_S are the number of distinct cluster string obtained using *Vanilla*, *Prominent* and *Significant* attributes, and $n^{0.5}$ is the limit on the number of top cluster strings to be merged using hierarchical clustering. The table shows that choosing a *Vanilla* approach (all attributes) leads to larger number of cluster strings, whereas with the proposed approach (either prominent or significant attributes) they are relatively smaller. This fact can be seen for the Soybean Small, Mushroom and Dermatology data. For Lung Cancer data $p \approx m$ therefore the number of cluster strings are equivalent. For Soybean Small and Mushroom datasets, for same number of *Prominent* and *Significant* attributes, the corresponding number of cluster strings are different ($CS_P \neq CS_S$). This is due to the fact that the set of p prominent and significant attributes is different because both methods select attributes by using different approaches. While the choice of prominent attributes (when they are less than m) should reduce the overall cluster strings as it selects attributes with fewer attribute values, the attributes selected by significance method may contain attributes with more attribute values that results in more cluster strings. For Zoo, Vote and Breast Cancer datasets, all the attributes were prominent therefore p and m are same and hence $CS_P = CS_A$. It is to be noted that the number of distinct cluster strings using proposed approach for Zoo and Mushroom datasets are within the bounds of $n^{0.5}$ limit.

5.4. Clustering results

In this section, we present the K-Modes clustering results that use the initial cluster centers computed with the proposed method. We conducted five set of experiments, their details are as follows:

- *Experiment1*: We compared the clustering results obtained by using prominent attributes to find initial cluster centers with the method of random selection of initial cluster centers and the methods described by Cao et al. (2009) and Wu et al. (2007). As mentioned in Section 2, there are some computational inaccuracies in the work of Bai et al. (2012), therefore we exclude their method from comparison with our work. For random initialization, we randomly group data objects into K clusters and compute their modes to be used as initial cluster centers.
- *Experiment2*: We compared the clustering results obtained by using prominent and significant attributes to find initial cluster centers.

Table 5
Clustering results for Soybean Small data.

Cluster	Class			
	D1	D2	D3	D4
(a) Confusion matrix				
D1	10	0	0	0
D2	0	10	0	0
D3	0	0	10	2
D4	0	0	0	15
	Random	Wu	Cao	Proposed
(b) Performance comparison				
AC	0.8644	**1**	**1**	0.9574
PR	0.8999	**1**	**1**	0.9583
RE	0.8342	**1**	**1**	0.9705

Table 6
Clustering results for Breast Cancer data.

Cluster	Class			
	Benign			Malignant
(a) Confusion matrix				
Benign	453			56
Malignant	5			185
	Random	Wu	Cao	Proposed
(b) Performance comparison				
AC	0.8364	0.9113	0.9113	**0.9127**
PR	0.8699	0.9292	0.9292	**0.9318**
RE	0.7743	0.8773	0.8773	**0.8783**

Table 7
Clustering results for Zoo data.

Cluster	Class						
	a	b	c	d	e	f	g
(a) Confusion matrix							
a	39	0	0	0	0	0	0
b	0	19	0	0	0	0	0
c	0	1	4	0	4	0	0
d	2	0	1	13	0	0	0
e	0	0	0	0	0	0	1
f	0	0	0	0	0	8	2
g	0	0	0	0	0	0	7
	Random		Wu		Cao		Proposed
(b) Performance comparison							
AC	0.8356		0.8812		0.8812		**0.8911**
PR	0.8072		0.8702		**0.8702**		0.7224
RE	0.6012		0.6714		0.6714		**0.7716**

Table 8
Clustering results for Lung Cancer data.

Cluster	Class		
	a	b	c
(a) Confusion matrix			
a	8	7	0
b	1	6	8
c	0	0	2
	Random	Wu	Cao	Proposed
(b) Performance comparison				
AC	**0.5210**	0.5000	0.5000	0.5000
PR	0.5766	0.5584	0.5584	**0.6444**
RE	0.5123	0.5014	0.5014	**0.5168**

Table 9
Clustering results for Mushroom data.

Cluster	Class			
	Poisonous			Edible
(a) Confusion matrix				
Poisonous	3052			98
Edible	864			4110
	Random	Wu	Cao	Proposed
(b) Performance comparison				
AC	0.7231	0.8754	0.8754	**0.8815**
PR	0.7614	**0.9019**	**0.9019**	0.8975
RE	0.7174	0.8709	0.8709	**0.8780**

Table 10
Clustering results for Congressional Vote data.

Cluster	Class	
	Republican	Democrat
(a) Confusion matrix		
Republican	158	55
Democrat	10	212
	Random	Proposed
(b) Performance comparison		
AC	0.4972	**0.8506**
PR	0.5030	**0.8484**
RE	0.5031	**0.8672**

- *Experiment3*: For some datasets, the *Vanilla* attributes are different from prominent attributes, for those cases we compared their clustering results.
- *Experiment4*: For all the three approaches to compute initial cluster centers i.e. *Vanilla*, *Prominent* and *Significant*, we computed the *matchMetric* to measure the quality of initial cluster centers in terms of their closeness to the actual modes or cluster centers of the data.
- *Experiment5*: We performed a scalability test by increasing the number of data objects $\approx$ 100,000 and recording the time spent in computing initial cluster centers. We also compared the order of time complexities of the proposed method with two other initialization methods.

Experiment1. Tables 5–11 show the clustering results, with confusion matrix representing the cluster structures obtained by seeding K-modes algorithm with the initial cluster centers computed using the proposed method. It can be seen that the proposed initialization method outperforms random cluster initialization when used as seed for K-modes clustering algorithm for the categorical data in accuracy, precision and recall. The random initialization method gives non-repeatable results, whereas the proposed method gives fixed clustering results. Therefore, repeatable and better cluster structures can be obtained by using the proposed method. In comparison to the initialization methods of Cao et al. and Wu et al., we evaluate our results in terms of:

- Accuracy – The proposed method outperforms or equals other methods in 4 cases and perform worse in one case.
- Precision – The proposed method performs well or equals other methods in 2 cases and performs worse in 3 cases.
- Recall – The proposed method outperforms or equals other methods in 4 cases and performs worse in 1 case.

The results for Congressional Vote and Dermatology data are not available from the papers from Cao et al. and Wu et al., therefore we compared the clustering accuracy of the proposed method against the random initialization method. The clustering results for

Table 11
Clustering results for Dermatology data.

Cluster	Class					
	Seboreic dermatitis	Psoriasis	Lichen planus	Cronic dermatitis	Pityriasis rosea	Pityriasis rubra pilaris
(*a*) *Confusion matrix*						
Seboreic dermatitis	53	7	5	2	36	0
Psoriasis	0	96	0	0	0	0
Lichen planus	0	0	66	0	0	0
Cronic dermatitis	2	1	0	39	0	0
Pityriasis rosea	6	8	1	11	13	4
Pityriasis rubra pilaris	0	0	0	0	0	16

	Random	Proposed
(b) Performance comparison		
AC	0.2523	**0.7732**
PR	0.2697	**0.7909**
RE	0.2954	**0.7570**

Table 12
Comparison of clustering results using *Prominent* and *Significant* attributes.

Dataset	*Prominent*			*Significant*		
	AC	PR	RE	AC	PR	RE
Soybean	**0.9574**	**0.9583**	**0.9705**	0.6809	0.7549	0.7176
Mushroom	**0.8815**	**0.8975**	**0.8780**	0.5086	0.7303	0.5256
Dermatology	**0.7732**	**0.7909**	**0.7570**	0.6502	0.5601	0.5512
Lung-Cancer	**0.5000**	**0.6444**	**0.5168**	0.46875	0.5079	0.4838
Zoo	0.8911	0.7224	0.7716	0.8911	0.7224	0.7716
Vote	0.8506	0.8484	0.8672	0.8506	0.8484	0.8672
Breast-Cancer	0.9127	0.9318	0.8783	0.9127	0.9318	0.8783

Table 13
Comparison of clustering results using *Vanilla* and *Prominent* attributes.

Dataset	*Vanilla*			*Prominent*		
	AC	PR	RE	AC	PR	RE
Soybean	**0.9787**	**0.9772**	**0.9853**	0.9574	0.9583	0.9705
Mushroom	0.6745	0.7970	0.6627	**0.8816**	**0.8976**	**0.87803**
Dermatology	0.4180	0.3889	0.3420	**0.7372**	**0.7909**	**0.7570**
Lung-Cancer	0.5000	6317	0.5017	**0.5**	**0.6444**	**0.5168**

Table 14
Comparison of *matchMetric* and its effect on K-modes algorithm convergence.

Dataset	Vanilla		*Prominent*		*Significant*	
	matchMetric	#Itr	*matchMetric*	#Itr	*matchMetric*	#Itr
(*a*) $p < m$						
Soybean	0.7357	2	**0.9643**	2	0.8428	2
Mushroom	0.6136	2	**0.8863**	2	0.5681	1
Dermatology	0.6176	5	**0.6813**	6	0.6274	6
Lug-Cancer	0.6726	6	0.7261	5	**0.7321**	8

Dataset	Vanilla	
	matchMetric	#Itr
(*b*) $p = m$		
Zoo	0.8661	2
Vote	0.7500	2
Breast-Cancer	0.8333	3

Dermatology data with random initialization are worse due to mixing up of data objects among various clusters.

The above results are very encouraging due to the fact that the proposed method attempts to find dense localized regions and discards the boundary cases, thus ensuring the selection of better initial cluster centers with log-linear worst-case time complexity. The method of Wu et al. induces random selection of data objects and Cao et al. can select boundary cases as initial cluster centers which can be detrimental to the clustering results. The accuracy values of proposed method are better than or equal to other methods. The only case where the proposed method perform worse in all three performance metric is the Soybean Small dataset. This dataset has only 47 data objects, our algorithm could not cluster only 2 data objects correctly. However, due to the small size of the dataset, the clustering error appears to be large.

We observe that on some datasets the proposed method gives worse values for precision, which implies that in those cases some data objects from non-classes are getting clustered in given classes. The recall values of proposed method are better than the other methods, which suggests that the proposed approach tightly controls the data objects from given classes to be not clustered to non-classes. Breast Cancer data has no prominent attribute in the data and uses all the attributes and produces comparable results to other methods. Lung Cancer data, though smaller in size has high dimension and the proposed method is able to produce better precision and recall rates than other methods. It is also observed that the proposed method performs well on large dataset such as Mushroom data with more than 8000 data objects. In our experiments we did not encounter a scenario where the distinct cluster strings are less than the desired number of clusters (step 7 of Algorithm 3).

Experiment2. Table 12 shows that for Zoo, Vote and Breast Cancer datasets, the clustering results using prominent and significant attributes are same. This is because all the attributes are considered in these datasets for computing the initial cluster centers. For other datasets, we observe that using prominent attributes is a better choice than significant attributes. Although we choose the same number of prominent and significant attributes (specially when all the attributes are not prominent), their clustering results varies because both of the attribute spaces may contain different set of attributes. The reason is that by definition (see Section 4), prominent and significant attributes use different criteria to choose relevant attributes for computing initial cluster centers. Moreover, generating the ranking of significant attributes is costlier in terms of time complexity than computing prominent attributes (see Section 4.4.3 for details).

Experiment3. As per Algorithm 1, for Zoo, Vote and Breast Cancer data all the attributes are prominent. For the rest of the datasets, this is not the case and the prominent attributes are less than the total number of attributes. We performed an experiment to analyze the scenario when there are fewer prominent attributes and its impact on overall clustering results. Table 13 shows that for all the datasets except Soybean Small, choosing prominent attributes less than the total number of attributes improve the clustering performance. Choosing all attributes in comparison to

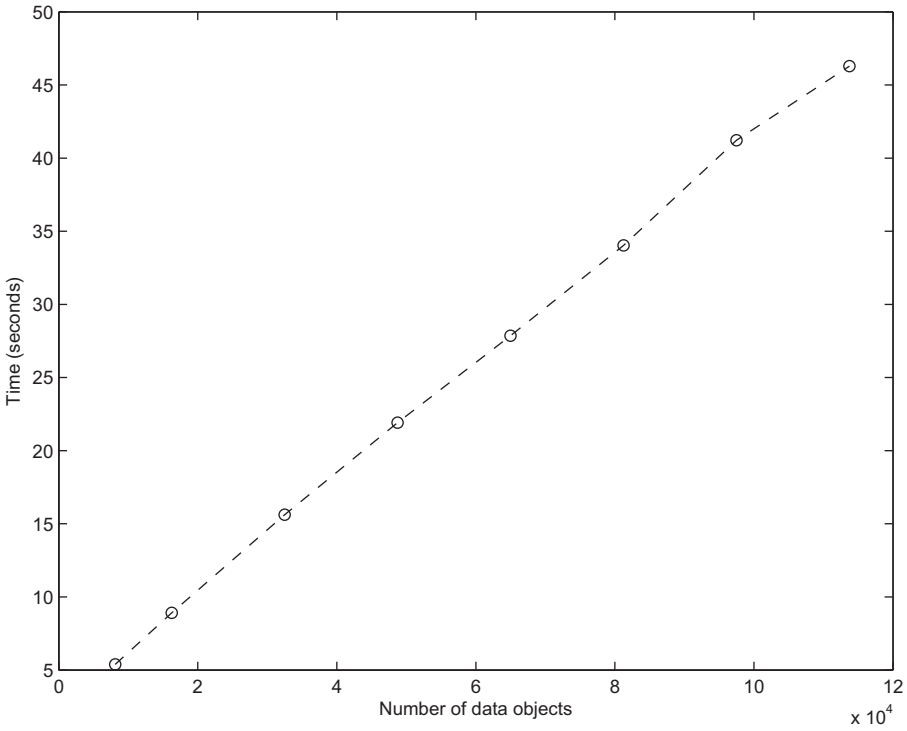

Fig. 1. Time consumption in computing initial cluster center for variable data size.

Table 15
Comparison of time complexities.

Initialization method	Order of complexity
Cao et al.	$O(nmK^2)$
Wu et al.	$O(cn)$, where c can be between 2 to $n^{0.5}$
Proposed method	$O(nm + rKm^2n + n\log n)$, for all/prominent attributes

fewer prominent attributes generates more cluster strings (see Table 4). If these cluster strings are more than $n^{0.5}$, then many relevant cluster strings may be not be chosen, which if included could have contributed in computation of initial cluster centers.

Experiment4. For all the datasets using the three methods of computing initial cluster centers, we computed the *matchMetric* (see Eq. (8)), which measures the degree of closeness of initial and actual modes. We also studied the impact of quality of initial cluster centers on the convergence of the K-modes algorithm (in terms of number of iterations, #Itr). Table 14(a) shows the case when prominent attributes are less than total number of attributes. The initial cluster centers selected by prominent/significant attributes are always closer to the actual modes of the datasets in terms of the *matchMetric* and therefore the K-modes algorithm converges in very few iterations with good cluster structures (see discussion of clustering results in *Experiment1*). Similar results were obtained when all the attributes are chosen as prominent attributes and used to compute initial cluster centers (see Table 14(b)). The high values of *matchMetric* show that the initial cluster centers are close to the actual cluster centers and the K-modes clustering algorithm with these initial cluster centers converges fast with good clustering performance. The reason for the initial cluster centers to be close to the actual cluster centers is that the proposed method finds dense localized clusters, merges them if needed and discards the insignificant clusters.

Experiment5. Time Scalability of the Proposed Algorithm. We performed an experiment to test the scalability of the proposed method for computing initial cluster centers for large datasets. We used the Mushroom dataset (see Section 5.1) that contains 8124 data objects described by 22 categorical attributes and 2 clusters. We made copies of this dataset in multiples of 2, 4, 6, 8, 10 and 12 such that the data size varies from 8124 to 113,736. We execute the proposed algorithm for computing initial cluster centers on each of these copies separately. We ran the experiment on a HP TouchSmart tm2 machine with Intel Pentium™ U4100 1.3 GHz processor, L2 cache 2048 KB and 4 GB RAM. Fig. 1 shows the plot between the different sizes of the data and the corresponding time consumed in computing the initial cluster centers. It can be observed that the time cost of the proposed method grows almost linearly with the increase in the number of data objects. The experimental results suggest that the proposed cluster center initialization method scales linearly and can be implemented for large datasets.

Table 15 compares the time complexities of the proposed cluster initialization algorithms with the two competing initialization methods of Cao et al. (2009) and Wu et al. (2007). In the proposed algorithm (with prominent attributes), if rKm^2 is larger than $\log n$ (which is more likely to be true for high dimensional dataset with large number of clusters), the complexity is decided by the second term, rKm^2n, which is linear in number of data objects and similar to Cao's method and better than Wu's method (with respect to the number of data objects). This linear time complexity behavior is also observed in our scalability experiments.

Multiple Attributes Clustering Challenges In Section 4, we mentioned some of the challenges in employing multiple clustering approaches (as defined by Müller et al. (2010)). The proposed method uses the multiple clustering approach to find initial cluster centers for a partitional clustering algorithm. The proposed approach successfully generated and detected the multiple clustering views from the data and can process different distinguishable clusters into relevant number of clusters by a modified hierarchical clustering approach or use them unaltered, whichever the case may be (as

discussed in Algorithm 3). In the worst-case, the proposed algorithm will generate clustering views equal to the number of total attributes in the data. This is a significant improvement over other approaches such as by Khan and Kant (2007), which can run arbitrary number of times for evidence accumulation. The proposed method is flexible and tested on various categorical datasets, however a known limitation is the advance knowledge of number of natural clusters in the data.

6. Conclusions

K-modes clustering algorithm is employed to partition the categorical data into pre-defined K clusters, however the clustering results intrinsically depend on the choice of random initial cluster centers, that can cause non-repeatable results and produce improper cluster structures. In this paper, we propose an algorithm to compute the initial cluster centers for the categorical data by performing multiple clustering of data based on the attribute values present in different attributes. The present algorithm is based on the experimental fact that similar data objects form the core of the clusters and are not affected by the selection of initial cluster centers, and that individual attribute also provides useful information in generating cluster structures. The proposed algorithm is composed of two parts – relevant attributes selection and computing initial cluster centers. For choosing the relevant attributes from the data, we presented two competitive methods. The first method chooses *Prominent* attributes on the basis of the attribute values present in an attribute and the second method computes the ranking of *Significant* attributes by an unsupervised learning method. Based on the selected attributes, the proposed algorithm partitions the data multiple times to generate multiple clustering views of the data. The multiplicity of clustering views is captured in the form of cluster strings, which produces distinct distinguishable clusters in the data that may be greater than, equal to or less than the desired number of clusters (K). If it is greater than K, then a modified hierarchical clustering is used to merge similar cluster strings into K clusters, if it is equal to K then the data objects corresponding to cluster strings can be directly used as initial cluster centers. An obscure possibility may arise when the cluster strings are less than K, in this case, it is assumed that the current value of K is not the true representative of the desired number of clusters. In our experiments we did not get such situation, largely because it can happen in a rare occurrence, when all the attribute values of different attributes cluster the data in the same way. These initial cluster centers when used as seed to K-modes clustering algorithm, improves the accuracy of the traditional K-modes clustering algorithm that uses random cluster centers as starting point. Since the proposed method provides a definitive choice of initial cluster centers (zero standard deviation), consistent and repetitive clustering results can be obtained. We also show that the initial cluster centers computed by using prominent attributes performs better than significant attribute selection approach and has the advantage of lower computational complexity. The initial cluster centers computed by the proposed approach are found to be very similar to the actual cluster centers of the data that leads to faster convergence of K-modes clustering algorithm and better clustering results. The performance of the proposed method is better than random initialization and better than or equal to the other two methods compared on all datasets except one case. The biggest advantage of the proposed method is the worst-case log-linear time complexity of computation and fixed choice of initial cluster centers from dense localized regions, whereas the other two methods lack one of them.

When the number of desired clusters is not available in advance, we would like to extend the proposed multi-clustering approach for the categorical data for finding out the natural number of clusters present in the data, in addition to computing the initial cluster centers for such cases. The present algorithm to compute *Prominent* attributes sometimes select all the attributes in the data, however our experiments indicate that considering fewer most relevant attributes is a better choice than choosing all attributes. We would like to further investigate such cases in future. We would like to extend the *Significant* attributes approach by ranking them according to their significance in the final consensus building instead of taking a fixed number of attributes. In other words, while computing the similarity of cluster strings in the merging algorithm, more weights will be given to the clustering results computed by using more significant attributes.

References

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In L. M. Haas & A. Tiwary (Eds.), *SIGMOD conference* (pp. 94–105). ACM Press.

Ahmad, A., & Dey, L. (2007a). A k-mean clustering algorithm for mixed numeric and categorical data. *Data Knowledge Engineering, 63*.

Ahmad, A., & Dey, L. (2007b). A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set. *Pattern Recognition Letters, 28*, 110–118.

Ahmad, A., & Dey, L. (2011). A k-means type clustering algorithm for subspace clustering of mixed numeric and categorical datasets. *Pattern Recognition Letters, 32*, 1062–1069.

Anderberg, M. R. (1973). *Cluster analysis for applications*. New York: Academic Press.

Bai, L., Liang, J., Dang, C., & Cao, F. (2012). A cluster centers initialization method for clustering categorical data. *Expert Systems with Applications, 39*, 8022–8029.

Boley, D., Gini, M., Gross, R., Han, E.-H., Karypis, G., Kumar, V., et al. (1999). Partitioning-based clustering for web document categorization. *Decision Support Systems, 27*, 329–341.

Bradley, P. S., & Fayyad, U. M. (1998). Refining initial points for k-means clustering. In J. W. Shavlik (Ed.), *ICML* (pp. 91–99). Morgan Kaufman.

Cao, F., Liang, J., & Bai, L. (2009). A new initialization method for categorical data clustering. *Expert Systems and Applications, 36*, 10223–10228.

Caruana, R., Elhawary, M. F., Nguyen, N., & Smith, C. (2006). Meta clustering. In *ICDM* (pp. 107–118). IEEE Computer Society.

Davidson, I., & Qi, Z. (2008). Finding alternative clusterings using constraints. In *ICDM* (pp. 773–778). IEEE Computer Society.

Bache, K., & Lichman, M., (2013). UCI machine learning repository, http://archive.ics.uci.edu/ml.

Fred, A. L. N., & Jain, A. K. (2002). Data clustering using evidence accumulation. In *ICPR (4)* (pp. 276–280).

Gowda, K. C., & Diday, E. (1991). Symbolic clustering using a new dissimilarity measure. *Pattern Recognition, 24*, 567–578.

Guha, S., Rastogi, R., & Shim, K. (1999). Rock: a robust clustering algorithm for categorical attributes. In *Proceedings of the 15th international conference on data engineering, 23–26 March 1999, Sydney, Austrialia* (pp. 512–521). IEEE Computer Society.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: an update. In *SIGKDD Explorations: Vol. 11 of 1*.

He, Z. (2006). Farthest-point heuristic based initialization methods for k-modes clustering. *CoRR, abs/cs/0610043*.

He, Z., Xu, X., & Deng, S. (2005). A cluster ensemble method for clustering categorical data. *Information Fusion, 6*, 143–151.

Huang, Z. (1997). A fast clustering algorithm to cluster very large categorical data sets in data mining. In *Research issues on data mining and knowledge discovery*.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery, 2*, 283–304.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc..

Ji, J., Pang, W., Zhou, C., Han, X., & Wang, Z. (2012). A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data. *Knowledge-Based Systems, 30*, 129–135.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: an introduction to cluster analysis*. John Wiley.

Khan, S. S., & Ahmad, A. (2004). Cluster center initialization algorithm for k-means clustering. *Pattern Recognition Letters, 25*, 1293–1302.

Khan, S. S., & Ahmad, A. (2003). Computing initial points using density based multiscale data condensation for clustering categorical data. In *Proceedings of 2nd international conference on applied artificial intelligence*.

Khan, S. S., & Ahmad, A. (2012). Cluster center initialization for categorical data using multiple attribute clustering. In E. Mülle, T. Seidl, S. Venkatasubramanian, & A. Zimek (Vol. Eds.), *Workshop proceedings of the 3rd multiclust workshop: discovering, summarizing and using multiple clusterings, USA* (pp. 3–10).

Khan, S. S., & Kant, S. (2007). Computation of initial modes for k-modes clustering algorithm using evidence accumulation. In *Proceedings of the 20th international joint conference on artificial intelligence (IJCAI)* (pp. 2784–2789).

Matas, J., & Kittler, J. (1995). Spatial and feature space clustering: applications in image analysis. In *CAIP* (pp. 162–173).

Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Density-based multiscale data condensation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 734–747.

Müller, E., Günnemann, S., Färber, I., & Seidl, T. (2010). Discovering multiple clustering solutions: grouping objects in different views of the data. In G. I. Webb, B. Liu, C. Zhang, D. Gunopulos, & X. Wu (Eds.), *ICDM* (pp. 1220). IEEE Computer Society.

Petrakis, E. G. M., & Faloutsos, C. (1997). Similarity searching in medical image databases. *IEEE Transactions on Knowledge Data Engineering, 9*, 435–447.

Ralambondrainy, H. (1995). A conceptual version of the k-means algorithm. *Pattern Recognition Letters, 16*, 1147–1157.

Sun, Y., Zhu, Q., & Chen, Z. (2002). An iterative initial-points refinement algorithm for categorical data clustering. *Pattern Recognition Letters, 23*, 875–884.

Ukkonen, E. (1995). On-line construction of suffix trees. *Algorithmica, 14*, 249–260.

Wu, S., Jiang, Q., & Huang, J. Z. (2007). A new initialization method for clustering categorical data. In *Proceedings of the 11th Pacific-Asia conference on advances in knowledge discovery and data mining PAKDD'07* (pp. 972–980). Berlin, Heidelberg: Springer-Verlag.