Glyph Based Visualization and Time Series Analysis for Software Project Management Data with Map Reduction

N.Zahira Jahan, MCA, M.Phil, S.Soundhirarasu
Associate Professor, Final year student
Master of computer Application, Nandha Engineering College (Anna University), Erode-52, India

Abstract — The k-means algorithm is well known for its effectiveness in clustering large data sets. However, working only on numeric values prohibit it from being used to cluster real world data containing categorical values. In this paper we present two algorithms which extend the k-means algorithm to categorical domains and domains with mixed numeric and categorical values. The k-modes algorithm uses a simple matching dissimilarity measure to deal with categorical objects, replaces the means of clusters with modes, and uses a frequency-based method to update modes in the clustering process to minimize the clustering cost function. With these extensions the k-modes algorithm enables the clustering of categorical data in a fashion similar to k-means. The k-prototypes algorithm, through the definition of a combined dissimilarity measure, further integrates the k-means and k-modes algorithms to allow for clustering objects described by mixed numeric and categorical attributes. We use the well known soybean disease and credit approval data sets to demonstrate the clustering performance of the two algorithms. Our experiments on two real world data set with half a million objects each show that the two algorithms are efficient when clustering large data sets, which is critical to data mining applications.

Keywords — k-means algorithm, k-modes algorithm, clusters, k-prototypes algorithm.

Introduction

Partition a set of objects in databases into standardized groups or clusters is a fundamental operation in data mining. Clustering is a popular approach to implementing the partitioning operation. This Clustering methods partition a set of objects into clusters such that objects in the same cluster are more similar to each other than objects in unlike clusters according to some clear criteria. Statistical clustering methods partition objects according to some (dis)similarity measures, whereas conceptual clustering methods cluster objects according to the concepts objects carry. The most distinct characteristic of data mining is that it deals with very large and complex data sets. The data sets to be mined often have millions of objects described by tens, hundreds or even thousands of different types of attributes or variables. Conceptual clustering algorithms developed in machine learning cluster data with categorical value and also produce conceptual descriptions of clusters. These algorithms are often revisions of some existing clustering methods. By using some carefully designed search methods, organizing structures and indices and these algorithms have shown some significant performance improvements in clustering very large data sets.

In this paper we present two new algorithms that use the k-means paradigm to cluster data having categorical values. The k-modes algorithm (extends the k-means paradigm to cluster categorical data by using (1) a simple matching difference measure for categorical objects, (2) modes instead of means for clusters and (3) a frequency-based method to update modes) in the k-means fashion clustering process to minimize the clustering cost function. The k-prototypes algorithm integrates the k-means and k-modes processes to cluster data with mixed numeric and categorical values. In the k-prototypes algorithm this define dissimilarity measure that takes into account both numeric and categorical attributes. The cluster process of the k-prototypes algorithm is similar to the k-means algorithm except that it uses the k-modes approach to updating the categorical attribute values of cluster prototypes. Since these algorithms use the same clustering process as k-means, they preserve the efficiency of the k-means algorithm which is highly desirable for data mining.

The major differences between CLARA and the k-prototypes algorithm are as follows: (1) CLARA clusters a large data set based on samples, whereas k-prototypes directly works on the whole data set. (2) CLARA optimizes its clustering result at the sample level. A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased. The k-prototypes
algorithm optimizes the cost function on the whole data set. It guarantees at least a locally optimal clustering. (3) The efficiency of CLARA depends on the sample size. The larger and more complex the whole data set is, the larger the sample is required. CLARA will no longer be efficient when the sample size exceeds a certain range, say thousands of objects. The k-prototypes algorithm has no such limitations.

1. Notation

We assume that in a database objects from the same domain are represent by the same as set of attribute, $A_1, A_2, ..., A_m$. Each attribute $A_i$ describes a domain of values, denoted by $DOM(A_i)$, associated with a defined semantic and data type. Different definitions of data types are used in data representation in databases and in data analysis. Simple data types commonly used in relational databases are integer, float, double, character and strings, where as data types concern data analysis are space, ratio, binary, ordinal, nominal, etc. According to the semantics of the attributes in the database one can always find a mapping between the related data types. In terms of the clustering algorithms to be discussed below, we only consider two general data types, numeric and categorical and assume other types can be mapped to one of these two types. The domains of attributes associated with these two types are called numeric and categorical, respectively.

A numeric domain represented by continuous values domain $DOM(A_i)$ is defined as categorical it is finite and unordered, e.g., for any $a, b \in DOM(A_i)$, either $a = b$ or $a \neq b$. A categorical domain contains only singletons. Combinational values like in are not allowed. A special value, denoted by $\exists$, is defined on all categorical domains and used to represent missing values. To simplify the dissimilarity measure we do not consider the conceptual inclusion relationships among values in a categorical domain like in such that car and vehicle are two categorical values in a domain and conceptually a car is also a vehicle. However, such relationships may exist in real world databases.

2. The k-means algorithm

The k-means algorithm one of the mostly used clustering algorithms is classified as a partitional or non hierarchical clustering method. Given a set of numeric objects $X$ and an integer number $k$ ($\leq n$), the k-means algorithm searches for a partition of $X$ into $k$ clusters that minimizes the within groups sum of squared errors (WGSS). This process is often formulated as the following mathematical program problem $P$

Problem $P$ can be generalized to allow $(w_{ij})^a$ where $w_{ij} \in [0,1], a \geq 1$

Problem $P$ can be solved by iteratively solving the following two problems:

1. Problem $P_1$: Fix $Q = Q^T$ and solve the reduced problem $P(W, \hat{Q})$.

2. Problem $P_2$: Fix $W = W^*$ and solve the reduced problem $P(W,Q)$.

Problem $P_1$ is solved by

$w_{ij} = 1$ if $d(X_i, Q_l) \leq d(X_i, Q_j)$, for $1 \leq t \leq k$

3) $w_{ij} = 0$ for $t \neq l$ and problem $P_2$ is solved by

(4) $l = 1$ will $\hat{P} = \hat{P}^*$

For $1 \leq l \leq k$, and $1 \leq j \leq m$.

The basic algorithm to solve problem $P$ is given as follows (Selim and Ismail, 1984; Bobrowski and Bezdek, 1991):

1. Choose an initial $Q^0$ and solve $P(W,Q^0)$ to obtain $W^0$. Set $t = 0$.

2. Let $\hat{W}^t = W$ and solve $P(W^*,Q)$ to obtain $Q^{t+1}$. If $P(W^*,Q) = P(W, Q^{t+1})$, output $\hat{W}, \hat{Q}$ and stop; otherwise, go to 3.

3. Let $\hat{Q}^t = Q^{t+1}$ and solve $P(W,Q^*)$ to obtain $W^{t+1}$. If $P(W,Q^*) = P(W^{t+1},Q^*)$, output $W, Q$ and stop; otherwise, let $t = t + 1$ and go to 2.

The k-means algorithm has the following important properties:

1. It is efficient in processing large data sets.

2. It often terminates at a local optimum (MacQueen, 1967; Selim and Ismail, 1984).

3. It works only on numeric values.

4. The clusters have convex shapes (Anderberg, 1973).

There exist a few variants of the k-means algorithm which differ in selection of the initial $k$ means, dissimilarity calculations and strategies to calculate cluster means. The sophisticated variants of the k-means algorithm include the well-known ISODATA algorithm and the fuzzy k-means algorithms. One difficulty in using the k-means algorithm is that the number of clusters has to be specified. Some variants like ISODATA include a procedure to search for the best $k$ at the cost of some performance.

3. The k-modes algorithm

In principle the formulation of problem $P$ in Section 3 is also valid for categorical and mixed data type objects. The cause that the k-means algorithm cannot cluster categorical objects is its dissimilarity measure and the method used to solve problem $P_2$. These barriers can be removed...
by making the following modifications to the k-means algorithm:

1. using a simple matching dissimilarity measure for categorical objects,
2. replacing means of clusters by modes, and
3. using a frequency-based method to find the modes to solve problem $P_2$.

4.1 Dissimilarity measure

Let $X, Y$ be two categorical objects described by $m$ categorical attributes. The dissimilarity measure between $X$ and $Y$ can be defined by the total mismatches of the corresponding attribute categories of the two objects. The smaller the number of mismatches is, the more similar the two objects. This measure is often referred to as simple matching mode of a data set $X$ is not unique. For example, the mode of set $\{[a, b], [a, c], [c, b], [b, c]\}$ can be either $[a, b]$ or $[a, c]$.

4.2 The k-modes algorithm

When (5) is used as the dissimilarity measure for categorical objects, the cost function (1) becomes

$$P(W, Q) = \sum_{i=1}^{k} \sum_{j=1}^{m} w_i l(\delta(x_i, q_i, j))$$

(8)

Where $\delta$, $W$ and $Q = [q_1, q_2, ..., q_m] \in Q$.

In the basic algorithm we need to calculate the total cost $P$ against the whole data set each time when a new $Q$ or $W$ is obtained. To make the computation more efficient we use the following algorithm instead in practice.

1. Select $k$ initial modes, one for each cluster.
2. Allocate an object to the cluster whose mode is the nearest to it according to (5). Update the mode of the cluster after each allocation according to Theorem 1.
3. After all objects have been allocated to clusters, 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, reallocating the object to that cluster and update the modes of both clusters.
4. Repeat 3 until no object has changed clusters after a full cycle test of the whole data set.

4. The k-prototypes algorithm

It is straightforward to integrate the k-means and k-modes algorithms into the k-prototypes

5.1.1. Real world data sets.

The first data set was the soybean disease data set, which has frequently been used to test conceptual clustering algorithms. We chose this data set to test the k-modes algorithm because all its attributes can be treated as categorical.

The second data set was the credit approval data set (Quinlan, 1993). We chose this data set to test the k-prototypes algorithm. The data set has 690 instances, each being described by 6 numeric and 9 categorical attributes. The instances were classified...
into two classes, approved labeled as "+" and rejected labeled as "-".

Thirty seven instances have missing values in seven attributes. Since our current implementation of the $k$-prototypes algorithm cannot handle missing values in numeric attributes, 24 instances with missing values in numeric attributes were removed.

Selection methods and different $\gamma$ values. In clustering we rescaled all numeric attributes to the range of $[0, 1]$. The average of standard deviations of all numeric attributes is 0.114. The tested $\gamma$ values ranged from 0 to 6.

6. Conclusions

The most attractive property of the $k$-means algorithm in data mining is its efficiency in clustering large data sets. However, that it only works on numeric data limits its use in many data mining applications because of the involvement of categorical data. The $k$-modes and $k$-prototypes algorithms have removed this limitation and extended the $k$-means paradigm into a generic data partitioning operation for data mining.

The clustering performance of the two algorithms has been evaluated using two real world data sets. The satisfactory results have demonstrated the effectiveness of the two algorithms in discovering structures in data. The scalability tests have shown that the two algorithms are efficient when clustering very large complex data sets in terms of both the number of records and the number of clusters. These properties are very important to data mining. In general, the $k$-modes algorithm is faster than the $k$-means and $k$- prototypes algorithm because it needs less iteration to converge.

This paper has focused on the technical issues of extending the $k$-means algorithm to cluster data with categorical values. Although we have demonstrated that the two new algorithms work well on two known data sets, we have to acknowledge that this mainly resulted from our a priori knowledge to the data sets. In practical applications such a priori knowledge is rarely available. Therefore, in using the two algorithms to solve practical data mining problems, we still face the common problem: How many clusters are in the data? There exist a number of techniques which tackle this problem (Milligan and Cooper, 1985; Dubes, 1987). These techniques are directly applicable to the two algorithms presented.

The weight $\gamma$ adds an additional problem to the use of the $k$-prototypes algorithm. We have suggested that the average standard deviation of numeric attributes may be used as guidance in specifying $\gamma$ (Huang, 1997a). However, it is too early to consider this as a general rule. The user’s knowledge to the data is important in specifying $\gamma$. If one thinks the clustering should be favored on numeric attributes, then one needs a small $\gamma$. If one believes categorical attributes are important, then one needs a large $\gamma$. Acknowledgments

The author wishes to acknowledge that this work was carried out within the Cooperative Research Centre for Advanced Computational Systems.

![Figure 2. Two misclassification matrices: (a) correspondence between clusters of test data set 1 and disease classes, and (b) correspondence between clusters of test data set 9 and disease classes.](http://www.ijettjournal.org)
established under the Australian Government’s Cooperative Research Centers Program.

The author is grateful to Murray Cameron at CSIRO for his critical review of the paper.

References


