# A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining

**Zhexue Huang***
Cooperative Research Centre for Advanced Computational Systems
CSIRO Mathematical and Information Sciences
GPO Box 664, Canberra 2601, AUSTRALIA
email:Zhexue.Huang@cmis.csiro.au

### Abstract

Partitioning a large set of objects into homogeneous clusters is a fundamental operation in data mining. The *k*-means algorithm is best suited for implementing this operation because of its efficiency in clustering large data sets. However, working only on numeric values limits its use in data mining because data sets in data mining often contain categorical values. In this paper we present an algorithm, called *k*-modes, to extend the *k*-means paradigm to categorical domains. We introduce new dissimilarity measures to deal with categorical objects, replace means of clusters with modes, and use a frequency based method to update modes in the clustering process to minimise the clustering cost function. Tested with the well known soybean disease data set the algorithm has demonstrated a very good classification performance. Experiments on a very large health insurance data set consisting of half a million records and 34 categorical attributes show that the algorithm is scalable in terms of both the number of clusters and the number of records.

## 1 Introduction

Partitioning a set of objects into homogeneous clusters is a fundamental operation in data mining. The operation is needed in a number of data mining tasks, such as unsupervised classification and data summation, as well as segmentation of large heterogeneous data sets into smaller homogeneous subsets that can be easily managed, separately modelled and analysed. Clustering is a popular approach used to implement this operation. 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 different clusters according to some defined criteria. Statistical clustering methods (Anderberg 1973, Jain and Dubes 1988) use similarity measures to partition objects whereas conceptual clustering methods cluster objects according to the concepts objects carry (Michalski and Stepp 1983, Fisher 1987).

The most distinct characteristic of data mining is that it deals with very large data sets (gigabytes or even terabytes). This requires the algorithms used in data mining to be scalable. However, most algorithms currently used in data mining do not scale well when applied to very large data sets because they were initially developed for other applications than data mining which involve small data sets. The study of scalable data mining algorithms has recently become a data mining research focus (Shafer et al. 1996).

In this paper we present a fast clustering algorithm used to cluster categorical data. The algorithm, called *k-modes*, is an extension to the well known *k-means* algorithm (MacQueen 1967). Compared to other clustering methods the *k*-means algorithm and its variants (Anderberg 1973) are efficient in clustering large data sets, thus very suitable for data mining. However, their use is often limited to numeric data because these algorithms minimise a cost function by calculating the means of clusters. Data mining applications frequently involve categorical data. The traditional approach to converting categorical data into numeric values does not necessarily produce meaningful results in the case where categorical domains are not ordered. The *k*-modes algorithm in this paper removes this limitation and extends the *k*-means paradigm to categorical domains whilst preserving the efficiency of the *k*-means algorithm.

In (Huang 1997) we have proposed an algorithm, called *k-prototypes*, to cluster large data sets with mixed numeric and categorical values. In the *k*-prototypes algorithm we define a dissimilarity measure that takes into account both numeric and categorical attributes. Assume $s_n$ is the dissimilarity measure on numeric attributes defined by the squared Euclidean distance and $s_c$ is the dissimilarity measure on categorical attributes defined as the number of mismatches of categories between two objects. We define the dissimilarity measure between two objects as $s_n + \gamma s_c$, where $\gamma$ is a weight to balance the two parts to avoid favouring either type of attribute. The clustering process of the *k*-prototypes algorithm is similar to the *k*-means algorithm except that a new method is used to update the categorical attribute values of cluster

* The author wishes to acknowledge that this work was carried out within the Cooperative Research Centre for Advanced Computational Systems (ACSys) established under the Australian Government's Cooperative Research Centres Program.

prototypes. A problem in using that algorithm is to choose a proper weight. We have suggested the use of the average standard deviation of numeric attributes as a guide in choosing the weight.

The *k*-modes algorithm presented in this paper is a simplification of the *k*-prototypes algorithm by only taking categorical attributes into account. Therefore, weight $\gamma$ is no longer necessary in the algorithm because of the disappearance of $s_n$. If numeric attributes are involved in a data set, we categorise them using a method as described in (Anderberg 1973). The biggest advantage of this algorithm is that it is scalable to very large data sets. Tested with a health insurance data set consisting of half a million records and 34 categorical attributes, this algorithm has shown a capability of clustering the data set into 100 clusters in about a hour using a single processor of a Sun Enterprise 4000 computer.

Ralambondrainy (1995) presented another approach to using the *k*-means algorithm to cluster categorical data. Ralambondrainy's approach needs to convert multiple category attributes into binary attributes (using 0 and 1 to represent either a category absent or present) and to treat the binary attributes as numeric in the *k*-means algorithm. If it is used in data mining, this approach requires to handle a large number of binary attributes because data sets in data mining often have categorical attributes with hundreds or thousands of categories. This will inevitably increase both computational and space costs of the *k*-means algorithm. The other drawback is that the cluster means, given by real values between 0 and 1, do not indicate the characteristics of the clusters. Comparatively, the *k*-modes algorithm directly works on categorical attributes and produces the cluster modes, which describe the clusters, thus very useful to the user in interpreting the clustering results.

Using Gower's similarity coefficient (Gower 1971) and other dissimilarity measures (Gowda and Diday 1991) one can use a hierarchical clustering method to cluster categorical or mixed data. However, the hierarchical clustering methods are not efficient in processing large data sets. Their use is limited to small data sets.

The rest of the paper is organised as follows. Categorical data and its representation are described in Section 2. In Section 3 we briefly review the *k*-means algorithm and its important properties. In Section 4 we discuss the *k*-modes algorithm. In Section 5 we present some experimental results on two real data sets to show the classification performance and computational efficiency of the *k*-modes algorithm. We summarise our discussions and describe our future work plan in Section 6.

# 2 Categorical Data

Categorical data as referred to in this paper is the data describing objects which have only categorical attributes. The objects, called categorical objects, are a simplified version of the symbolic objects defined in (Gowda and Diday 1991). We consider all numeric (quantitative) attributes are categorised and do not consider categorical attributes that have combinational values, e.g., Languages-spoken (Chinese, English). The following two subsections define the categorical attributes and objects accepted by the algorithm.

## 2.1 Categorical Domains and Attributes

Let $A_1$, $A_2$, …, $A_m$ be *m* attributes describing a space $\boldsymbol{\Omega}$ and DOM($A_1$), DOM($A_2$), …, DOM($A_m$) the domains of the attributes. A domain DOM($A_j$) is defined as categorical if it is finite and unordered, e.g., for any $a, b \in$ DOM($A_j$), either $a = b$ or $a \neq b$. $A_j$ is called a categorical attribute. $\boldsymbol{\Omega}$ is a categorical space if all $A_1$, $A_2$, …, $A_m$ are categorical.

A categorical domain defined here contains only singletons. Combinational values like in (Gowda and Diday 1991) are not allowed. A special value, denoted by ε, 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 (Kodratoff and Tecuci 1988) 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.

## 2.2 Categorical Objects

Like in (Gowda and Diday 1991) a categorical object $X \in \boldsymbol{\Omega}$ is logically represented as a conjunction of attribute-value pairs $[A_1 = x_1] \wedge [A_2 = x_2] \wedge \ldots \wedge [A_m = x_m]$, where $x_j \in$ DOM($A_j$) for $1 \leq j \leq m$. An attribute-value pair $[A_j = x_j]$ is called a *selector* in (Michalski and Stepp 1983). Without ambiguity we represent $X$ as a vector $[x_1, x_2, \ldots, x_m]$. We consider every object in $\boldsymbol{\Omega}$ has exactly *m* attribute values. If the value of attribute $A_j$ is not available for an object $X$, then $A_j = \varepsilon$.

Let $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$ be a set of *n* categorical objects and $\boldsymbol{X} \subseteq \boldsymbol{\Omega}$. Object $X_i$ is represented as $[x_{i,1}, x_{i,2}, \ldots, x_{i,m}]$. We write $X_i = X_k$ if $x_{i,j} = x_{k,j}$ for $1 \leq j \leq m$. The relation $X_i = X_k$ does not mean that $X_i$, $X_k$ are the same object in the real world database. It means the two objects have equal categorical values in attributes $A_1$, $A_2$, …, $A_m$. For example, two patients in a data set may have equal values in attributes Sex, Disease and Treatment. However, they are distinguished in the hospital database by other attributes such as ID and Address which were not selected for clustering.

Assume $\boldsymbol{X}$ consists of $n$ objects in which $p$ objects are distinct. Let $N$ be the cardinality of the Cartesian product $\mathrm{DOM}(A_1) \times \mathrm{DOM}(A_2) \times \ldots \times \mathrm{DOM}(A_m)$. We have $p \le N$. However, $n$ may be larger than $N$, which means there are duplicates in $\boldsymbol{X}$.

# 3 The *K*-means Algorithm

The *k*-means algorithm (MacQueen 1967, Anderberg 1973) is built upon four basic operations: (1) selection of the initial $k$ means for $k$ clusters, (2) calculation of the dissimilarity between an object and the mean of a cluster, (3) allocation of an object to the cluster whose mean is nearest to the object, (4) Re-calculation of the mean of a cluster from the objects allocated to it so that the intra cluster dissimilarity is minimised. Except for the first operation, the other three operations are repeatedly performed in the algorithm until the algorithm converges.

The essence of the algorithm is to minimise the cost function

$$E = \sum_{l=1}^{k} \sum_{i=1}^{n} y_{i,l} d(X_i, Q_l) \qquad (1)$$

where $n$ is the number of objects in a data set $\boldsymbol{X}$, $X_i \in \boldsymbol{X}$, $Q_l$ is the mean of cluster $l$, and $y_{i,l}$ is an element of a *partition* matrix $\boldsymbol{Y}_{n \times l}$ as in (Hand 1981). $d$ is a dissimilarity measure usually defined by the squared Euclidean distance.

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 (Anderberg 1973, Bobrowski and Bezdek 1991). The sophisticated variants of the *k*-means algorithm include the well-known ISODATA algorithm (Ball and Hall 1967) and the fuzzy *k*-means algorithms (Ruspini 1969, 1973).

Most *k*-means type algorithms have been proved convergent (MacQueen 1967, Bezdek 1980, Selim and Ismail 1984). The *k*-means algorithm has the following important properties.

1. It is efficient in processing large data sets. The computational complexity of the algorithm is $O(tkmn)$, where $m$ is the number of attributes, $n$ is the number of objects, $k$ is the number of clusters, and $t$ is the number of iterations over the whole data set. Usually, $k, m, t << n$. In clustering large data sets the *k*-means algorithm is much faster than the hierarchical clustering algorithms whose general computational complexity is $O(n^2)$ (Murtagh 1992).
2. It often terminates at a local optimum (MacQueen 1967, Selim and Ismail 1984). To find out the global optimum, techniques such as deterministic annealing (Kirkpatrick et al. 1983, Rose et al. 1990) and genetic algorithms (Goldberg 1989, Murthy and Chowdhury 1996) can be incorporated with the *k*-means algorithm.
3. It works only on numeric values because it minimises a cost function by calculating the means of clusters.
4. The clusters have convex shapes (Anderberg 1973). Therefore, it is difficult to use the *k*-means algorithm to discover clusters with non-convex shapes.

One difficulty in using the *k*-means algorithm is to specify the number of clusters. Some variants like ISODATA include a procedure to search for the best $k$ at the cost of some performance.

The *k*-means algorithm is best suited for data mining because of its efficiency in processing large data sets. However, working only on numeric values limits its use in data mining because data sets in data mining often have categorical values. Development of the *k*-modes algorithm to be discussed in the next section was motivated by the desire to remove this limitation and extend its use to categorical domains.

# 4 The *K*-modes Algorithm

The *k*-modes algorithm is a simplified version of the *k*-prototypes algorithm described in (Huang 1997). In this algorithm we have made three major modifications to the *k*-means algorithm, i.e., using different dissimilarity measures, replacing $k$ means with $k$ modes, and using a frequency based method to update modes. These modifications are discussed below.

## *4.1 Dissimilarity Measures*

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. Formally,

$$d(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j) \qquad (2)$$

where

$$\delta(x_j, y_j) = \begin{cases} 0 & (x_j = y_j) \\ 1 & (x_j \neq y_j) \end{cases} \qquad (3)$$

$d(X,Y)$ gives equal importance to each category of an attribute. If we take into account the frequencies of categories in a data set, we can define the dissimilarity measure as

$$d_{\chi^2}(X,Y) = \sum_{j=1}^{m} \frac{(n_{x_j} + n_{y_j})}{n_{x_j} n_{y_j}} \delta(x_j, y_j) \qquad (4)$$

where $n_{x_j}$, $n_{y_j}$ are the numbers of objects in the data set that have categories $x_j$ and $y_j$ for attribute $j$. Because

$d_{\chi^2}(X,Y)$ is similar to the chi-square distance in (Greenacre 1984), we call it *chi-square distance*. This dissimilarity measure gives more importance to rare categories than frequent ones. Eq. (4) is useful in discovering under-represented object clusters such as fraudulent claims in insurance databases.

### 4.2 Mode of a Set

Let $\boldsymbol{X}$ be a set of categorical objects described by categorical attributes $A_1, A_2, \ldots, A_m$.

**Definition**: *A mode of* $\boldsymbol{X}$ *is a vector* $Q = [q_1, q_2, \ldots, q_m] \in \boldsymbol{\Omega}$ *that minimises*

$$D(Q, \boldsymbol{X}) = \sum_{i=1}^{n} d(X_i, Q) \quad (5)$$

*where* $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$ *and d can be either defined as in Eq.* (2) *or in Eq.* (4). *Here, Q is not necessarily an element of* $\boldsymbol{X}$.

### 4.3 Find a Mode for a Set

Let $n_{c_{k,j}}$ be the number of objects having category $c_{k,j}$ in attribute $A_j$ and $f_r(A_j = c_{k,j} | \boldsymbol{X}) = \frac{n_{c_{k,j}}}{n}$ the relative frequency of category $c_{k,j}$ in $\boldsymbol{X}$.

**Theorem**: *The function D(Q,**X**) is minimised iff* $f_r(A_j = q_j | \boldsymbol{X}) \geq f_r(A_j = c_{k,j} | \boldsymbol{X})$ *for* $q_j \neq c_{k,j}$ *for all j* = 1..*m*.

The proof of the theorem is given in the Appendix.

The theorem defines a way to find $Q$ from a given $\boldsymbol{X}$, and therefore is important because it allows to use the *k*-means paradigm to cluster categorical data without losing its efficiency. The theorem implies that the mode of a data set $\boldsymbol{X}$ 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*].

### 4.4 The *k*-modes *Algorithm*

Let $\{S_1, S_2, \ldots, S_k\}$ be a partition of $\boldsymbol{X}$, where $S_l \neq \emptyset$ for $1 \leq l \leq k$, and $\{Q_1, Q_2, \ldots, Q_k\}$ the modes of $\{S_1, S_2, \ldots, S_k\}$. The total cost of the partition is defined by

$$E = \sum_{l=1}^{k} \sum_{i=1}^{n} y_{i,l} d(X_i, Q_l) \quad (6)$$

where $y_{i,l}$ is an element of a partition matrix $\boldsymbol{Y}_{n \times l}$ as in (Hand 1981) and *d* can be either defined as in Eq. (2) or in Eq. (4).

Similar to the *k*-means algorithm, the objective of clustering $\boldsymbol{X}$ is to find a set $\{Q_1, Q_2, \ldots, Q_k\}$ that can minimise *E*. Although the form of this cost function is the same as Eq. (1), *d* is different. Eq. (6) can be minimised by the *k*-modes algorithm below.

The *k*-modes algorithm consists of the following steps (refer to (Huang 1997) for the detailed description of the algorithm):

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 *d*. Update the mode of the cluster after each allocation according to the Theorem.
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, reallocate 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.

Like the *k*-means algorithm the *k*-modes algorithm also produces locally optimal solutions that are dependent on the initial modes and the order of objects in the data set. In Section 5 we use a real example to show how appropriate initial mode selection methods can improve the clustering results.

In our current implementation of the *k*-modes algorithm we include two initial mode selection methods. The first method selects the first *k* distinct records from the data set as the initial *k* modes. The second method is implemented in the following steps.

1. Calculate the frequencies of all categories for all attributes and store them in a category array in the descending order of frequency as shown in Figure 1. Here, $c_{i,j}$ denotes category *i* of attribute *j* and $f(c_{i,j}) \geq f(c_{i+1,j})$ where $f(c_{i,j})$ is the frequency of category $c_{i,j}$.

$$\begin{Bmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} \\ c_{3,1} & & c_{3,3} & c_{3,4} \\ c_{4,1} & & c_{4,3} & \\ & & c_{5,3} & \end{Bmatrix}$$

**Figure 1.** The category array of a data set with 4 attributes having 4, 2, 5, 3 categories respectively.

2. Assign the most frequent categories equally to the initial *k* modes. For example in Figure 1, assume *k* = 3. We assign $Q_1 = [q_{1,1}=c_{1,1}, q_{1,2}=c_{2,2}, q_{1,3}=c_{3,3}, q_{1,4}=c_{1,4}]$, $Q_2 = [q_{2,1}=c_{2,1}, q_{2,2}=c_{1,2}, q_{2,3}=c_{4,3}, q_{2,4}=c_{2,4}]$ and $Q_3 = [q_{3,1}=c_{3,1}, q_{3,2}=c_{2,2}, q_{3,3}=c_{1,3}, q_{3,4}=c_{3,4}]$.
3. Start with $Q_1$. Select the record most similar to $Q_1$ and substitute $Q_1$ with the record as the first initial

mode. Then select the record most similar to $Q_2$ and substitute $Q_2$ with the record as the second initial mode. Continue this process until $Q_k$ is substituted. In these selections $Q_l \neq Q_t$ for $l \neq t$.

Step 3 is taken to avoid the occurrence of empty clusters. The purpose of this selection method is to make the initial modes diverse, which can result in better clustering results (see Section 5.1.3).

# 5 Experimental Results

We used the well known soybean disease data to test classification performance of the algorithm and another large data set selected from a health insurance database to test computational efficiency of the algorithm. The second data set consists of half a million records, each being described by 34 categorical attributes.

## *5.1 Tests on Soybean Disease Data*

### 5.1.1 Test Data Sets

The soybean disease data is one of the standard test data sets used in the machine learning community. It has often been used to test conceptual clustering algorithms (Michalski and Stepp 1983, Fisher 1987). We chose this data set to test our algorithm because of its publicity and because all its attributes can be treated as categorical without categorisation.

The soybean data set has 47 observations, each being described by 35 attributes. Each observation is identified by one of the 4 diseases -- Diaporthe Stem Canker, Charcoal Rot, Rhizoctonia Root Rot, and Phytophthora Rot. Except for Phytophthora Rot which has 17 observations, all other diseases have 10 observations each. Eq. (2) was used in the tests because all disease classes are almost equally distributed. Of the 35 attributes we only selected 21 because the other 14 have only one category.

To study the effect of record order, we created 100 test data sets by randomly reordering the 47 observations. By doing this we were also selecting different records for the initial modes using the first selection method. All disease identifications were removed from the test data sets.

### 5.1.2 Clustering Results

We used the *k*-modes algorithm to cluster each test data set into 4 clusters with the two initial mode selection methods and produced 200 clustering results. For each clustering result we used a misclassification matrix to analyse the correspondence between clusters and the disease classes of the observations. Two misclassification matrices for the test data sets 1 and 9 are shown in Figure 2. The capital letters D, C, R, P in the first column of the matrices represent the 4 disease classes. In figure 2(a) there is one to one correspondence between clusters and disease classes, which means the observations in the same disease classes were clustered into the same clusters. This represents a complete recovery of the 4 disease classes from the test data set.

In Figure 2(b) two observations of the disease class P were misclassified into cluster 1 which was dominated by the observations of the disease class R. However, the observations in the other two disease classes were correctly clustered into clusters 3 and 4. This clustering result can also be considered good.

| | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 |
|---|---|---|---|---|
| D | | | 10 | |
| C | | | | 10 |
| R | 10 | | | |
| P | | 17 | | |

(a)

| | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 |
|---|---|---|---|---|
| D | | | | 10 |
| C | | | 10 | |
| R | 10 | | | |
| P | 2 | 15 | | |

(b)

**Figure 2.** Two misclassification matrices. (a) Correspondence between clusters of test data set 1 and disease classes. (b) Correspondence between clusters of test data set 9 and disease classes.

If we use the number of misclassified observations as a measure of a clustering result, we can summarise the 200 clustering results in Table 1. The first column in the table gives the number of misclassified observations. The second and third columns show the numbers of clustering results.

**Table 1**.

| Misclassified Observations | First Selection Method | Second Selection Method |
|---|---|---|
| 0 | 13 | 14 |
| 1 | 7 | 8 |
| 2 | 12 | 26 |
| 3 | 4 | 9 |
| 4 | 7 | 6 |
| 5 | 2 | 1 |
| >5 | 55 | 36 |

If we consider the number of misclassified observations less than 6 as a "good" clustering result, then 45 good results were produced with the first selection method and 64 good results with the second selection method. Both selection methods produced more than 10 complete recovery results (0 misclassification). These results indicate that if we randomly choose one test data set, we have a 45% chance to obtain a good clustering result with the first selection method and a 64% chance with the second selection method.

Table 2 shows the relationships between the clustering results and the clustering costs (values of Eq. (6)). The numbers in brackets are the numbers of clustering results having the corresponding clustering cost values. All total mismatches of "bad" clustering results are greater than those of "good" clustering results. The minimal total mismatch number in these tests is 194 which is likely the global minimum. These relationships indicate that we can use the clustering cost values from several runs to choose a good clustering result if the original classification of data is unknown.

We did the same tests using a *k*-means algorithm which is based on the versions 3 and 5 of subroutine KMEAN in (Anderberg 1973). In these tests we simply treated all attributes as numeric and used the squared Euclidean distance as the dissimilarity measure. The initial means were selected by the first method. Of 100 clustering results we only got 4 good ones of which 2 had a complete recovery. Comparing the cost values of the 4 good clustering results with other clustering results, we found that the clustering results and the cost values are not related. Therefore, a good clustering result cannot be selected according to its cost value.

**Table 2.**

| Misclassified Observations | Total mismatches for method 1 | Total mismatches for method 2 |
|---|---|---|
| 0 | 194(13) | 194(14) |
| 1 | 194(7) | 194(7), 197(1) |
| 2 | 194(12) | 194(25),195(1) |
| 3 | 195(2),197(1), 201(1) | 195(6),196(2),197(1) |
| 4 | 195(2),196(3),197(2) | 195(4),196(1),197(1) |
| 5 | 197(2) | 197(1) |
| >5 | 203-261 | 209-254 |

**Table 3.**

| No. of classes | No. of runs | Mean cost | Std Dev |
|---|---|---|---|
| 1 | 1 | 247 | - |
| 2 | 28 | 222.3 | 24.94 |
| 3 | 66 | 211.9 | 19.28 |
| 4 | 5 | 194.6 | 1.34 |

The effect of initial modes on clustering results is shown in Table 3. The first column is the number of disease classes the initial modes have and the second is the corresponding number of runs with the number of disease classes in the initial modes. This table indicates that the more diverse the disease classes are in the initial modes, the better the clustering results. The initial modes selected by the second method have 3 disease types, therefore more good cluster results were produced than by the first method.

From the modes and category distributions of different attributes in different clusters the algorithm can also produce discriminative characteristics of clusters similar to those in (Michalski and Stepp 1983).

## 5.2 Tests on a Large Data Set

The purpose of this experiment was to test the scalability of the *k*-modes algorithm in clustering very large real world data sets. We selected a large data set from a health insurance database. The data set consists of 500000 records, each being described by 34 categorical attributes in which 4 have more than 1000 categories each.

We tested two scalabilities of the algorithm using this large data set. The first one is the scalability of the algorithm against the number of clusters for a given number of objects and the second is the scalability against the number of objects for a given number of clusters. Figures 3 and 4 show the results produced using a single processor of a Sun Enterprise 4000 computer. The plots in the figures represent the average time performance of 5 independent runs.

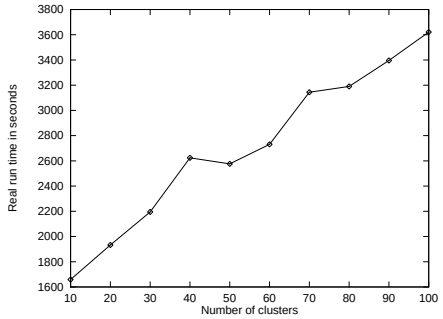


**Figure 3.** Scalability to the number of clusters in clustering 500000 records.

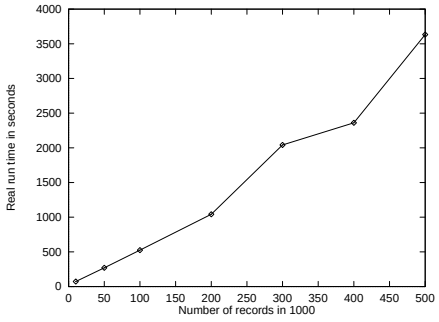


**Figure 4.** Scalability to the number of records clustered into 100 clusters.

These results are very encouraging because they show clearly a linear increase in time as both the number of clusters and number of records increase. Clustering half a million objects into 100 clusters took about a hour, which is quite acceptable. Compared with the results of clustering data with mixed values (Huang 1997), this algorithm is much faster than its previous version because it needs many less iterations to converge.

The above soybean disease data tests indicate that a good clustering result should be selected from multiple runs of the algorithm over the same data set with different record orders and/or different initial modes. This can be done in practice by running the algorithm in parallel on a parallel computing system. Other parts of the algorithm such as the operation to allocate an object to a cluster can also be parallelised to improve the performance.

# 6 Summary and Future Work

The biggest advantage of the *k*-means algorithm in data mining applications is its efficiency in clustering large data sets. However, its use is limited to numeric values. The *k*-modes algorithm presented in this paper has removed this limitation whilst preserving its efficiency.

The *k*-modes algorithm has made the following extensions to the *k*-means algorithm:

1. replacing means of clusters with modes,
2. using new dissimilarity measures to deal with categorical objects, and
3. using a frequency based method to update modes of clusters.

These extensions allow us to use the *k*-means paradigm directly to cluster categorical data without need of data conversion.

Another advantage of the *k*-modes algorithm is that the modes give characteristic descriptions of clusters. These descriptions are very important to the user in interpreting clustering results.

Because data mining deals with very large data sets, scalability is a basic requirement to the data mining algorithms. Our experimental results have demonstrated that the *k*-modes algorithm is indeed scalable to very large and complex data sets in terms of both the number of records and the number of clusters. In fact the *k*-modes algorithm is faster than the *k*-means algorithm because our experiments have shown that the former often needs less iterations to converge than the later.

Our future work plan is to develop and implement a parallel *k*-modes algorithm to cluster data sets with millions of objects. Such an algorithm is required in a number of data mining applications, such as partitioning very large heterogeneous sets of objects into a number of smaller and more manageable homogeneous subsets that can be more easily modelled and analysed, and detecting under-represented concepts, e.g., fraud in a very large number of insurance claims.

## Acknowledgments

The author is grateful to Dr Markus Hegland at The Australian National University, Mr Peter Milne and Dr Graham Williams at CSIRO for their comments on the paper.

## References

**Anderberg, M. R.** (1973) *Cluster Analysis for Applications*, Academic Press.

**Ball, G. H. and Hall, D. J.** (1967) *A Clustering Technique for Summarizing Multivariate Data*, Behavioral Science, **12**, pp. 153-155.

**Bezdek, J. C.** (1980) *A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms*, IEEE Transactions on Pattern Analysis and Machine Intelligence**, 2**(8), pp. 1-8.

**Bobrowski, L. and Bezdek, J. C.** (1991) *c-Means Clustering with the $l_1$ and $l_\infty$ Norms*, IEEE Transactions on Systems, Man and Cybernetics, **21**(3), pp. 545-554.

**Fisher, D. H.** (1987) *Knowledge Acquisition Via Incremental Conceptual Clustering*, Machine Learning, **2**(2), pp.139-172.

**Goldberg, D. E.** (1989) *Genetic Algorithms in Search, Optimisation, and Machine Learning*, Addison-Wesley.

**Gowda, K. C. and Diday, E.** (1991) *Symbolic Clustering Using a New Dissimilarity Measure*, Pattern Recognition, **24**(6), pp. 567-578.

**Gower, J. C.** (1971) *A General Coefficient of Similarity and Some of its Properties*, BioMetrics, **27**, pp. 857-874.

**Greenacre, M. J.** (1984) *Theory and Applications of Correspondence Analysis*, Academic Press.

**Hand, D. J.** (1981) *Discrimination and Classification*, John Wiley & Sons.

**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.

**Jain, A. K. and Dubes, R. C.** (1988) *Algorithms for Clustering Data*, Prentice Hall.

**Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P.** (1983) *Optimisation by Simulated Annealing*, Science, **220**(4598), pp.671-680.

**Kodratoff, Y. and Tecuci, G.** (1988) *Learning Based on Conceptual Distance*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **10**(6), pp. 897-909.

**MacQueen, J. B.** (1967) *Some Methods for Classification and Analysis of Multivariate Observations*, In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297.

**Michalski, R. S. and Stepp, R. E.** (1983) *Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **5**(4), pp. 396-410.

**Murtagh, F.** (1992*) Comments on "Parallel Algorithms for Hierarchical Clustering and Cluster Validity*", IEEE Transactions on Pattern Analysis and Machine Intelligence, **14**(10), pp. 1056-1057.

**Murthy, C. A. and Chowdhury, N.** (1996) *In Search of Optimal Clusters Using Genetic Algorithms*, Pattern Recognition Letters, **17**, pp. 825-832.

**Ralambondrainy, H.** (1995) *A Conceptual Version of the k-Means Algorithm*, Pattern Recognition Letters, **16**, pp. 1147-1157.

**Rose, K., Gurewitz, E. and Fox, G.** (1990) A *Deterministic Annealing Approach to Clustering*, Pattern Recognition Letters, **11**, pp. 589-594.

**Ruspini, E. R.** (1969) *A New Approach to Clustering*, Information Control, **19**, pp. 22-32.

**Ruspini, E. R.** (1973) *New Experimental Results in Fuzzy Clustering*, Information Sciences, **6**, pp. 273-284.

**Selim, S. Z. and Ismail, M. A.** (1984) *K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **6**(1), pp. 81-87.

**Shafer, J., Agrawal, R. and Metha, M.** (1996) *SPRINT: A Scalable Parallel Classifier for Data Mining*, In Proceedings of the 22nd VLDB Conference, Bombay, India, pp. 544-555.

## Appendix

The theorem in Section 4.3 can be proved as follows ($A_j$ stands for DOM($A_j$) here):

Let $f_r(A_j = c_{k,j} | \boldsymbol{X}) = \frac{n_{c_{k,j}}}{n}$ be the relative frequency of category $c_{k,j}$ of attribute $A_j$, where $n$ is the total number of objects in $\boldsymbol{X}$ and $n_{c_{k,j}}$ the number of objects having category $c_{k,j}$. For the dissimilarity measure $d(x,y) = \sum_{j=1}^{m} \delta(x_j, y_j)$, we write

$$\begin{aligned}
&\sum_{i=1}^{n} d(X_i, Q) \\
&\quad = \sum_{i=1}^{n}\sum_{j=1}^{m} \delta(x_{i,j}, q_j) \\
&\quad = \sum_{j=1}^{m}(\sum_{i=1}^{n} \delta(x_{i,j}, q_j)) \\
&\quad = \sum_{j=1}^{n} n(1 - \frac{n_{q_j}}{n}) \\
&\quad = \sum_{j=1}^{m} n(1 - f_r(A_j = q_j | \boldsymbol{X})
\end{aligned}$$

Because $n(1 - f_r(A_j = q_j | \boldsymbol{X})) \geq 0$ for $1 \leq j \leq m$, $\sum_{i=1}^{n} d(X_i, Q)$ is minimised iff every $n(1 - f_r(A_j = q_j | \boldsymbol{X}))$ is minimal. Thus, $f_r(A_j = q_j | \boldsymbol{X})$ must be maximal.

For the dissimilarity measure $d_{\chi^2}(x,y) = \sum_{j=1}^{m} \frac{(n_{x_j} + n_{y_j})}{n_{x_j} n_{y_j}} \delta(x_j, y_j)$, we write

$$\begin{aligned}
&\sum_{i=1}^{n} d_{\chi^2}(X_i, Q) \\
&\quad = \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{(n_{x_{i,j}} + n_{q_j})}{n_{x_{i,j}} n_{q_j}} \delta(x_{i,j}, q_j) \\
&\quad = \sum_{j=1}^{m}\sum_{i=1}^{n} (\frac{1}{n_{q_j}} + \frac{1}{n_{x_{i,j}}}) \delta(x_{i,j}, q_j) \\
&\quad = \sum_{j=1}^{m}\sum_{i=1}^{n} \frac{1}{n_{q_j}} \delta(x_{i,j}, q_j) + \sum_{j=1}^{m}\sum_{i=1}^{n} \frac{1}{n_{x_{i,j}}} \delta(x_{i,j}, q_j)
\end{aligned}$$

Now we have

$$\begin{aligned}
&\sum_{i=1}^{n} \frac{1}{n_{x_{i,j}}} \delta(x_{i,j}, q_j) \\
&\quad = \sum_{k=1}^{n_{cj}} \frac{n}{n_{c_{k,j}}} f_r(A_j = c_{k,j} | \boldsymbol{X}) - \frac{n}{n_{q_j}} f_r(A_j = q_j | \boldsymbol{X}) \\
&\quad = n_{c_j} - 1
\end{aligned}$$

where $n_{c_j}$ is the number of categories in $A_j$ and $n_{c_{k,j}}$ the number of objects having category $c_{k,j}$. Consequently, we get

$$\sum_{i=1}^{n} d_{\chi^2}(X_i, Q) = \sum_{j=1}^{m} \frac{n}{n_{q_j}} (1 - f_r(A_j = c_{k,j} | \boldsymbol{X})) + \sum_{j=1}^{m} (n_{c_j} - 1)$$

Because $\frac{n}{n_{q_j}}(1 - f_r(A_j = q_j | \boldsymbol{X})) \geq 0$ and $\sum_{j=1}^{m} (n_{c_j} - 1)$ is a constant for a given $\boldsymbol{X}$, $\sum_{i=1}^{n} d_{\chi^2}(X_i, Q)$ is minimised iff every $\frac{n}{n_{q_j}}(1 - f_r(A_j = q_j | \boldsymbol{X}))$ is minimal. Thus, $f_r(A_j = q_j | \boldsymbol{X})$ must be maximal .