Automated Variable Weighting in *k*-Means Type Clustering

Joshua Zhexue Huang, Michael K. Ng, Hongqiang Rong, and Zichen Li

Abstract—This paper proposes a k-means type clustering algorithm that can automatically calculate variable weights. A new step is introduced to the k-means clustering process to iteratively update variable weights based on the current partition of data and a formula for weight calculation is proposed. The convergency theorem of the new clustering process is given. The variable weights produced by the algorithm measure the importance of variables in clustering and can be used in variable selection in data mining applications where large and complex real data are often involved. Experimental results on both synthetic and real data have shown that the new algorithm outperformed the standard k-means type algorithms in recovering clusters in data.

Index Terms—Clustering, data mining, mining methods and algorithms, feature evaluation and selection.

1 Introduction

Clustering is a process of partitioning 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. The k-means type clustering algorithms [1], [2] are widely used in real world applications such as marketing research [3] and data mining to cluster very large data sets due to their efficiency and ability to handle numeric and categorical variables that are ubiquitous in real databases.

A major problem of using the k-means type algorithms in data mining is selection of variables. The k-means type algorithms cannot select variables automatically because they treat all variables equally in the clustering process. In practice, selection of variables for a clustering problem such as customer segmentation is often made based on understanding of the business problem and data to be used. Tens or hundreds of variables are usually extracted or derived from the database in the initial selection which form a very high-dimensional space. It is well-known that an interesting clustering structure usually occurs in a subspace defined by a subset of the initially selected variables. To find the clustering structure, it is important to identify the subset of variables.

In this paper, we propose a new k-means type algorithm called W-k-means that can automatically weight variables based on the importance of the variables in clustering. W-k-means adds a new step to the basic k-means algorithm to update the variable weights based on the current partition of data. We present a weight calculation formula that minimizes the cost function of clustering given a fixed partition of data. The convergency theorem of the new clustering process is proven.

We present a series of experiments conducted on both synthetic and real data. The results have shown that the new algorithm outperformed the standard k-means type algorithms in recovering clusters in data.

The variable weights produced by W-k-means measure the importance of variables in clustering. The small weights reduce or eliminate the effect of insignificant (or noisy) variables. The weights can be used in variable selection in data mining applications where large and complex real data are often involved.

The rest of this paper is organized as follows: Section 2 is a brief survey of related work on variable weighting and selection. Section 3 introduces the basic k-means algorithm. Section 4 presents the W-k-means algorithm. Experiments on both synthetic and real data are presented in Section 5. We conclude this paper in Section 6.

2 Related Work

Variable selection and weighting have been important research topics in cluster analysis [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13].

Desarbo et al. [4] introduced the first method for variable weighting in k-means clustering in the SYNCLUS algorithm. The SYNCLUS process is divided into two stages. Starting from an initial set of weights, SYNCLUS first uses the k-means clustering to partition data into k clusters. It then estimates a new set of optimal weights by optimizing a weighted mean-square, stress-like cost function. The two stages iterate until they converge to an optimal set of weights. The algorithm is time-consuming computationally [3], so it cannot process large data sets.

De Soete [5], [6] proposed a method to find optimal variable weights for ultrametric and additive tree fitting. This method was used in the hierarchical clustering methods to solve the variable weighting problem. Since

- *J.Z. Huang is with the E-Business Technology Institute, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: jhuang@eti.hku.hk.*
- *M.K. Ng is with the Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: mng@maths.hku.hk.*
- *H. Rong is with the Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: hqrong@cs.hku.hk.*
- *Z. Li is with the Department of Computer Science and Technology, Henan Polytechnic University, Jiaozuo City, Henan Province, China, 454003. E-mail: lizc@hpu.edu.cn.*

Manuscript received 2 Nov. 2003; revised 9 Aug. 2004; accepted 22 Sept. 2004; published online 11 Mar. 2005.
Recommended for acceptance by K. Yamamoto.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0349-1103.

the hierarchical clustering methods are computationally complex, De Soete's method cannot handle large data sets. Makarenkov and Legendre [11] extended De Soete's method to optimal variable weighting for the k-means clustering. The basic idea is to assign each variable a weight w_i in calculating the distance between two objects and find the optimal weights by optimizing the cost function $L_p(w_1, w_2, \ldots, w_p) = \sum_{k=1}^{K}(\sum_{i,j=1}^{n_k} d_{ij}^2/n_k)$. Here, K is the number of clusters, n_k is the number of objects in the kth cluster, and d_{ij} is the distance between the ith and the jth objects. The Polak-Ribiere optimization procedure is used in minimization, which makes the algorithm very slow. The simulation results in [11] show that the method is effective in identifying important variables, but not scalable to large data sets.

Modha and Spangler [13] very recently published a new method for variable weighting in k-means clustering. This method aims to optimize variable weights in order to obtain the best clustering by minimizing the ratio of the average within-cluster distortion over the average between-cluster distortion, referred to as the generalized Fisher ratio Q. To find the minimal Q, a set of feasible weight groups were defined. For each weight group, the k-means algorithm was used to generate a data partition and Q was calculated from the partition. The final clustering was determined as the partition having the minimal Q. This method of finding optimal weights from a predefined set of variable weights may not guarantee that the predefined set of weights would contain the optimal weights. Besides, it is also a practical problem to decide the predefined set of weights for high-dimensional data.

Friedman and Meulman [12] recently published a method to cluster objects on subsets of attributes. Instead of assigning a weight to each variable for the entire data set, their approach is to compute a weight for each variable in each cluster. As such, $p * L$ weights are computed in the optimization process, where p is the total number of variables and L is the number of clusters. Since the objective function is a complicated highly nonconvex function, direct method to minimize it has not been found. An approximation method is used to find clusters on different subsets of variables by combining conventional distance-based clustering methods with a particular distance measure. Friedman and Meulman's work is related to the problem of subspace clustering [14]. Scalability is a concern because their approximation method is based on the hierarchical clustering methods.

The fuzzy k-means type clustering algorithms [15], [16] use the k-means clustering process to calculate the weights of clusters for each object that can determine which cluster(s) the object should be assigned to. In this paper, we adopt a similar approach that can be used to calculate the weights for variables. In this way, variable weights can be automatically calculated within the clustering process without sacrificing the efficiency of the algorithm. The weights are optimized from the entire weight space rather than from a limited number of candidates as in Modha and Spangler's approach [13].

3 The *k*-Means Type Algorithms

Let $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$ be a set of n objects. Object $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,m})$ is characterized by a set of m variables (attributes). The k-means type algorithms [2], [17] search for a partition of $\boldsymbol{X}$ into k clusters that minimizes the objective function P with unknown variables U and Z as follows:

$$P(U, Z) = \sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{i,l}\, d(x_{i,j}, z_{l,j}) \tag{1}$$

subject to

$$\sum_{l=1}^{k} u_{i,l} = 1, \quad 1 \le i \le n, \tag{2}$$

where

- U is an $n \times k$ partition matrix, $u_{i,l}$ is a binary variable, and $u_{ij} = 1$ indicates that object i is allocated to cluster l;
- $Z = \{Z_1, Z_2, \ldots, Z_k\}$ is a set of k vectors representing the centroids of the k clusters;
- $d(x_{i,j}, z_{l,j})$ is a distance or dissimilarity measure between object i and the centroid of cluster l on the jth variable. If the variable is numeric, then

$$d(x_{i,j}, z_{l,j}) = (x_{i,j} - z_{l,j})^2. \tag{3}$$

If the variable is categorical, then

$$d(x_{i,j}, z_{l,j}) = \begin{cases} 0 & (x_{i,j} = z_{l,j}) \\ 1 & (x_{i,j} \ne z_{l,j}). \end{cases} \tag{4}$$

The algorithm is called k-modes if all variables in the data are categorical or k-prototypes if the data contains both numeric and categorical variables [1].

The above optimization problem can be solved by iteratively solving the following two minimization problems:

1. Problem P_1: Fix $Z = \hat{Z}$ and solve the reduced problem $P(U, \hat{Z})$,
2. Problem P_2: Fix $U = \hat{U}$ and solve the reduced problem $P(\hat{U}, Z)$.

Problem P_1 is solved by

$$\begin{cases} u_{i,l} = 1 & \text{if} \quad \sum_{j=1}^{m} d(x_{i,j}, z_{l,j}) \le \sum_{j=1}^{m} d(x_{i,j}, z_{t,j}) \quad \text{for } 1 \le t \le k \\ u_{i,t} = 0 & \text{for} \quad t \ne l \end{cases} \tag{5}$$

and problem P_2 is solved for the numeric variables by

$$z_{l,j} = \frac{\sum_{i=1}^{n} u_{i,l}\, x_{i,j}}{\sum_{i=1}^{n} u_{i,l}} \quad \text{for } 1 \le l \le k \text{ and } 1 \le j \le m. \tag{6}$$

If the variable is categorical, then

$$z_{l,j} = a_j^r, \tag{7}$$

where a_j^r is the mode of the variable values in cluster l, [1].

The basic algorithm to minimize the objective function P is given in [1], [15], [18].

One of the drawbacks of the k-means type algorithms is that they treat all variables equally in deciding the cluster memberships of objects in (5). This is not desirable in many applications such as data mining where data often contains

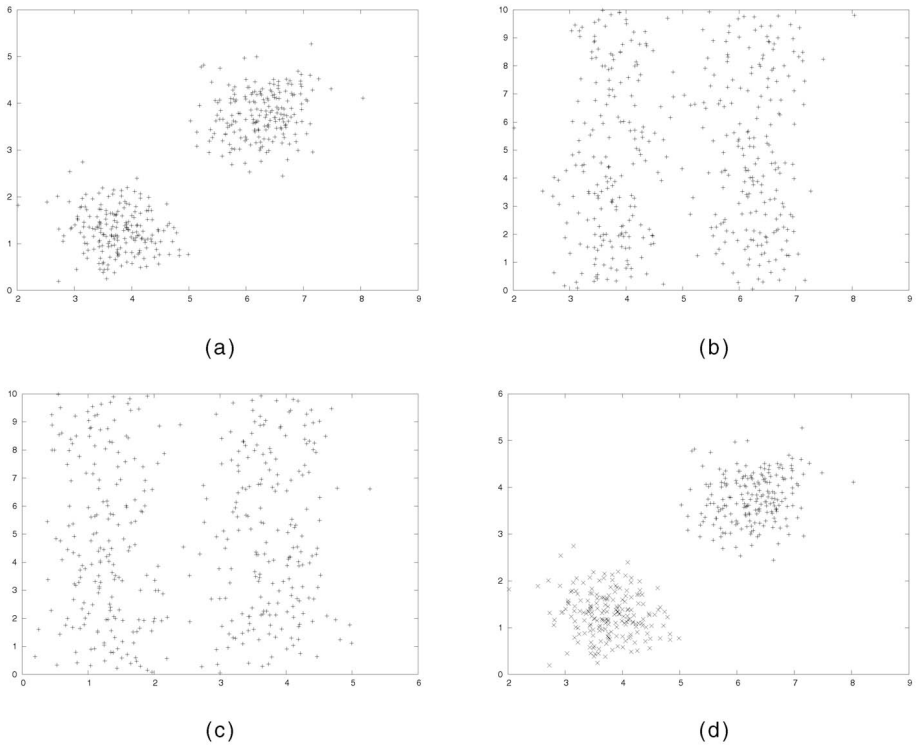


Fig. 1. Clustering with noise data. (a) Two clusters in the subspace of x_1, x_2. (b) Plot of the subspace of x_1, x_3. (c) Plot of the subspace of x_2, x_3. (d) Two discovered clusters in the subspace of x_1, x_2.

a large number of diverse variables. A cluster structure in a given data set is often confined to a subset of variables rather than the entire variable set. Inclusion of other variables can only obscure the discovery of the cluster structure by a clustering algorithm.

Fig. 1 shows the effect of a noise variable to the clustering results of the k-means algorithm. The data set X has three variables x_1, x_2, x_3. Two normally distributed clusters are found in (x_1, x_2) (see Fig. 1a). x_3 is a random variable with a uniform distribution. No cluster structure can be found in (x_1, x_3) and (x_2, x_3) (see Figs. 1b and 1c). If we apply the k-means algorithm to X, the two clusters in (x_1, x_2) may not be discovered because of the noise variable x_3. However, if we assign weights 0.47, 0.40, and 0.13 to variables x_1, x_2, and x_3, respectively, in the distance function (5), the k-means algorithm will recover the two clusters as plotted in Fig. 1d. Real data sets in data mining often have variables in the hundreds and records in the hundreds of thousands, such as the customer data sets in large banks. How to calculate the variable weights automatically in the clustering process to distinguish good variables like x_1, x_2 from noise variables like x_3 is a great challenge. In the next section, we present the W-k-means algorithm that can automatically calculate the variable weights.

4 The W-*k*-Means Type Algorithms

Let $W = [w_1, w_2, \ldots, w_m]$ be the weights for m variables and β be a parameter for attribute weight w_j, we modify (1) as

$$P(U, Z, W) = \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} u_{i,l} w_j^{\beta} d(x_{i,j}, z_{l,j}) \tag{8}$$

subject to

$$\begin{cases} \sum_{l=1}^{k} u_{i,l} = 1, & 1 \le i \le n \\ u_{i,l} \in \{0, 1\}, & 1 \le i \le n, \quad 1 \le l \le k \\ \sum_{j=1}^{m} w_j = 1, & 0 \le w_j \le 1. \end{cases} \tag{9}$$

Similar to solving (1), we can minimize (8) by iteratively solving the following three minimization problems:

1. Problem P_1: Fix $Z = \hat{Z}$ and $W = \hat{W}$, solve the reduced problem $P(U, \hat{Z}, \hat{W})$;
2. Problem P_2: Fix $U = \hat{U}$ and $W = \hat{W}$, solve the reduced problem $P(\hat{U}, Z, \hat{W})$;
3. Problem P_3: Fix $U = \hat{U}$ and $Z = \hat{Z}$, solve the reduced problem $P(\hat{U}, \hat{Z}, W)$.

Problem P_1 is solved by

$$\begin{cases} u_{i,l} = 1 \quad \text{if} \quad \sum_{j=1}^{m} w_j^{\beta} d(x_{i,j}, z_{l,j}) \le \sum_{j=1}^{m} w_j^{\beta} d(x_{i,j}, z_{t,j}) \\ \qquad\qquad\qquad\qquad \text{for } 1 \le t \le k \\ u_{i,t} = 0 \quad \text{for} \quad t \ne l \end{cases} \tag{10}$$

and problem P_2 is solved in (6) and (7). The solution to problem P_3 is given in Theorem 1.

Theorem 1. *Let $U = \hat{U}$ and $Z = \hat{Z}$ be fixed.*

1. *When $\beta > 1$ or $\beta \le 0$, $P(\hat{U}, \hat{Z}, W)$ is minimized iff*

$$\hat{w}_j = \begin{cases} 0 & \text{if } D_j = 0 \\ \dfrac{1}{\sum_{t=1}^{h} \left[\frac{D_j}{D_t}\right]^{\frac{1}{\beta-1}}} & \text{if } D_j \ne 0, \end{cases} \tag{11}$$

where

$$D_j = \sum_{l=1}^{k}\sum_{i=1}^{n} \hat{u}_{i,l} d(x_{i,j}, z_{l,j}) \tag{12}$$

and h is the number of variables where $D_j \neq 0$.

2. *When $\beta = 1$, $P(\hat{U}, \hat{Z}, W)$ is minimized iff*

$$\hat{w}_{j6\prime} = 1 \quad \text{and} \quad \hat{w}_j = 0, \quad j \neq j',$$

where $D_{j'} \leq D_j$ for all j.

Proof.

1. We rewrite problem P_3 as

$$\begin{aligned} P(\hat{U}, \hat{Z}, W) &= \sum_{j=1}^{m} w_j^{\beta} \sum_{l=1}^{k}\sum_{i=1}^{n} \hat{u}_{i,l} d(x_{i,j}, z_{l,j}) \\ &= \sum_{j=1}^{m} w_j^{\beta} D_j, \end{aligned} \tag{13}$$

where D_js are m constants for fixed $\hat{U}$ and $\hat{Z}$.

If $D_j = 0$, according to (12), the jth variable has a unique value in each cluster. This represents a degenerate solution, so we assign $\hat{w}_j = 0$ to all variables where $D_j = 0$.

For the $h(\leq m)$ variables where $D_j \neq 0$, we consider the relaxed minimization of P_3 via a Lagrange multiplier obtained by ignoring the constraint $\sum_{j=1}^{m} w_j = 1$. Let α be the multiplier and $\Psi(W, \alpha)$ be the Lagrangian

$$\Psi(W, \alpha) = \sum_{j=1}^{h} w_j^{\beta} D_j + \alpha\Big(\sum_{j=1}^{h} w_j - 1\Big). \tag{14}$$

If $(\hat{W}, \hat{\alpha})$ is to minimize $\Psi(W, \alpha)$, its gradient in both sets of variables must vanish. Thus,

$$\frac{\partial \Psi(\hat{W}, \hat{\alpha})}{\partial \hat{w}_j} = \beta \hat{w}_j^{\beta-1} D_j + \hat{\alpha} = 0 \quad \text{for } 1 \leq j \leq h, \tag{15}$$

$$\frac{\partial \Psi(\hat{W}, \hat{\alpha})}{\partial \hat{\alpha}} = \sum_{j}^{h} \hat{w}_j - 1 = 0. \tag{16}$$

From (15), we obtain

$$\hat{w}_j = \Big(\frac{-\hat{\alpha}}{\beta D_j}\Big)^{\frac{1}{\beta-1}} \quad \text{for } 1 \leq j \leq h. \tag{17}$$

Substituting (17) into (16), we have

$$\sum_{t=1}^{h} \Big(\frac{-\hat{\alpha}}{\beta D_t}\Big)^{\frac{1}{\beta-1}} = 1. \tag{18}$$

From (18), we derive

$$(-\hat{\alpha})^{\frac{-1}{\beta-1}} = 1 \Big/ \Big[\sum_{t=1}^{h} \Big(\frac{1}{\beta D_t}\Big)^{\frac{1}{\beta-1}}\Big]. \tag{19}$$

Substituting (19) into (17), we obtain

$$\hat{w}_j = \frac{1}{\sum_{t=1}^{h} \Big[\frac{D_j}{D_t}\Big]^{\frac{1}{\beta-1}}}. \tag{20}$$

2. It is clear that, when $w_{j'} = 1$, the corresponding objective function value is equal to $D_{j'}$. Let $(w_1, w_2, \cdots, w_m)$ be a feasible solution of the optimization problem. Using the fact that $\sum_{j=1}^{m} w_j = 1$ and the feasible solution space ($\sum_{j=1}^{m} w_j = 1$ and $0 \leq w_j \leq 1$ for $1 \leq j \leq n$) is convex, it is straightforward to show that

$$D_{j'} \leq \sum_{j=1}^{m} w_j D_j.$$

Therefore, when we set $w_{j'} = 1$, the optimal solution can be determined. □

The algorithm to solve (8) is given as follows:

Algorithm—(The W-k-Means type algorithms)

Step1. Randomly choose an initial $Z^0 = \{Z_1, Z_2, \ldots, Z_k\}$ and randomly generate a set of initial weights $W^0 = [w_1^0, w_2^0, \ldots, w_m^0]$ ($\sum_{j=1}^{m} w_j = 1$). Determine U^0 such that $P(U^0, Z^0, W^0)$ is minimized. Set $t = 0$;

Step2. Let $\hat{Z} = Z^t$ and $\hat{W} = W^t$, solve problem $P(U, \hat{Z}, \hat{W})$ to obtain U^{t+1}. If $P(U^{t+1}, \hat{Z}, \hat{W}) = P(U^t, \hat{Z}, \hat{W})$, output $(U^t, \hat{Z}, \hat{W})$ and stop; otherwise, go to Step 3;

Step3. Let $\hat{U} = U^{t+1}$ and $\hat{W} = W^t$, solve problem $P(\hat{U}, Z, \hat{W})$ to obtain Z^{t+1}. If $P(\hat{U}, Z^{t+1}, \hat{W}) = P(\hat{U}, Z^t, \hat{W})$, output $(\hat{U}, Z^t, \hat{W})$ and stop; otherwise, go to Step 4;

Step4. Let $\hat{U} = U^{t+1}$ and $\hat{Z} = Z^{t+1}$, solve problem $P(\hat{U}, \hat{Z}, W)$ to obtain W^{t+1}. If $P(\hat{U}, \hat{Z}, W^{t+1}) = P(\hat{U}, \hat{Z}, W^t)$, output $(\hat{U}, \hat{Z}, W^t)$ and stop; otherwise, set $t = t + 1$ and go to Step 2.

Theorem 2. *The above algorithm converges to a local minimal solution in a finite number of iterations.*

Proof. We first note that there are only a finite number of possible partitions U. We then show that each possible partition U appears at most once by the algorithm. Assume that $U^{t_1} = U^{t_2}$, where $t_1 \neq t_2$. We note that, given U^t, we can compute the minimizer Z^t which is independent of W^t. For U^{t_1} and U^{t_2}, we have the minimizers Z^{t_1} and Z^{t_2}, respectively. It is clear that $Z^{t_1} = Z^{t_2}$ since $U^{t_1} = U^{t_2}$. Using U^{t_1} and Z^{t_1}, and U^{t_2} and Z^{t_2}, we can compute the minimizers W^{t_1} and W^{t_2}, respectively, (Step 4) according to Theorem 1. Again, $W^{t_1} = W^{t_2}$. Therefore, we have

$$P(U^{t_1}, Z^{t_1}, W^{t_1}) = P(U^{t_2}, Z^{t_2}, W^{t_2}).$$

However, the sequence $P(\cdot, \cdot, \cdot)$ generated by the algorithm is strictly decreasing. Hence, the result follows.

Since the W-k-means algorithm is an extension to the k-means algorithm by adding a new step to calculate the variable weights in the iterative process, it does not seriously affect the scalability of the k-means type algorithms in clustering large data; therefore, it is suitable for data mining applications.

The computational complexity of the algorithm is $O(tmnk)$, where t is the total number of iterations required for performing Step2, Step3, and Step4, k is the number of clusters, m is the number of attributes, and n is the number of objects. □

4.1 Variable Weighting

Given a data partition, the principal for variable weighting is to assign a larger weight to a variable that has a smaller sum of the within cluster distances and a smaller one to a

TABLE 1
Centroids and Standard Deviations of Clusters in Different Variables

Cluster	Cluster centroids	Standard deviations	No. of points
1	(0.547,0.728,0.424,0.492,0.561)	(0.054,0.044,0.071,0.288,0.302)	100
2	(0.299,0.585,0.318,0.555,0.455)	(0.061,0.044,0.069,0.269,0.274)	100
3	(0.422,0.452,0.636,0.520,0.536)	(0.055,0.050,0.075,0.263,0.274)	100

variable that has a larger sum of the within cluster distances. The sum of the within cluster distances for variable x_j is given by (12) in Theorem 1 and the weight $\hat{w}_j$ for x_j is calculated by (11). However, the real weight w_j^β to variable x_j is also dependent on the value of β (see the objective function (8)). Based on the above principal, we can analyze what values we can choose for β.

When $\beta = 0$, (8) is equal to (1), regardless of $\hat{w}_j$.

When $\beta = 1$, w_j is equal to 1 for the smallest value of D_j. The other weights are equal to 0. Although the objective function is minimized, the clustering is made by the selection of one variable. It may not be desirable for high-dimensional clustering problems.

When $0 < \beta < 1$, the larger D_j, the larger w_j, so does w_j^β. This is against the variable weighting principal, so we cannot choose $0 < \beta < 1$.

When $\beta > 1$, the larger D_j, the smaller w_j, and the smaller w_j^β. The effect of variable x_j with large D_j is reduced.

When $\beta < 0$, the larger D_j, the larger w_j. However, w_j^β becomes smaller and has less weighting to the variable in the distance calculation because of negative β.

From the above analysis, we can choose $\beta < 0$ or $\beta > 1$ in the W-k-means algorithm.

5 Experiments

In this section, we use experimental results to demonstrate the clustering performance of the W-k-means algorithm in discovering clusters and identifying insignificant (or noisy) variables from given data sets. Both synthetic data and real data were used in these experiments. In clustering real data with mixed numeric and categorical values, the k-prototypes algorithm [1] and the W-k-prototypes algorithm were used.

5.1 Experiment on Synthetic Data

Synthetic data is often used to validate a clustering algorithm [19]. In this experiment, we used a constructed synthetic data set with known normally distributed clusters and noise variables to verify the performance of the W-k-means algorithm in discovering the clusters inherent in a subspace of the data domain and the properties of the algorithm in identifying the noise variables in the data set.

5.1.1 Synthetic Data Set

The synthetic data set contains five variables and 300 records that are divided into three clusters normally distributed in the first three variables. Each cluster has 100 points. The last two variables represent uniformly distributed noise points. The centroids and standard deviations of the three clusters are given in Table 1. The centroids of the first three variables are well separated and the standard deviations are small. The centroids of the two noise variables are very close and the standard deviations are much larger than those of the first three variables.

Fig. 2 plots the 300 points in different two-dimensional subspaces. In the figure, x_0, x_1, x_2 represent the three variables that contain three normally distributed clusters, while x_3, x_4 are the two noise variables that are uniformly distributed in the unit square. Because the three clusters are not identifiable in a subspace with a noise variable, the noise variables introduce difficulties to the discovery of the three clusters embedded in the data set. With this noise data set, we could demonstrate that the W-k-means algorithm was able to recover the three clusters and identify the two noise variables.

5.1.2 Evaluation Method

Since the cluster labels of the data points in the synthetic data set were known, the Rand Index was used to evaluate the performance of the clustering algorithm [20]. Let $C = \{C_1, C_2, C_3\}$ be the set of three clusters in the data set and $C' = \{C'_1, C'_2, C'_3\}$ the set of three clusters generated by the clustering algorithm. Given a pair of points (X_i, X_j) in the data set, we refer to it as

1. SS if both points belong to the same cluster in C and the same cluster in C',
2. DD if both points belong to two different clusters in C and two different clusters in C',
3. SD if the two points belong to the same cluster in C and different clusters in C',
4. DS if the two points belong to two different clusters in C and to the same cluster in C'.

Let a, b, c, and d be the number of SS, SD, DS, and DD pairs of points, respectively. Then, $a + b + c + d = M$, where $M = N(N-1)/2$ is the total number of possible point pairs in the data set and N is the number of points. The Rand Index is calculated as

$$R = \frac{a+d}{M}. \tag{21}$$

The Rand Index measures the fraction of the total number of pairs that are either SS or DD. The larger the value, the higher the agreement between C and C'.

In the meantime, we also calculated the clustering accuracy as

$$r = 100\frac{\sum_{i=1}^{k} a_i}{N}, \tag{22}$$

where a_i is the number of points in C_i that were clustered to C'_i and N is the number of points in the data set. r is the percentage of the points that were correctly recovered in a clustering result.

5.1.3 Results

It is well known that the standard k-means clustering process produces a local optimal solution. The final result depends on the initial cluster centroids. In the W-k-means clustering

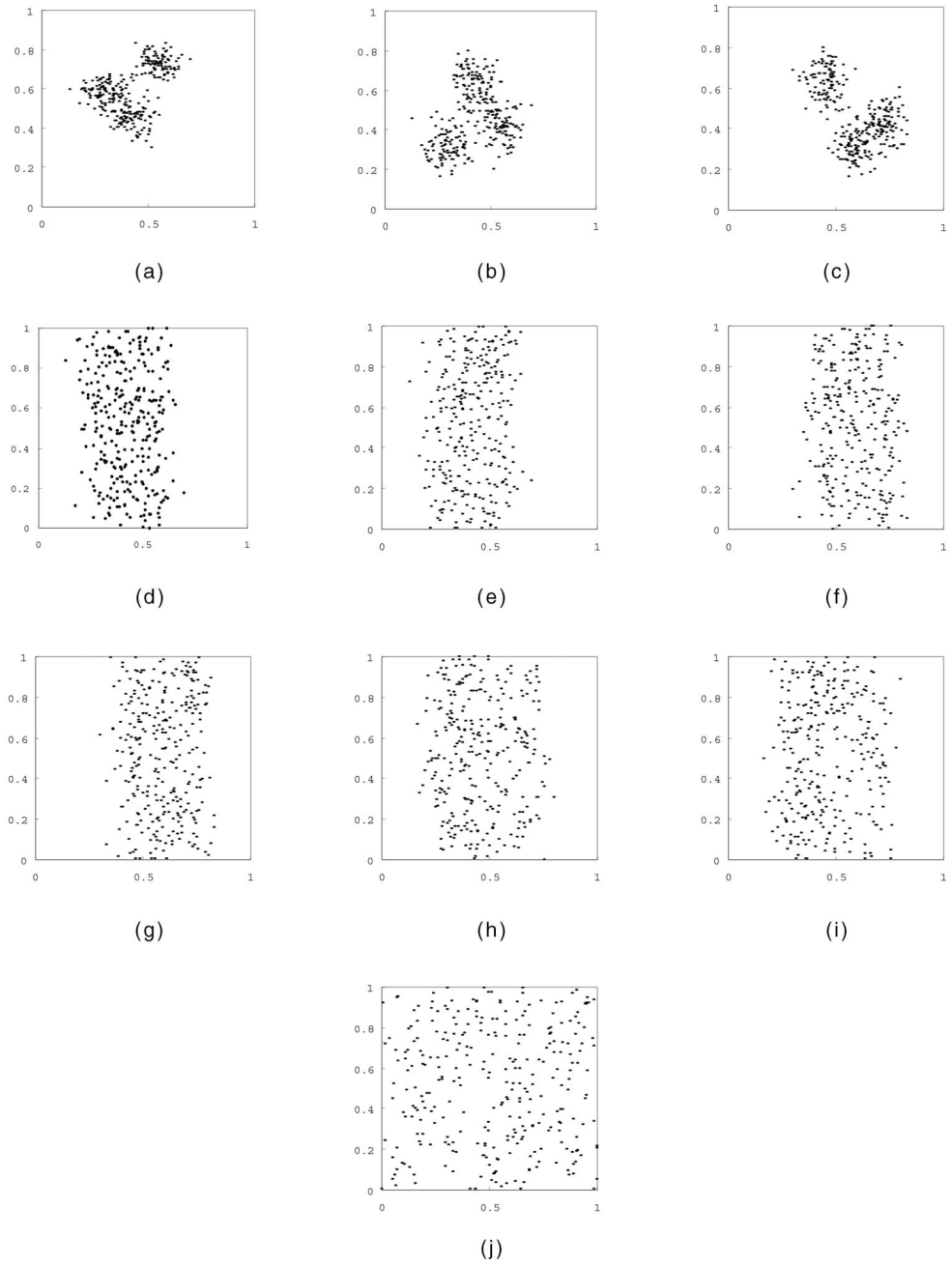


Fig. 2. Synthetic data set with three normally distributed clusters in the three-dimensional subspace of x_0, x_1, x_2 and two noise variables x_3, x_4. (a) The subspace of x_0, x_1. (b) The subspace of x_0, x_2. (c) The subspace of x_1, x_2. (d) The subspace of x_0, x_3. (e) The subspace of x_0, x_4. (f) The subspace of x_1, x_3. (g) The subspace of x_1, x_4. (h) The subspace of x_2, x_3. (i) The subspace of x_2, x_4. (j) The subspace of x_3, x_4.

process, the initial weights also affect the final result of clustering. To test the performance of the W-k-means algorithm, we first fixed the initial cluster centroids and ran the W-k-means algorithm on the synthetic data with different sets of initial weights. Then, we ran the W-k-means algorithm with different sets of initial cluster centroids and initial weights. Here, we set $\beta = 8$ in these experiments. We compared the W-k-means results with the results from the standard k-means algorithm and the k-means algorithm with the weighted distance function, i.e., a set of weights were used in the distance function, but the weights did not change during the k-means clustering process.

Given a fixed set of initial cluster centroids, the Monte Carlo sampling method [21] was used to generate a set of random initial weights. First, we used the weights in the distance function to run the k-means algorithm and generated a clustering result. Second, we used the weights as the initial weights to run the W-k-means algorithm in the following way. Given the initial weights, the k-means algorithm was run with the weighted distance function until it converged. Then, a new set of weights were calculated using (11). With the new weights as the initial weights and the current cluster centroids as the initial centroids, the k-means algorithm restarted to produce another partition. This process repeated until it converged, i.e., the objective function (8) was minimized.

Fig. 3 shows a typical convergence curve of the W-k-means algorithm. The horizontal axis represents the number of iterations and the vertical axis represents the value of the

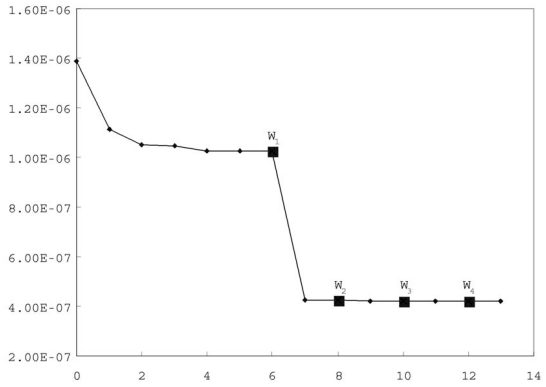


Fig. 3. Convergence curve of the W-k-means algorithm.

objective function (8). Each point on the curve represents a partition generated by one iteration of the k-means clustering process. Starting from a set of initial centroids and a set of initial weights, the algorithm first converged after six iterations. A new set of weights W_1 was computed. Using W_1 as the initial weights and the current cluster centroids, the k-means process restarted again. We can see that the objective function had a significant drop after the new weights W_1 were introduced. The k-means process converged again after two new iterations. Then, a set of new weights W_2 was computed. This process continued until the local minimal value of the objective function was reached. The final set of weights W_4 was obtained. We note that, in each step, the weighted distance function was fixed since the weights for the variables were fixed. Therefore, the corresponding weighted distance function space was well-defined at each step. We expected that the smaller the value of objective function value, the closer the data points under the weighted distance function space would be. We remark that by using similar arguments to those in the proof of Theorem 2, it can be shown that the above process is also convergent.

Table 2 lists 10 sets of randomly generated weights and the clustering results by the k-means algorithm. All clustering runs started with the same initial cluster centroids. Rand Index and clustering accuracy were used to evaluate the clustering results. We can see that only the second run produced a more accurate clustering result.

Using the 10 sets of randomly generated weights as the initial weights, we ran the W-k-means algorithm 10 times on the same data set, each starting with the same initial cluster centroids. The clustering results and the final weights are listed in Table 3. We can see that five runs achieved very accurate clustering results. Two results achieved 100 percent of recovery of the original clusters in the data set. These results show that the W-k-means algorithm was much superior to the k-means algorithm that used randomly generated weights to weight variables in the distance function.

From Table 3, we can also observe that the five good clustering results have very similar weights for the variables. The first three weights are much larger than the last two weights. These weights clearly separated the noise variables from the normal ones. In the good clustering results, because of the larger weighting values, the normal variables had much bigger impact on clustering than the noise variables. The weights of the five good clustering results are plotted in Fig. 4a and the corresponding randomly generated initial weights are plotted in Fig. 4b. The noise variables were not identifiable from the randomly generated weights, but could be easily identified from the final weights produced by the W-k-means algorithm.

Because the k-means type algorithms do not produce the global optimal solution, their clustering results depend on the initial cluster centroids and the initial weights. In practice, we need to run the W-k-means algorithm several times on the

TABLE 2
Ten Randomly Generated Weights and the Clustering Results by the k-Means Algorithm

Num	weight0	weight1	weight2	weight3	weight4	Rand Index	Accuracy
1	0.2185	0.2845	0.0809	0.2457	0.1704	0.7577	0.7467
2	0.2968	0.3261	0.0982	0.1740	0.1049	0.9738	0.9800
3	0.3637	0.1018	0.1642	0.2899	0.0804	0.7766	0.7967
4	0.2661	0.1881	0.0680	0.2413	0.2365	0.6738	0.6033
5	0.3841	0.1989	0.0841	0.1500	0.1829	0.7795	0.7933
6	0.3337	0.0510	0.0496	0.2351	0.3305	0.6174	0.5367
7	0.3377	0.0285	0.1386	0.0844	0.4109	0.5661	0.4367
8	0.2804	0.2525	0.0821	0.0172	0.3678	0.5663	0.4367
9	0.3569	0.1190	0.0654	0.4327	0.0261	0.5545	0.3767
10	0.2503	0.1202	0.1236	0.3400	0.1658	0.5545	0.3733

TABLE 3
Ten Final Weights and the Clustering Results by the W-k-Means Algorithm

Num	weight0	weight1	weight2	weight3	weight4	Rand Index	Accuracy
1	0.3021	0.4137	0.2268	0.0301	0.0273	1.0000	1.0000
2	0.3021	0.4137	0.2268	0.0301	0.0273	1.0000	1.0000
3	0.3078	0.4035	0.2310	0.0302	0.0274	0.9956	0.9967
4	0.3078	0.4035	0.2310	0.0302	0.0274	0.9956	0.9967
5	0.3078	0.4035	0.2310	0.0302	0.0274	0.9956	0.9967
6	0.3249	0.1362	0.1212	0.0814	0.3362	0.6204	0.5533
7	0.1204	0.0942	0.0850	0.0601	0.6403	0.5721	0.4500
8	0.1204	0.0942	0.0850	0.0601	0.6403	0.5721	0.4500
9	0.1092	0.0826	0.0772	0.6822	0.0487	0.5545	0.3767
10	0.1091	0.0826	0.0772	0.6824	0.0487	0.5545	0.3733

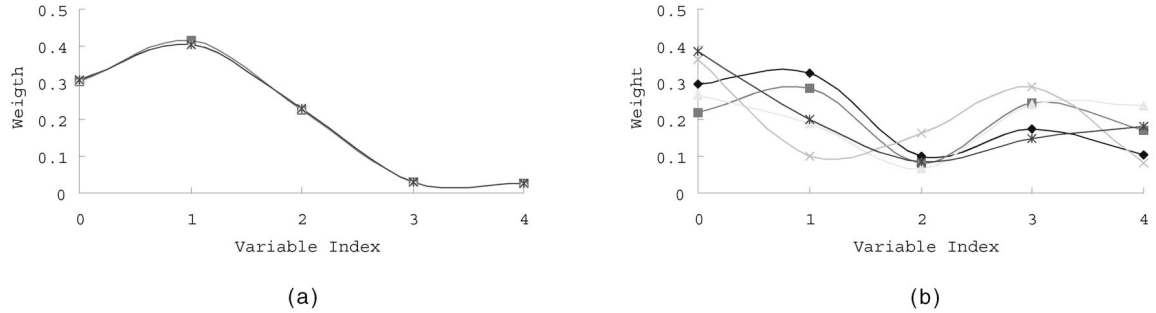


Fig. 4. (a) Plots of the final weights of the five good clustering results. (b) Plots of the initial weights of the five good clustering results.

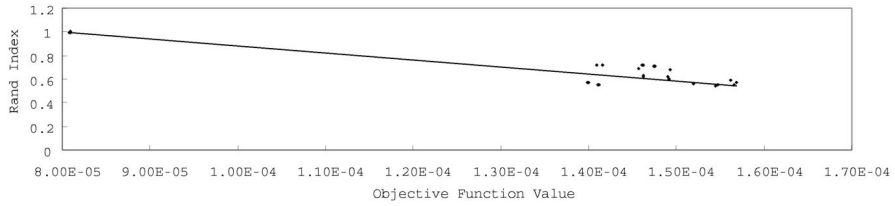


Fig. 5. Plot of the Rand Index against the values of the objective function (8) over 100 runs with different initial weights and the same initial cluster centroids.

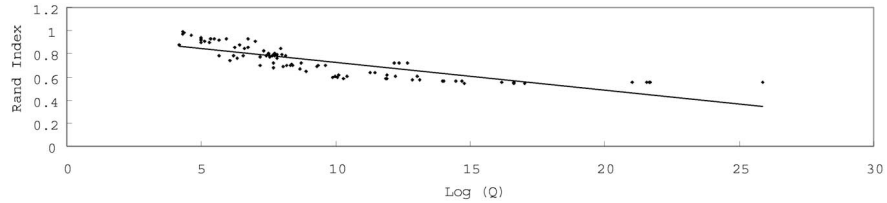


Fig. 6. Plot of the Rand Index against the values of the Log(Q) over 100 runs with different weights and the same initial cluster centroids.

same data set with different initial centroids and initial weights. To select the best result from several runs, we investigated the relationships between the Rand Index and the value of the objective function (8). Fig. 5 shows the plot of the Rand Index against the values of the objective function (8) over 100 runs with the same initial cluster centroids and different initial weights. From this figure, we can see the linear relationship between the Rand Index and the value of the objective function. The smaller the value of the objective function, the higher the Rand Index. This indicates that we can select the result with the minimal objective function value as the best result from several runs. A further interesting observation was that the upper left point in Fig. 5 was, in fact, a presentation of 67 clustering results that achieved the highest Rand Index (equal to 1.0). This result implies that the W-k-means algorithm has a high chance to produce a good clustering result in just a few runs with different initial centroids and initial weights. This nice property can save a lot of time in finding a good clustering from large data sets in data mining.

To compare with the method of using the generalized Fisher ratio Q to select the best result from several runs, we simulated the process in [13]. For each set of randomly generated weights by the Monte Carlo method, we ran the k-means algorithm with the weights to weight the variables in the distance function and calculated Q from the clustering result. We remark that the weights for the variables were fixed in the clustering process, which was different from the W-k-means algorithm. Fig. 6 shows the plot of the Rand Index against Log(Q) over 100 runs with the same initial cluster centroids and different initial weights. We can also observe the linear relationship between the Rand Index and Log(Q), which shows that Q is also an indicator to the best clustering result from several runs. However, we have found in Fig. 6 that the result with the minimal Log(Q) value did not give the largest Rand Index. Moreover, the upper left point in Fig. 6 only represents a few results that achieved the highest Rand Index. This implies that, if Q is used as an indicator, more runs are needed to obtain a good clustering.

According to the above results, we see that the W-k-means algorithm can improve the clustering results by updating the weights for the variables through minimization of the objective function (8). We recall in Fig. 4a that the important variables and the noise variables can be identified in the proposed clustering process. However, chance is very small for the randomly generated weights to identify noise variables. It is not feasible to predefine a very large number of weights for high-dimensional data, as used in [13]. Furthermore, using Q as an objective function to optimize, the fast, single automated algorithm has not yet been found [13]. Therefore, the W-k-means algorithm to optimize the objective function (8) has an advantage in processing very large data sets.

Table 4 shows the clustering results from 10 sets of randomly generated initial centroids. For each set of initial centroids, we first ran the standard k-means algorithm and calculated the clustering accuracy and the Rand Index. The 10 results are given in the second column in the table. The first value in a bracket is the clustering accuracy and the second value is the Rand Index. We then ran the k-means algorithm with randomly generated weights to weight the distance function. For each set of initial cluster centroids,

TABLE 4
Comparison of Results

Num	No Weights	Fixed Weights	Weights Changed
1	(0.4767, 0.5768)	(0.6764, 0.7317)	(0.8225, 0.8671)
2	(0.4833, 0.5796)	(0.6990, 0.7462)	(0.8453, 0.8809)
3	(0.5267, 0.6052)	(0.6871, 0.7429)	(0.7830, 0.8357)
4	(0.7200, 0.7652)	(0.6880, 0.7448)	(0.7893, 0.8403)
5	(0.7800, 0.7877)	(0.6938, 0.7445)	(0.8682, 0.8963)
6	(0.4764, 0.5780)	(0.6930, 0.7444)	(0.8337, 0.8713)
7	(0.7167, 0.7610)	(0.6960, 0.7479)	(0.7992, 0.8474)
8	(0.7767, 0.7884)	(0.6778, 0.7361)	(0.8003, 0.8478)
9	(0.7800, 0.7877)	(0.7040, 0.7515)	(0.8426, 0.8776)
10	(0.7200, 0.7589)	(0.6740, 0.7379)	(0.7810, 0.8341)
Average	(0.6457, 0.6989)	(0.6889, 0.7428)	(0.8156, 0.8599)

100 sets of random weights were tested. The average clustering accuracy and the average Rand Index value are shown in the third column. Finally, we ran the W-k-means algorithm on 100 sets of initial weights for each set of initial cluster centroids. The average clustering accuracy and the average Rand Index value are shown in the fourth column. The last row of the table gives the average results of the multiple runs with different settings. From the average clustering accuracy and the average Rand Index value, we can clear see the superiority of the W-k-means algorithm in clustering this data set.

5.2 Experiments on Real Data

5.2.1 Real Data Sets

Two real data sets, the Heart Diseases data and the Australian Credit Card data, were obtained from the UCI Machine Learning Repository. Both of them have numerical and categorical attributes. We used the variable weighting version of the W-k-prototypes algorithm in our experiments [1] because it is able to handle mixed numeric and categorical values.

The Australian credit card data set consists of 690 instances, each with six numerical and nine categorical attributes. The data sets are originally classified into two clusters: *approved* and *rejected*. Since some objects have missing values in seven attributes, only 653 instances were considered.

The Heart Diseases data set consists of 270 instances, each with five numerical and eight categorical attributes. The records are classified into two classes: *absence* and *presence*. In this data set, all instances were considered.

To study the effect of the initial cluster centers, we also randomly reordered the original data records and created 100 test data sets for each real data set. The accuracy of clustering results is calculated in the same way as the synthetical data experiments.

5.2.2 Results

Each test data set was clustered 20 times with different integer β values ranging from -10 to 10, excluding 1. The result of $\beta = 0$ was equivalent to the result of the k-means clustering without variable weighting. Here, we would like to investigate how to set the value of β to affect the clustering results in terms of the Rand Index and the clustering accuracy. Each β value resulted in 100 clustering results from one real data set. The accuracy of each clustering result was calculated.

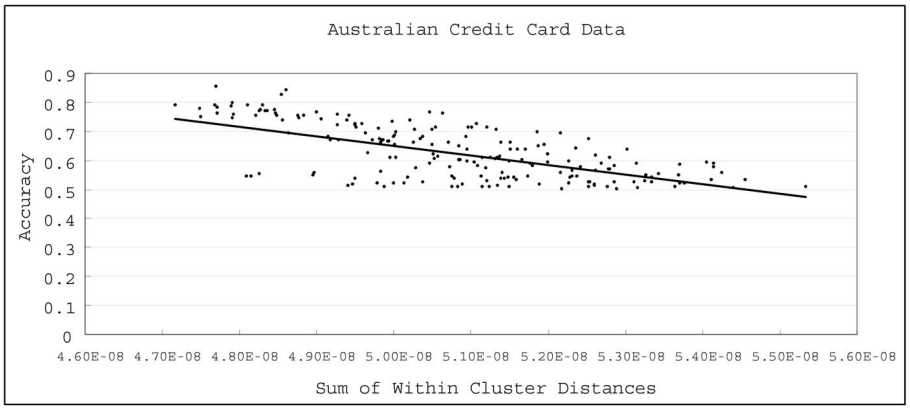
Table 5 is the summary of the 2,000 clustering results of the Australian Credit data set with 20 different β values. The left column indicates the clustering accuracy (we use clustering accuracy here because clustering accuracy is more obvious to compare with the real classes). Each column of a particular β value represents 100 clustering results. Each number indicates the number of clustering results achieving the corresponding accuracy. The left column ($\beta = 0$) lists the results produced by the algorithm with equal variable weights.

TABLE 5
The Credit Data Set: Summary of 2,000 Clustering Results Produced with Various β Values

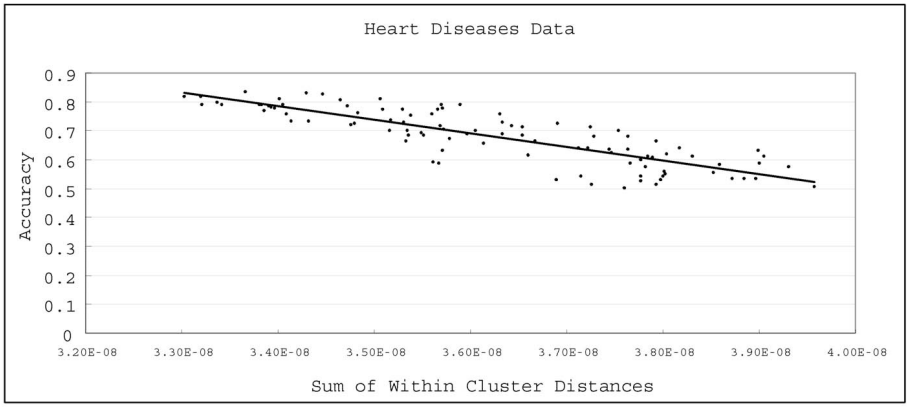
Accuracy	β=-10	β=-9	β=-8	β=-7	β=-6	β=-5	β=-4	β=-3	β=-2	β=-1	β=2	β=3	β=4	β=5	β=6	β=7	β=8	β=9	β=10	β=0
0.85																	1	1	1	
0.84																				
0.83																	1	1	1	
0.82																4				3
0.81	4	4	6	5	7	13	10	7	12	11			8	46	39	42				13
0.80	32	32	27	23	22	15	19	19	10	16			6	11	18	11				6
0.79	6	6	8	8	7	7	7	8	8	1			2							4
0.78	3	3	3	3	4	2	1	2	2	5			3							2
0.77	7	6	6	6	4	5	5	5	7	5			19				3	3	3	2
0.76	1	2	2	2	4	5	5	6	3								3	3	3	10
0.75									4	8							4	4	4	3
0.74																	4	4	4	3
0.73								1									3	3	3	2
0.72								1												
≤0.71	47	47	48	53	52	53	53	53	54	54	100	100	62	43	43	43	81	81	81	52

TABLE 6
The Heart Data Set: Summary of 2,000 Clustering Results Produced with Various β Values

Accuracy	β=-10	β=-9	β=-8	β=-7	β=-6	β=-5	β=-4	β=-3	β=-2	β=-1	β=2	β=3	β=4	β=5	β=6	β=7	β=8	β=9	β=10	β=0
0.85																	1	1	1	
0.84						1	3		5											
0.83	2	4	5	6	8	11	13	14	2	13							5	5	5	13
0.82					1				6								4	4	4	
0.81			1	1	1	2	6	50	53	5							2	2	2	
0.80					1	52	72	21	10	44							3	3	3	49
0.79			1	5	63	17			3	14				4	1	8	4	4	4	23
0.78	93	91	88	83	7	9			4	6			4	41	97	91				
0.77					12							1	88	55	2		1	1	1	
0.76												73					3	3	3	
0.75												5					2	2	2	
0.74												2	2				5	5	5	
0.73								1			1	3					7	7	7	
0.72								1				2	4							
≤0.71	5	5	5	5	7	8	9	13	17	18	99	14	2			1	63	63	63	15



(a)



(b)

Fig. 7. Credit Card Data with $\beta = 9$. (a) The relationship between clustering accuracy and the value of the objective function (8). Shows the results of 100 clusterings of the Australian Credit Card data. Shows the results of 100 clusterings of the Heart Disease data. (b) Heart Diseases Data with $\beta = 9$.

We achieved two best results of 85 percent of the clustering accuracy (of 0.74 of the Rand Index) at $\beta = 9$ and $\beta = 10$. This result is 2 percent higher than our previous clustering result [1] and the clustering result reported in [13] on the same data set. This demonstrates that the new clustering algorithm with variable weighting was able to produce highly accurate clustering results. The overall good clustering results occurred with β ranging from 4 to 7, where more than 50 percent of the clustering results got a high accuracy, while only 22 percent of the clustering results from $\beta = 0$ got accuracy of 80 percent.

Table 6 is the summary of the results of the Heart data set. In this data set, we obtained three best results of 85 percent of the clustering accuracy (of 0.74 of the Rand Index) at $\beta \in \{8, 9, 10\}$, which is also 2 percent higher than the result reported in [13] on the same data set. In this data set, the overall good results occurred at $\beta \in \{-5, -4, -3, -2\}$ in comparison with the results of $\beta = 0$.

Fig. 7 shows the relationship between the clustering accuracy and the value of the objective function (8). Fig. 7a is the result of the Australian Credit Card data and Fig. 7b is the result of the Heart Disease data. Each figure represents 100 clustering results with $\beta = 9$. From these figures, we can see that good clustering results were correlated to small values of the objective function (8). This indicates that, when we use this algorithm to cluster a data set, we can select the result with the minimal cost value from several clustering results on the same data set.

5.2.3 Variable Selection

In clustering a data set, the algorithm produced a weight for each variable. The importance of the variable in generating the clustering result can be analyzed from the weights. Table 7 shows the weights of variables of two best clustering results from the two real data sets. According to the weight values, we removed the last two variables from the Australian Credit Card data set and the seventh variable from the Heart Disease data set. We then

TABLE 7
The Weights of Variables from Two Good Results of the Two Data Sets

Credit Card Data				Heart Disease Data			
v_1	0.0130	v_9	0.1670	v_1	0.1176	v_9	0.0122
v_2	0.1652	v_{10}	0.0139	v_2	0.0091	v_{10}	0.1553
v_3	0.1871	v_{11}	0.0088	v_3	0.0069	v_{11}	0.0104
v_4	0.0167	v_{12}	0.0083	v_4	0.1492	v_{12}	0.0070
v_5	0.0167	v_{13}	0.0167	v_5	0.3331	v_{13}	0.0122
v_6	0.0044	$^{**}v_{14}$	0.0044	v_6	0.0123		
v_7	0.0093	$^{**}v_{15}$	0.0021	$^{**}v_7$	0.0064		
v_8	0.5167			v_8	0.1684		

*** Denotes that the variables are deleted in the new clustering results.*

TABLE 8
The Credit Data Set with the 14 and 15 Variables Removed: Summary of 2,000 Clustering Results Produced with Various β Values

Accuracy	β=-10	β=-9	β=-8	β=-7	β=-6	β=-5	β=-4	β=-3	β=-2	β=-1	β=2	β=3	β=4	β=5	β=6	β=7	β=8	β=9	β=10	β=0
0.86																		1	1	
0.85																				
0.84																				
0.83																				
0.82																2	4			
0.81	2	1	1	2	1		1						6	32	51	41	45	2	2	
0.80	35	36	40	38	38	37	36	29	28	24			7	33	16	24	19			31
0.79	10	10	6	7	5	4	3	11	10	9			1	1		1				17
0.78	3	3	3	3	4	4	4	3	1				3					1	1	10
0.77	20	20	20	20	20	20	20	21	15	11			29					2	2	10
0.76									10	16			1					4	4	
0.75																		5	5	
0.74															1			2	2	2
0.73								1										4	4	
0.72								1										2	2	
$\leq$0.71	30	30	30	30	32	35	36	36	36	40	100	100	53	34	32	32	32	77	77	30

TABLE 9
The Heart Data Set with the Seventh Variable Removed: Summary of 2,000 Clustering Results Produced with Various β Values

Accuracy	β=-10	β=-9	β=-8	β=-7	β=-6	β=-5	β=-4	β=-3	β=-2	β=-1	β=2	β=3	β=4	β=5	β=6	β=7	β=8	β=9	β=10	β=0
0.84	1									5							4	4	4	
0.83	12	12	11	8	1	1	1	2	5	14							4	4	4	23
0.82	5	7	6	5	14	6	4										2	2	2	
0.81	33	61	48	48	15	26	73		3	3							3	3	3	26
0.80	37	13	28	28	52	49	2	78	72	60							3	3	3	44
0.79	6	1	1	4	14	15	17	17	14	9				2	5	6	5	5	5	
0.78													2	81	92	90	2	2	2	
0.77												5	91	14			8	8	8	
0.76												11	1				3	3	3	
0.75														2			2	2	2	
0.74												71					1	1	1	
0.73																	5	5	5	
0.72																	3	3	3	
$\leq$0.71	6	6	6	7	4	3	3	3	6	9	100	13	6	1	3	4	55	55	55	7

conducted similar cluster analysis on the two reduced data sets. Tables 8 and 9 show the clustering results. We obtained the two best results of 86 percent of the clustering accuracy (of 0.742 of the Rand Index) from the reduced Australian Credit Card data, higher than the best result shown in Table 5. The other improvement was that occurrences of high accuracy results increased at most β values. This indicates that, after less important variables removed, it is easier to obtain a good clustering result. Improvement was also observed from the results of the Heart Disease data set (see Table 9).

6 Conclusions

In this paper, we have presented W-k-means, a new k-means type algorithm that can calculate variable weights automatically. Based on the current partition in the iterative k-means clustering process, the algorithm calculates a new weight for each variable based on the variance of the within cluster distances. The new weights are used in deciding the cluster memberships of objects in the next iteration. The optimal weights are found when the algorithm converges. The weights can be used to identify important variables for clustering and the variables which may contribute noise to the clustering process and can be removed from the data in the future analysis.

The experimental results on both synthetic data and real data sets have shown that the W-k-means algorithm outperformed the k-means type algorithms in recovering clusters in data. The synthetic data experiments have demonstrated that the weights can effectively distinguish noise variables from the normal variables. We have also demonstrated that the insignificant variables can be identified according to the weight values and removal of these variables could improve the clustering results. This capability is very useful in selecting variables for clustering in real data mining applications.

Acknowledgments

J.Z. Huang's research was supported in part by NSFC grants 60473091 and 60475026. M.K. Ng's research was supported in part by Research Grant Council Grant Nos. HKU 7130/02P, 7046/03P, and 7035/04P.

References

[1] Z. Huang, "Extensions to the k-Means Algorithms for Clustering Large Data Sets with Categorical Values," *Data Ming and Knowledge Discovery,* vol. 2, no. 3, pp. 283-304, 1998.
[2] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observation," *Proc. Fifth Berkeley Symp. Math. Statistica and Probability,* pp. 281-297, 1967.
[3] P.E. Green, F.J. Carmone, and J. Kim, "A Preliminary Study of Optimal Variable Weighting in k-Means Clustering," *J. Classification,* vol. 7, pp. 271-285, 1990.
[4] W.S. Desarbo, J.D. Carroll, L.A. Clark, and P.E. Green, "Synthesized Clustering: A Method for Amalgamating Clustering Bases with Differential Weighting Variables," *Psychometrika,* vol. 49, pp. 57-78, 1984.
[5] G. De Soete, "Optimal Variable Weighting for Ultrametric and Additive Tree Clustering," *Quality and Quantity,* vol. 20, pp. 169-180, 1986.
[6] G. De Soete, "OVWTRE: A Program for Optimal Variable Weighting for Ultrametric and Additive Tree Fitting," *J. Classification,* vol. 5, pp. 101-104, 1988.
[7] E. Fowlkes, R. Gnanadesikan, and J. Kettenring, "Variable Selection in Clustering," *J. Classification,* vol. 5, pp. 205-228, 1988.
[8] G. Milligan, "A Validation Study of a Variable Weighting Algorithm for Cluster Analysis," *J. Classification,* vol. 6, pp. 53-71, 1989.
[9] R. Gnanadesikan, J. Kettenring, and S. Tsao, "Weighting and Selection of Variables for Cluster Analysis," *J. Classification,* vol. 12, pp. 113-136, 1995.
[10] V. Makarenkov and B. Leclerc, "An Algorithm for the Fitting of a Tree Metric According to a Weighted Least-Squares Criterion," *J. Classification,* vol. 16, pp. 3-26, 1999.
[11] V. Makarenkov and P. Legendre, "Optimal Variable Weighting for Ultrametric and Additive Trees and K-Means Partitioning: Methods and Software," *J. Classification,* vol. 18, pp. 245-271, 2001.
[12] J.H. Friedman and J.J. Meulman, "Clustering Objects on Subsets of Attributes," *J. Royal Statistical Soc. B.,* 2002.

[13] D.S. Modha and W.S. Spangler, "Feature Weighting in k-Means Clustering," *Machine Learning*, vol. 52, pp. 217-237, 2003.
[14] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications," *Proc. ACM SIGMOD*, pp. 94-105, June 1998.
[15] J. Bezdek, "A Convergence Theorem for the Fuzzy Isodata Clustering Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, no. 1, pp. 1-8, 1980.
[16] Z. Huang and M. Ng, "A Fuzzy k-Modes Algorithm for Clustering Categorical Data," *IEEE Trans. Fuzzy Systems*, vol. 7, no. 4, pp. 446-452, 1999.
[17] M. Anderberg, *Cluster Analysis for Applications.* Academic Press, 1973.
[18] S. Selim and M. Ismail, "K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 81-87, 1984.
[19] G. Milligan and P. Isaac, "The Valiadation of Four Ultrametric Clustering Algorithms," *Pattern Recognition*, vol. 12, pp. 41-50, 1980.
[20] A. Jain and R. Dubes, *Algorithms for Clustering Data.* Prentice Hall, 1988.
[21] G.S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*, p. 19. Springer-Verlag, 1996.

Joshua Zhexue Huang received the PhD degree from the Royal Institute of Technology in Sweden. He is the assistant director of the E-Business Technology Institute (ETI) at the University of Hong Kong. Before joining ETI in 2000, he was a senior consultant at the Management Information Principles, Australia, consulting on data mining and business intelligence systems. From 1994 to 1998, he was a research scientist at CSIRO Australia. His research interests include data mining, machine learning, clustering algorithms, and grid computing.

Michael K. Ng received the BSc and MPhil degrees in mathematics from the University of Hong Kong in 1990 and 1993, respectively, and the PhD degree in mathematics from the Chinese University of Hong Kong in 1995. From 1995 to 1997, he was a research fellow at the Australian National University. He is an associate professor in the Department of Mathematics at the University of Hong Kong. His research interests are in the areas of data mining, operations research, and scientific computing. He has been selected as one of the recipients of the Outstanding Young Researcher Award of the University of Hong Kong in 2001.

Hongqiang Rong received the BS and MS degrees in computer science and engineering from Harbin Institute of Technology in China in 1998 and 2000, respectively. He is a PhD candidate in the Department of Computer Science, the University of Hong Kong.

Zichen Li received the BS and MS degrees in applied mathematics and the PhD degree in electrical engineering in 1985, 1986, and 1999, respectively. He is now the director of the Department of Computer Secince and Technology, Henan Polytechnic University, China. His research interests include coding theory, communications theory, modern cryptography, signal processing, and data mining.