PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT

Pattern Recognition 37 (2004) 943–952

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

An optimization algorithm for clustering using weighted dissimilarity measures☆

Elaine Y. Chan[a], Wai Ki Ching[a], Michael K. Ng[a,*], Joshua Z. Huang[b]

[a] *Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong*
[b] *E-Business Technology Institute, The University of Hong Kong, Pokfulam Road, Hong Kong*

Received 11 March 2003

Abstract

One of the main problems in cluster analysis is the weighting of attributes so as to discover structures that may be present. By using weighted dissimilarity measures for objects, a new approach is developed, which allows the use of the k-means-type paradigm to efficiently cluster large data sets. The optimization algorithm is presented and the effectiveness of the algorithm is demonstrated with both synthetic and real data sets.

Keywords: Clustering; Data mining; Optimization; Attributes weights

1. Introduction

Partitioning a set of objects into homogeneous clusters is a fundamental operation in data science, see for instance Anderberg [1], and Ball and Ball [2]. The operation is required in a number of data analysis 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 modeled and analyzed.

The clustering problem has been studied extensively and many algorithms have been developed according to the specific domain requirements (e.g. Jain [3]). Usually, the distance or dissimilarity functions of these algorithms involve all attributes of the data set. This is applicable if all or most attributes are important to every cluster. However, the clustering results become less accurate if a significant number of attributes are not important to some clusters. In the literature, the problem of the selection of important attributes is solved by feature selection techniques, see Duda and Hart [4]. These techniques derive a set of important features from all the attributes for the clustering analysis. The feature selection can be viewed as a preprocessing step in the clustering procedure. The limitation of this preprocessing step is that the clustering algorithm is still based on the whole set of new features. However, each cluster may have a different set of important attributes, and each cluster may contain some unimportant features. The traditional feature selection techniques cannot handle these characteristics of clusters.

1.1. An example

Let us consider an example with nine objects measured on four attributes (x_1, x_2, x_3, x_4) as shown in Table 1. The plots of x_1 versus x_2 in Fig. 1(left) and x_3 versus x_4 in Fig. 1 (right) reveal three well separated homogeneous clusters. Thus x_1 and x_2 are the important attributes for two clusters, where the objects 1, 2 and 3 (represented by "x") form one cluster and the objects 4, 5 and 6 (represented by "o") form another cluster. We see that the objects 7, 8 and 9

☆ The research is supported in part by Hong Kong Research Grants Council Grant Nos. HKU 7126/02P, HKU 7130/02P and HKU 7046/03P.

* Corresponding author. Tel.: +852-2859-2252; fax: +852-2559-2225.
E-mail address: mng@maths.hku.hk (M.K. Ng).

0031-3203/$30.00
doi:10.1016/j.patcog.2003.11.003

Table 1
An example data set

Object	1	2	3	4	5	6	7	8	9
x_1	6	7	8	2	3	4	1	3	10
x_2	14	15	13	3	1	2	15	8	3
x_3	1	5	6	7	7	1	12	14	14
x_4	13	1	5	6	12	2	3	4	2

(represented by "+") do not form a cluster using x_1 and x_2 attributes. However, the objects 7–9 indeed form a cluster using x_3 and x_4 attributes.

1.2. The main idea

To tackle the above clustering problem, the weighted dissimilarity measure is defined. In addition to finding out the cluster of objects belong to, the weighted dissimilarity measure also gives the set of important attributes for each cluster. We will use the term "a set of attributes with high weights" to describe the set of attributes with the highest power in distinguishing the objects of a cluster from other objects. For example, in Table 2, we give different attributes weights for the three clusters of the data set in Table 1. We see that the weights of the attributes x_1 and x_2 (the attributes x_3 and x_4) in the clusters "x" and "o" are high (are low), it reflects they are more important (are less important). There are similar attributes weighting results for the cluster "+".

In this paper we introduce a attributes-weighting clustering algorithm which generalizes k-means-type algorithms in Ref. [3]. This is achieved by the development of a new procedure to generate a weight for each attribute from each cluster within the framework of the k-means-type algorithm. The main result of this paper is to provide a method to find the weight for each attribute from each cluster. We remark that the weight of each attribute for each cluster in Table 2 is generated by our proposed algorithm.

The rest of the paper is organized as follows. In Section 2, we present our attributes weighting clustering algorithm. In Section 3, we demonstrate the effectiveness of our method with experimental results. Finally, a summary is given to conclude the paper in Section 4.

2. The attributes-weighting clustering algorithm

We assume the set of objects to be clustered is stored in a database table $\boldsymbol{T}$ defined by a set of attributes, $A_1, A_2, \ldots, A_m$. Each attribute A_j describes a domain of values, denoted by $\mathrm{DOM}(A_j)$, associated with a defined semantic and a data type. In this paper, we only consider two general data types, *numeric* and *categorical* and assume other types used in database systems can be mapped to one of these two types. The domains of attributes associated with these two types are called numeric and categorical, respectively. A numeric domain consists of real numbers. A domain $\mathrm{DOM}(A_j)$ is defined as categorical if it is finite and unordered, e.g., for any $a, b \in \mathrm{DOM}(A_j)$, either $a = b$ or $a \neq b$, see for instance Gowda and Diday [5].

An object X in $\boldsymbol{T}$ can be logically represented as a conjunction of attribute-value pairs

$$[A_1 = x_1] \wedge [A_2 = x_2] \wedge \cdots \wedge [A_m = x_m],$$

where $x_j \in \mathrm{DOM}(A_j)$ for $1 \leqslant j \leqslant m$. Without ambiguity, we represent X as a vector

$$[x_1, x_2, \ldots, x_m].$$

The vector X is called a categorical object if it has only categorical values. We assume each object has exactly m attribute values. If the value of an attribute A_j is

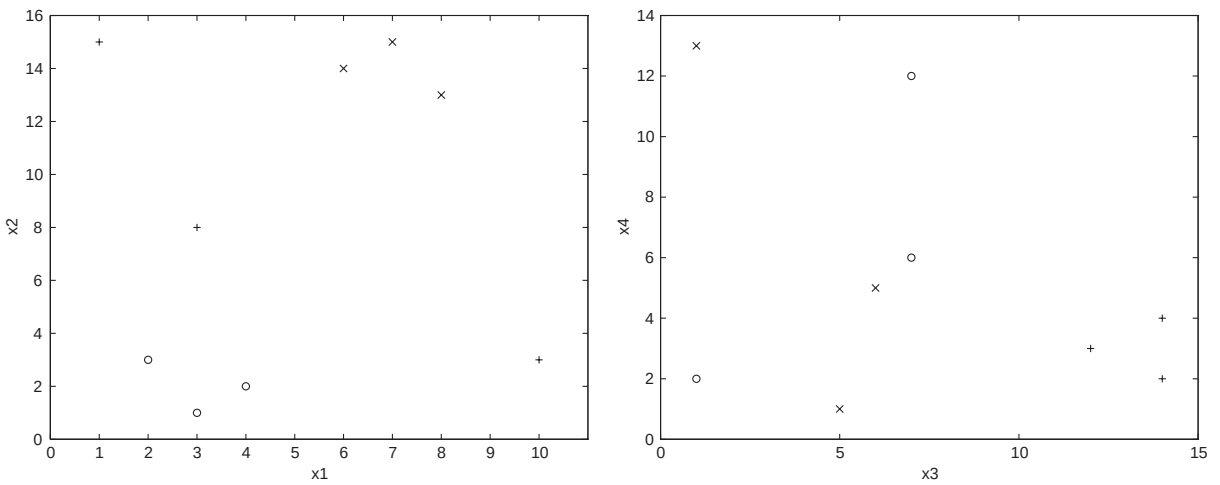

Fig. 1. The plots of x_1 verus x_2 (left) and x_3 verus x_4 (right).

Table 2
Attributes weights

Attribute	x_1	x_2	x_3	x_4
cluster "x"	0.2910	0.4895	0.1675	0.0520
cluster "o"	0.2996	0.5038	0.1317	0.0649
cluster "+"	0.0730	0.0962	0.4547	0.3761

missing, then we denote the attribute value of A_j by ε. Let $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$ be a set of n objects. 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 $1 \leqslant j \leqslant m$. The relation $X_i = X_k$ does not mean that X_i and X_k are the same object in the real world database, but rather they are two objects having equal values in the attributes $A_1, A_2, \ldots, A_m$.

In this paper, we only consider numerical and categorical attributes of the objects to be clustered.

2.1. The optimization problem

Let $\boldsymbol{X}$ be a set of n objects described by m attributes. The attributes-weighting clustering algorithm to cluster $\boldsymbol{X}$ into k clusters can be stated as an algorithm which attempts to minimize the cost function

$$F(W, Z, \Lambda) = \sum_{l=1}^{k} \sum_{j=1}^{n} \sum_{i=1}^{m} w_{l,j} \lambda_{l,i}^{\beta} d(z_{l,i},\ x_{j,i}) \tag{1}$$

subject to

$$w_{l,j} \in \{0, 1\}, \quad 1 \leqslant l \leqslant k,\ 1 \leqslant j \leqslant n, \tag{2}$$

$$\sum_{l=1}^{k} w_{l,j} = 1, \quad 1 \leqslant j \leqslant n, \tag{3}$$

$$0 < \sum_{i=1}^{n} w_{l,j} < n, \quad 1 \leqslant l \leqslant k, \tag{4}$$

$$\lambda_{l,i} \geqslant 0, \quad 1 \leqslant l \leqslant k, \quad 1 \leqslant i \leqslant m, \tag{5}$$

and

$$\sum_{i=1}^{m} \lambda_{l,i} = 1, \quad 1 \leqslant l \leqslant k, \tag{6}$$

where $k (\leqslant n)$ is a known number of clusters, β is an exponent greater than 1, $W = [w_{l,j}]$ is a k-by-n integer matrix, $Z = [Z_1, Z_2, \ldots, Z_k] \in \mathbb{R}^{m \times k}$ contains the cluster centers, $\Lambda = [\lambda_{l,i}]$ is a k-by-m real matrix, and $d(z_{l,i},\ x_{j,i}) (\geqslant 0)$ is a dissimilarity measure between the ith attribute of the center Z_l and the object X_j. The main idea of the above optimization problem is to minimize the dissimilarity measure between the cluster centers and the objects. The dissimilarity measure is defined by m weighted attributes. Such objective function allows us to consider the weight for each attribute from each cluster.

In the literature, the Euclidean norm

$$d_n(z_{l,i},\ x_{j,i}) = |z_{l,i} - x_{j,i}|^2$$

is commonly used for numerical attributes in clustering algorithms. For categorical attributes, the simple matching dissimilarity measure [6,7] between $x_{j,i}$ and $z_{l,i}$ is defined as follows:

$$d_c(z_{l,i},\ x_{j,i}) \equiv \delta(x_j, y_j) = \begin{cases} 0, & x_{j,i} = z_{l,i} \\ 1, & x_{j,i} \neq z_{l,i}. \end{cases} \tag{7}$$

Minimization of $F(\cdot, \cdot, \cdot)$ in Eq. (1) with constraints (2)–(6) forms a class of constrained nonlinear optimization problems whose solutions are unknown. The usual method towards optimization of $F(\cdot, \cdot, \cdot)$ in Eq. (1) is to use partial optimization for Z, Λ and W. In this method we first fix Z and Λ and find necessary conditions on W to minimize $F(\cdot, \cdot, \cdot)$. Then we fix W and Λ and minimize $F(\cdot, \cdot, \cdot)$ with respect to Z. We then fix W and Z and minimize $F(\cdot, \cdot, \cdot)$ with respect to Λ. The process is repeated until no more improvement in the objective function value can be made. The above procedure is formalized in the algorithm as follows:

Algorithm 1. The Attributes-Weighting Algorithm

1. 1.] Choose an initial matrix $Z^{(1)} \in \mathbb{R}^{m \times k}$ and set $\Lambda^{(1)}$ be a k-by-m matrix with all the entries being equal to $1/m$. Set $t = 1$.
2. Determine $W^{(t+1)}$ such that $F(W^{(t+1)}, Z^{(t)}, \Lambda^{(t)})$ is minimized. If

   $$F(W^{(t+1)}, Z^{(t)}, \Lambda^{(t)}) = F(W^{(t)}, Z^{(t)}, \Lambda^{(t)}),$$

   then stop; otherwise goto step 3.
3. Determine $Z^{(t+1)}$ such that $F(W^{(t+1)}, Z^{(t+1)}, \Lambda^{(t)})$ is minimized. If

   $$F(W^{(t+1)}, Z^{(t+1)}, \Lambda^{(t)}) = F(W^{(t+1)}, Z^{(t+1)}, \Lambda^{(t)}),$$

   then stop; otherwise goto Step 4.
4. Determine $\Lambda^{(t+1)}$ such that $F(W^{(t+1)}, Z^{(t+1)}, \Lambda^{(t+1)})$ is minimized. If

   $$F(W^{(t+1)}, Z^{(t+1)}, \Lambda^{(t+1)}) = F(W^{(t+1)}, Z^{(t+1)}, \Lambda^{(t)}),$$

   then stop; otherwise goto Step 2.

The matrices W, Z and Λ are computed according to the following theorems.

Theorem 1. *Let $\tilde{Z}$ and $\tilde{\Lambda}$ be fixed. The minimizer $\hat{W}$ of the optimization problem*

$$\min_{W} F(W,\tilde{Z},\tilde{\Lambda}) \quad \text{subject to} \quad (2)\text{–}(4)$$

is given by

$$\hat{w}_{l,j} = \begin{cases} 1 & \text{if } \sum_{i=1}^{m} \tilde{\lambda}_{l,i}^{\beta}(\tilde{z}_{l,i} - x_{j,i})^2 \leqslant \sum_{i=1}^{m} \tilde{\lambda}_{h,i}^{\beta}(\tilde{z}_{h,i} - x_{j,i})^2, \\ & 1 \leqslant h \leqslant k, \\ 0 & \text{otherwise.} \end{cases}$$

The proof of Theorem 1 can be found in Ref. [8]. We remark that the minimum solution $\hat{W}$ is not unique, so $w_{l,j}=1$ may arbitrarily be assigned to the first minimizing index l, and the remaining entries of this column are put to zero.

Theorem 2. *Let $\tilde{W}$ and $\tilde{\Lambda}$ be fixed. The minimizer $\hat{Z}$ of the optimization problem*

$$\min_{Z} F(\tilde{W},Z,\tilde{\Lambda})$$

is given by

$$\hat{z}_{l,i} = \frac{\sum_{j=1}^{n} \tilde{w}_{l,j} x_{i,j}}{\sum_{j=1}^{n} \tilde{w}_{l,j}}, \quad \text{where } 1 \leqslant l \leqslant k,$$

when the ith attribute is numeric, or

$$\hat{z}_{l,i} = a_i^{(r)} \in \mathrm{DOM}(A_i)$$

with

$$|\{\tilde{w}_{l,j} | x_{i,j} = a_i^{(r)}, \tilde{w}_{l,j} = 1\}| \geqslant |\{\tilde{w}_{l,j} | x_{i,j} = a_i^{(t)}, \tilde{w}_{l,j} = 1\}|, \quad \forall t \in \mathrm{DOM}(A_i),$$

when the ith attribute is categorical. Here $|Y|$ denotes the number of elements in the set Y.

Proof. When $\tilde{W}$ and $\tilde{\Lambda}$ are fixed, all the inner sums of the quantity

$$\sum_{l=1}^{k} \sum_{i=1}^{m} \tilde{\lambda}_{l,i}^{\beta} \sum_{j=1}^{n} \tilde{w}_{l,j} d(z_{l,i}, x_{j,i})$$

are nonnegative and independent. Minimizing the quantity is equivalent to minimizing each inner sums

$$\sum_{j=1}^{n} \tilde{w}_{l,j} d(z_{l,i}, x_{j,i})$$

for $1 \leqslant l \leqslant k$ and $1 \leqslant i \leqslant m$. We note that the inner sum is independent of $\tilde{\lambda}_{l,i}$. When the ith attribute is numeric, the inner sum is minimized if and only if

$$\sum_{j=1}^{n} \tilde{w}_{l,j} |\hat{z}_{l,i} - x_{j,i}|^2 = 0.$$

Hence the result follows. When the ith attribute is categorical, we write the inner sum as

$$\sum_{j=1}^{n} \tilde{w}_{l,j} \delta(z_{l,j}, x_{j,i}) = n \left(1 - \frac{|\{\tilde{w}_{l,j} | x_{j,i} = z_{l,i}, \tilde{w}_{l,j} = 1\}|}{n}\right).$$

The inner sum is minimized if and only if

$$\left(1 - \frac{|\{\tilde{w}_{l,j} | x_{j,i} = z_{l,i}, \tilde{w}_{l,j} = 1\}|}{n}\right)$$

is minimal, or the term $|\{\tilde{w}_{l,j} | x_{j,i} = z_{l,i}, \tilde{w}_{l,j} = 1\}|$ must be maximal. The result follows. □

According to the above theorem, the value or the category of the ith attribute of the lth cluster center is determined by the mean value or the mode, respectively, in the set of objects belonging to the cluster.

Theorem 3. *Let $\tilde{W}$ and $\tilde{Z}$ be fixed. The minimizer $\hat{\Lambda}$ of the optimization problem*

$$\min_{\Lambda} F(\tilde{W},\tilde{Z},\Lambda) \quad \text{subject to} \quad (5) \text{ and } (6)$$

is given by

$$\hat{\lambda}_{l,i} = \begin{cases} \dfrac{1}{m_i} & \text{if } \sum_{j=1}^{n} \tilde{w}_{l,j}(\tilde{z}_{l,i} - x_{j,i})^2 = 0, \text{ and} \\ & m_i = \left| \left\{ t : \sum_{j=1}^{n} \tilde{w}_{l,j}(\tilde{z}_{l,t} - x_{j,t})^2 = 0 \right\} \right|, \\ 0 & \text{if } \sum_{j=1}^{n} \tilde{w}_{l,j}(\tilde{z}_{l,i} - x_{j,i})^2 \neq 0, \text{ but} \\ & \sum_{j=1}^{n} \tilde{w}_{l,j}(\tilde{z}_{l,t} - x_{j,t})^2 = 0, \text{ for some } t, \\ \dfrac{1}{\sum_{t=1}^{m} \left[\frac{\sum_{j=1}^{n} \tilde{w}_{l,j}(\tilde{z}_{l,i} - x_{j,i})^2}{\sum_{j=1}^{n} \tilde{w}_{l,j}(\tilde{z}_{l,t} - x_{j,t})^2} \right]^{1/(\beta-1)}} & \\ & \text{if } \sum_{j=1}^{n} \tilde{w}_{l,j}(\tilde{z}_{l,t} - x_{j,t})^2 \neq 0, \quad \forall 1 \leqslant t \leqslant m. \end{cases}$$

The proof of Theorem 3 is similar to Theorem 2 in Ref. [8]. In Ref. [8], Bezdek considered a similar optimization problem for the fuzzy clustering.

Combining the results in Theorems 1–3, Algorithm 1 is our attributes-weighting algorithm, in which the partition matrix is computed according to Theorem 1, the cluster centers in each iteration are updated according to Theorem 2, the attributes-weighting matrix is calculated according to Theorem 3.

Table 3
The clustering accuracy results for $\tau = 0.5$

k	m	(γ, α) $(100, 0.8)$	(γ, α) $(10, 0.8)$	(γ, α) $(5, 0.8)$	(γ, α) $(100, 0.6)$	(γ, α) $(10, 0.6)$	(γ, α) $(5, 0.6)$
3	3	98.67 (87.47)	93.67 (89.07)	95.80 (88.33)	98.47 (89.73)	96.67 (89.00)	95.35 (84.53)
	4	99.35 (78.75)	98.60 (79.15)	96.00 (82.55)	99.50 (78.20)	98.70 (78.50)	94.93 (84.70)
	5	96.08 (75.60)	93.20 (76.16)	91.28 (78.16)	96.13 (71.80)	86.88 (71.76)	86.18 (74.08)
4	4	100.0 (93.20)	99.70 (94.70)	99.45 (96.20)	99.95 (91.90)	99.25 (92.05)	98.55 (92.30)
	5	97.04 (87.72)	96.44 (90.24)	94.04 (89.88)	99.78 (82.16)	88.80 (84.16)	95.28 (86.56)
	6	90.71 (85.50)	94.17 (85.00)	90.00 (87.20)	90.73 (83.10)	93.90 (84.23)	93.30 (83.83)
	7	90.77 (79.46)	88.29 (81.49)	94.33 (81.06)	96.23 (80.63)	94.94 (81.51)	89.51 (80.63)
5	5	100.0 (98.24)	99.96 (98.28)	99.92 (98.32)	100.0 (96.40)	99.88 (96.24)	99.64 (96.32)
	6	100.0 (93.77)	99.87 (94.83)	99.60 (94.77)	100.0 (93.37)	99.73 (93.80)	99.30 (93.57)
	7	99.69 (89.51)	93.69 (89.11)	93.60 (89.34)	95.52 (89.54)	95.06 (91.14)	94.83 (91.23)
	8	94.39 (84.38)	95.00 (84.80)	90.15 (84.93)	96.25 (90.80)	99.90 (87.15)	93.08 (89.55)
	9	88.84 (84.31)	93.75 (81.51)	85.47 (80.89)	94.10 (84.36)	89.78 (82.33)	94.09 (85.24)

Theorem 4. *The attributes-weighting algorithm converges in a finite number of iterations.*

Proof. We first note that there are only a finite number of possible partitions W. We can show that each possible partition W appears at most once by the algorithm. Assume that $W^{(t_1)} = W^{(t_2)}$ where $t_1 \neq t_2$. We note that given $W^{(t)}$, we can compute the minimizer $Z^{(t)}$ which is independent of $\Lambda^{(t)}$. For $W^{(t_1)}$ and $W^{(t_2)}$, we have the minimizers $Z^{(t_1)}$ and $Z^{(t_2)}$ respectively. It is clear that $Z^{(t_1)} = Z^{(t_2)}$ since $W^{(t_1)} = W^{(t_2)}$. Using $W^{(t_1)}$ and $Z^{(t_1)}$, and $W^{(t_2)}$ and $Z^{(t_2)}$, we can compute the minimizers $\Lambda^{(t_1)}$ and $\Lambda^{(t_2)}$ respectively (Step 3) according to Theorem 1. Again, $\Lambda^{(t_1)} = \Lambda^{(t_2)}$. Therefore, we have

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

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

We have shown that the attributes-weighting algorithm is convergent. However, we remark that it may terminate at a local minimum, see for instance Bezdek [8] and MacQueen [9]. The computational complexity of the algorithm is $O(tmnk)$ where t is the total number of cycles required, k is the number of clusters, m is the number of attributes and n is the number of objects. As for the storage, we need $O(n(m + k) + 2km)$ space to hold the set of n objects, the cluster centers Z and the partition matrix W, the attributes-weighting matrix. Therefore, the proposed clustering algorithm is best suited for dealing with large data sets.

3. Experimental results

To evaluate the performance and efficiency of the attributes-weighting algorithm and compare it with the algorithm without weighting on the attributes. The latter one is just the standard k-means algorithm [3,9], we carried out several tests of these algorithms on both real and artificial data sets. A clustering result is measured by the clustering accuracy r defined as

$$r = \frac{\sum_{l=1}^{k} a_l}{n},$$

where a_l is the number of instances occurring in both cluster l and its corresponding generated cluster label, and n is the number of instances in the data set.

3.1. Synthetic data sets

We conducted experiments on some synthetic data sets with different number of attributes ($m = 3, 4, 5, 6$) and numbers of clusters ($k = m, m + 1, \ldots, 2m - 1$). For each cluster, there are 50 objects.

We randomly assign the ith attribute of the lth cluster being important ($\lambda_{li} = 1$) or not important ($\lambda_{li} = 0$). To generate the data set, we first generate the center point $Z_1 = [z_{1,1}, z_{1,2}, \ldots, z_{1,m}]$ of the first cluster, and then the ith attributes of 50 objects for this cluster are generated according to the following rules:

$$\begin{cases} z_{l,i} + a & \text{if } \lambda_{l,i} = 1, \\ z_{l,i} + \gamma \cdot b & \text{if } \lambda_{l,i} = 0, \end{cases} \tag{8}$$

where a and b are the numbers randomly generated by the Gaussian distribution of zero mean and variance 1. Here γ is a number greater than 1 to control the variance of the Gaussian distribution for the unimportant attributes. We remark that when γ is large, the variance of the ith attribute value will be large, and hence the attribute value will be spread out from the center point.

After we generate the first cluster, we generate another cluster center point. In order to control the distance between

Table 4
The clustering accuracy results for $\alpha = 0.6$

k	m	(γ,τ) $(100, 0.5)$	(γ,τ) $(10, 0.5)$	(γ,τ) $(5, 0.5)$	(γ,τ) $(100, 0.4)$	(γ,τ) $(10, 0.4)$	(γ,τ) $(5, 0.4)$
3	3	98.47 (89.73)	96.67 (89.00)	95.35 (84.53)	98.47 (87.53)	96.73 (87.53)	94.40 (87.40)
	4	99.50 (78.20)	98.70 (78.50)	94.93 (84.70)	99.50 (78.50)	98.70 (78.50)	95.35 (84.70)
	5	96.13 (71.80)	86.88 (71.76)	86.18 (74.08)	96.13 (70.84)	86.84 (70.84)	86.88 (73.16)
4	4	99.95 (91.90)	99.25 (92.05)	98.55 (92.30)	99.95 (92.05)	99.25 (92.05)	98.55 (92.30)
	5	99.78 (82.16)	88.80 (84.16)	95.28 (86.56)	99.78 (84.16)	88.80 (84.16)	95.28 (83.60)
	6	90.73 (83.10)	93.90 (84.23)	93.30 (83.83)	92.70 (82.13)	94.93 (82.13)	90.00 (82.07)
	7	96.23 (80.63)	94.94 (81.51)	89.51 (80.63)	96.23 (80.69)	94.94 (80.69)	84.91(80.94)
5	5	100.0 (96.40)	99.88 (96.24)	99.64 (96.32)	100.0 (96.24)	99.88 (96.24)	99.64 (96.32)
	6	100.0 (93.37)	99.73 (93.80)	99.30 (93.57)	100.0 (93.80)	99.73 (93.80)	99.30 (93.57)
	7	95.52 (89.54)	95.06 (91.14)	94.83 (91.23)	95.52 (91.14)	95.06 (91.14)	94.83 (91.23)
	8	96.25 (90.80)	99.90 (87.15)	93.08 (89.55)	96.25 (87.15)	99.90 (87.15)	93.08 (89.55)
	9	94.10 (84.36)	89.78 (82.33)	94.09 (85.24)	94.10 (82.33)	89.78 (82.33)	94.09 (85.24)

Table 5
The Australian credit card data set

k	Our algorithm			Case (i)			Case (ii)		
	acc	best-acc	worst-acc	acc	best-acc	worst-acc	acc	best-acc	worst-acc
2	0.8167	0.8346	0.7963	0.6222	0.6294	0.6217	0.7228	0.8070	0.5467
4	0.8047	0.8545	0.7504	0.6083	0.6110	0.6034	0.7784	0.8499	0.6156
8	0.8141	0.8591	0.7458	0.6350	0.6677	0.6141	0.7996	0.8515	0.7274
16	0.8215	0.8606	0.7688	0.6719	0.7136	0.6447	0.8146	0.8499	0.7672

Table 6
The heart disease data set

k	Our algorithm			Case (i)		
	acc	best-acc	worst-acc	acc	best-acc	worst-acc
2	0.8071	0.8296	0.7815	0.7944	0.8185	0.5519
4	0.8017	0.8444	0.7259	0.7895	0.8370	0.7296
8	0.8057	0.8481	0.7519	0.8103	0.8556	0.7444
16	0.8193	0.8519	0.7815	0.8129	0.8667	0.7519

two cluster centers, the new center point is generated according to the following rules:

$$Z_l = X_{l-1} + \alpha \cdot d_{l-1} \cdot c.$$

Here X_{l-1} is the furthest object away from the cluster center point of the $(l-1)$th cluster, d_{l-1} is the corresponding distance, c is a number randomly generated by the Gaussian distribution of zero mean and variance 1. The parameter α is to control the position of the new cluster center point. In order to keep clusters separated, we require the distances between the new cluster center point and the previously generated cluster center points should be greater than or equal to

$$\tau d_v c, \quad v = 1, 2, \ldots, l-1.$$

Here τ is a parameter to control the separation. After we generate the new cluster center point, the objects for the new cluster are generated according to the rules in (8). The process is repeated until all the clusters are generated.

We used the above procedure to generate data sets and used these data sets to test our attributes-weighting algorithm to recover the cluster structures. For the initialization for the algorithm, we randomly assign objects in the data set

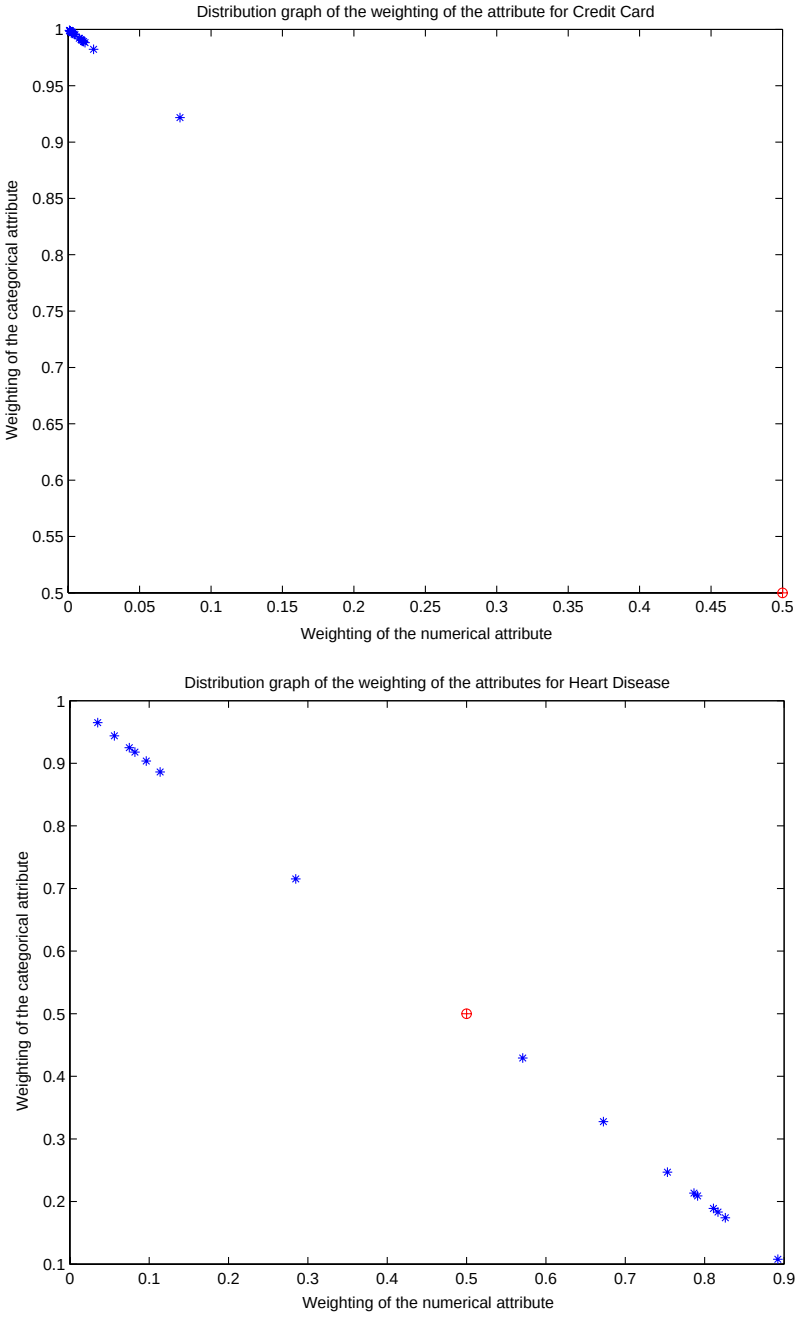

Fig. 2. The weights of the numeric and categorical attributes. *: our algorithm and o: case (i).

to be the cluster centers, i.e., choose an initial matrix Z in Algorithm 1.

The clustering accuracy results are reported in Tables 3 and 4. We remark that our algorithm is run 100 times, and each entry gives the average accuracy (i.e., the average percentages of the correctly classified objects over 100 runs) of clustering by our algorithm. For the comparison, we also give the clustering accuracy results (the number in the bracket in the table) of the k-means algorithm without weighting on the attributes.

We see from the tables that the attributes-weighting algorithm gives better clustering accuracy than the algorithm without weighting on the attributes for different parameters setting. We remark that the attributes-weighting algorithm

may terminate at a local minimum, and therefore it may not give an optimal solution to the clustering problems. To improve the method, one may use the Tabu search technique to incorporate in the attributes-weighting algorithm. Tabu search is a global search technique that provides means for escaping from local optimal solutions. It allows the search to explore the solution space beyond local optimality and attempts to find the global optimal solution. In Ref. [10], we have used the Tabu search technique in k-means algorithm and improved the clustering accuracy results. However, the computational cost is higher when the Tabu search technique is incorporated.

3.2. Real data sets

We have applied this algorithm to two real data sets: the Australian credit card data set and the Heart diseases data set [11]. Both of them have numerical and categorical attributes. The Australian credit card data set has 690 objects, each with 6 numeric and 9 categorical attributes. The data sets are originally classified into two clusters: "approved" or "rejected". Since some objects have missing values in 7 attributes, only 666 objects are considered. The Heart diseases data sets has 270 records with 13 attributes, 7 numeric and 6 categorical attributes. The records are classified into two classes: "absence" or "presence". Since there is no missing value in the data, all records are considered. We use these two mixed numeric and categorical data sets to test for the clustering accuracy of our algorithm.

In Refs. [6,10], a parameter β is used for the mixed data sets to balance the numeric and categorical parts to avoid favoring either type of attribute. Different values of β are tested and then the best value corresponding to the higher clustering accuracy is determined. Moreover, the value of β is fixed for each cluster in the experiments conducted in Refs. [6,10]. In this paper, we test our attributes-weighting algorithm with only two weighting variables for each cluster, i.e., $\lambda_{l,1}$ and $\lambda_{l,2}$. One is for all the numeric attributes ($\lambda_{l,1}$) and the other is for all the categorical attributes ($\lambda_{l,2}$).

The average clustering accuracy (acc) results are summarized in Tables 5 and 6. Here each algorithm is run 100 times. The best clustering accuracy (best-acc) and the worst clustering accuracy (worst-acc) results are listed. For the comparison, we show the clustering accuracy results of the Australian credit card data set for two cases:

(i) $\lambda_{l,1} = \lambda_{l,2} = 0.5$ for all clusters and
(ii) $\lambda_{l,1} = 0.4348$ and $\lambda_{l,2} = 0.5652$ for all clusters.

The latter case is the best parameter determined by the experimental results in [10]. For the Heart disease data set, we only compare our algorithm with the case $\lambda_{l,1} = \lambda_{l,2} = 0.5$ for all clusters. From Tables 5 and 6, we find that the clustering accuracy of our attributes-weighting algorithm is better than that of using fixed weight for each cluster. It is interesting to note that the attributes weighting are generated automatically in our algorithm. It is not necessary to try different values of attributes weighting and then select the best one. Also we remark that in the two cases of real data, the measure of accuracy assumes that all observations in a cluster are correctly labeled. One can never be sure of it in a real diagnostic situation. And this may be the reason why our proposed weighting of attributes works better for the credit card data than the Heart disease data.

Next we show the values of $\lambda_{l,1}$ (the weight of the numeric attribute) and $\lambda_{l,2}$ (the weight of the categorical attribute) computed by our algorithm in Fig. 2 when the number of

Table 7
The effect of β on clustering accuracy results (left) the credit card data set and (right) the Heart disease data set

β	average	best-acc	worst-acc
1.1	0.8065	0.8270	0.7933
1.2	0.8093	0.8331	0.7933
1.3	0.8071	0.8346	0.5467
1.4	0.8111	0.8346	0.6080
1.5	0.8143	0.8361	0.7933
1.6	0.8125	0.8361	0.6110
1.7	0.8136	0.8361	0.5467
1.8	0.8167	0.8346	0.7963
1.9	0.8140	0.8361	0.5758
2.0	0.8158	0.8315	0.7933
2.1	0.8129	0.8285	0.5467
2.2	0.8113	0.8254	0.5467
2.3	0.8044	0.8285	0.5467
2.4	0.8002	0.8254	0.5467
2.5	0.7955	0.8254	0.5467
2.6	0.7961	0.8224	0.5467
2.7	0.8038	0.8239	0.7642
2.8	0.7929	0.8254	0.5467
2.9	0.7789	0.8086	0.5467
3.0	0.7815	0.8009	0.7580
1.1	0.7555	0.7963	0.5593
1.2	0.7712	0.8037	0.5519
1.3	0.7873	0.8037	0.6704
1.4	0.7910	0.8111	0.7815
1.5	0.7886	0.8037	0.6519
1.6	0.7953	0.8111	0.6519
1.7	0.7963	0.8222	0.6630
1.8	0.8066	0.8259	0.7778
1.9	0.8046	0.8296	0.6630
2.0	0.8019	0.8296	0.5519
2.1	0.8019	0.8148	0.6037
2.2	0.8010	0.8148	0.6185
2.3	0.8041	0.8185	0.6630
2.4	0.8043	0.8185	0.6630
2.5	0.8044	0.8185	0.7815
2.6	0.7976	0.8111	0.5519
2.7	0.8041	0.8111	0.7741
2.8	0.7988	0.8111	0.5519
2.9	0.8037	0.8111	0.6630
3.0	0.8041	0.8111	0.5778

Table 8
Average number of iterations with different number of objects, fixed number of attributes ($m = 5$) and fixed number of clusters ($k = 10$)

Number of objects (n)	500	1000	1500	2000
Average number of iterations	8.21	7.9	7.65	7.88

Table 9
Average number of iterations with different number of clusters, fixed number of attributes ($m = 5$) and fixed number of objects ($n = 1000$)

Number of clusters (k)	5	10	15	20
Average number of iterations	6.42	7.2	9.32	8.16

Table 10
Average number of iterations with different number of attributes, fixed number of clusters ($k = 10$) and fixed number of objects ($n = 1000$)

Number of attributes (m)	4	8	12	16
Average number of iterations	8.5	6.99	4.81	4.6

clusters is equal to 16, i.e., $k = 16$. From the figures, we see that the sixteen $(\lambda_{l,1}, \lambda_{l,2})$ pairs resulted from our algorithm are different from the case (i) where $\lambda_{l,1} = \lambda_{l,2} = 0.5$. In Fig. 2, we see the sixteen $(\lambda_{l,1}, \lambda_{l,2})$ pairs represented by the stars in our case are far away from the case (i) represented by the circles. This implies that the weighting effect is important in the credit card data set, thus our algorithm can significantly improve the clustering accuracy. In Fig. 2, we see that the λ pair represented by the circles lies in between the sixteen λ pairs represented by the stars. This indicates that the weighted effects becomes less significant in the Heart disease data set. Therefore, the improvement of clustering accuracy in this case may not so significant.

Next we test the effect of the index β for the attributes weights. We use different values of β and observe the effect on the clustering accuracy. In Table 7, we test both the credit card and the Heart disease data sets when the number of clusters is equal to 2. We note that the algorithm is repeated 100 times and the average clustering accuracy results are reported. For the credit card data set, the best value of β corresponding to the largest clustering accuracy is equal to 1.8. For the Heart disease data set, the best value of β corresponding to the largest clustering accuracy is equal to 1.8. Moreover, we see that for the credit card data set, the clustering accuracy is about the same for $1.4 \leqslant \beta \leqslant 2.2$, for the Heart disease data set, the clustering accuracy is about the same for $1.4 \leqslant \beta \leqslant 3.0$. Both indicate the clustering accuracy is not sensitive to a range of values of β. This is important phenomenon since it is not necessary to select the particular value of β in order to obtain good clustering accuracy results. Here we reply upon computing accuracies by considering various β values and then choosing the one that leads to the best accuracy result, i.e., $\beta = 1.8$ for both data sets.

3.3. Efficiency

The purpose of the second experiment is to test the efficiency of the attributes-weighting algorithm when clustering large data sets. In this experiment we used an artificial data set to test the efficiency of the algorithm. We see from Tables 8–10 that the average number of iterations required for convergence is at most 10. The computational complexity of the algorithm is $O(mnk)$ and therefore it is best suited for dealing with large data sets with large number of attributes.

4. Summary

In this paper we have presented the attributes-weighting algorithm. This is achieved by the development of a new procedure to generate the weight for each attribute from each cluster within the framework of the k-means-type algorithm. The main result of this paper is to provide a method to find the weight for each attribute from each cluster. Experimental results have shown the effectiveness of the new algorithm.

References

[1] M.R. Anderberg, Cluster Analysis for Applications, Academic Press, New York, 1973.
[2] G.H. Ball, D.J. Ball, A clustering technique for summarizing multivariate data, Behavioral Sci. 12 (1967) 153–155.

[3] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[4] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[5] K.C. Gowda, E. Diday, Symbolic clustering using a new dissimilarity measure, Pattern Recogn. 24 (6) (1991) 567–578.
[6] J.Z. Huang, Extensions to the *k*-means algorithm for clustering large data sets with categorical values, Data Mining Knowledge Discovery 2 (3) (1998) 283–304.
[7] J.Z. Huang, M.K. Ng, A fuzzy *k*-modes algorithm for clustering categorical data, IEEE Trans. Fuzzy Syst. 7 (4) (1999) 446–452.
[8] J.C. Bezdek, A convergence theorem for the fuzzy ISODATA clustering algorithms, IEEE Trans. Pattern Anal. Machine Intell. 2 (1980) 1–8.
[9] J.B. Macqueen, Some methods for classification and analysis of multivariate observings, in: Proceedings of the 5th Symposium on Mathematical Statistics and Probability, Vol. 7(1), Berkley, CA, 1967, pp. 281–297.
[10] M.K. Ng, J.C. Wong, Clustering categorical data sets using Tabu search techniques, Pattern Recogn. 35 (2002) 2783–2790.
[11] World Wide Web URL http://www.ncc.up.pt/liacc/ML/statlog/datasets.html.

About the Author—ELAINE Y. CHAN was born in Hong Kong, China. She received B. Sc. and M. Phil. degrees in Mathematics from the University of Hong Kong, in 2001 and 2003, respectively.

About the Author—WAI-KI CHING is a lecturer in the Department of Mathematics of the University of Hong Kong. He obtained his B. Sc. and M. Phil. degrees in Mathematics and Applied Mathematics from the University of Hong Kong in 1991 and 1994, respectively. He then obtained his Ph.D. degree in Systems Engineering and Engineering Management from the Chinese University of Hong Kong in 1998. He was a visiting post-doc fellow in the Judge Institute of Management Studies of the Cambridge University (1999–2000). Before joining his Alma Mater, he was a lecturer at the Faculty of Mathematical Studies of the University of Southampton (2000–2001). Ching's research interests are in the areas of Mathematical Modeling, Operations Research and Numerical Algorithms. He was awarded the Best Student Paper Prize in the Copper Mountain Conference (Colorado University and SIAM) U.S.A. (1998), the Outstanding Ph.D. Thesis Prize from the Chinese University of Hong Kong (1998) and the Hong Kong Croucher Foundation Fellowship (1999).

About the Author—MICHAEL K. NG was born in Hong Kong, China, in 1967. He received B. Sc. and M. Phil. degrees in Mathematics from the University of Hong Kong, in 1990 and 1993, respectively, and Ph.D. 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 currently an Associate Professor in the Department of Mathematics at the University of Hong Kong. Ng's 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.

About the author—JOSHUA HUANG, Senior Researcher and Assistant Director of E-Business Technology Institute at the University of Hong Kong. He received his Ph.D. degree from The Royal Institute of Technology in Sweden. He has many years experience in industry consulting and development of large software systems. Before joining ETI, he was a senior consultant at The Management Information Principles, Australia, consulting on data mining and business intelligence systems. He worked as a research scientist at CSIRO, Australia, and invented two influential data clustering algorithms k-modes and k-prototypes for clustering categorical data, which are widely used in research and industry applications.