Fuzzy *K*-Prototypes Algorithm for Clustering Mixed Numeric and Categorical Valued Data*

CHEN Ning[1], CHEN An[2,3], ZHOU Long-xiang[1]

[1](*Academy of Mathematics and System Sciences, The Chinese Academy of Sciences, Beijing* 100080, *China*);
[2](*Institute of Policy and Management, The Chinese Academy of Sciences, Beijing* 100080, *China*);
[3](*Institute of Software, The Chinese Academy of Sciences, Beijing* 100080, *China*)

E-mail: nchen@math08.math.ac.cn; anchen1@yahoo.com

http://www.amss.ac.cn

Received July 24, 2000; accepted January 12, 2001

Abstract: The capacity of dealing with mixed numeric and categorical valued data is undoubtedly important for clustering algorithms because there is usually a mixture of numeric and categorical valued attributes in real databases. The use of fuzzy techniques makes clustering algorithms robust against noise and missing values in the databases. In this paper, a fuzzy *k*-prototypes algorithm integrating *k*-means and *k*-modes algorithm is presented and is used to mixed databases. Experiments on several real databases demonstrate that fuzzy algorithm can get better result than the corresponding hard algorithm. Some properties of fuzzy *k*-prototypes algorithm are also discussed.

Key words: numeric attribute; categorical attribute; hard clustering; fuzzy clustering

Clustering has been discussed extensively in many areas such as similarity search, customer segmentation, pattern recognition and trend analysis. The capacity to deal with both numeric and categorical valued attributes is undoubtedly important for clustering algorithms owing to the fact that there is usually a mixture of numeric and categorical valued attributes in real databases. Although many clustering algorithms have been proposed so far, most clustering algorithms are focused on numeric data in which the distance between two objects is defined based on the inherent geometric properties of data. However, those algorithms that use distances or vector product methods are not appropriate for non-numeric valued data.

K-means algorithm is very popular and well known for clustering because of its efficiency and effectiveness. It randomly selects *k* objects as the initial centers (means) and assignments for each remaining object to its closest center. In the next step, the centers of each cluster are updated and objects are reassigned. If the clustering quality increases, the current clustering is replaced by the new clustering. This process iterates until no more quality improvement is possible. The basic *k*-means algorithm has many variations, such as alternate methods of choosing the initial centers and updating the centers. Since means only exist for numeric attributes, this clustering algorithm

* Supported by the National Natural Science Foundation of China under Grant No. 69983011 (国家自然科学基金)

CHEN Ning was born in 1974. She received her Ph. D. degree in computer software from Mathematics Institute, The Chinese Academy of Sciences in 2001. Her research interests are data mining and knowledge discovery. **CHEN An** was born in 1970. He received his Ph. D. degree from Economics and Management School, Beijing University of Aeronautics & Astronautic in 2001. His research interests are supply chain management, operational research and data mining. **ZHOU Long-xiang** was born in 1938. He is a professor and doctoral supervisor of Mathematics Institute, The Chinese Academy of Sciences. His current research interests are distributed database, multimedia database, data mining and data warehouse.

is not applicable to domains with non-numeric attributes. Huang[1,2] presents two algorithms, k-modes and k-prototypes, which extend k-means paradigm to categorical domains and domains with mixed numeric and categorical values whilst preserving its efficiency. The k-modes algorithm uses a simple matching dissimilarity measure to deal with categorical data replacing the means of clusters with modes, and uses a frequency-based method to update modes in the clustering process. The k-prototypes algorithm integrates the k-means and k-modes algorithms through the definition of a combined dissimilarity measure to allow for clustering objects described by mixed attributes.

Hard (crisp, exclusive) clustering does not allow the overlap of clusters and separates the data into disjoint groups. However, in practical situations there are many cases in which hard clustering is not suitable for natural subgroups. For example, a spatial object intersects the area of two clusters at the same time or two objects have the same distance from two clusters. In such cases, fuzzy clustering, admitting varying degrees of data membership in multiple clusters, is more appropriate than hard clustering. The membership matrix generated by fuzzy clustering can provide more information than a simple cluster identifier produced by hard clustering, which helps users to make multiple decisions such as identifying those boundary objects belonging to multiple clusters with similar degree. A pioneering work for applying the concept of fuzzy sets to a cluster analysis was done by Ruspini[3]. Since fuzzy k means clustering algorithm was proposed by Bezedek[4], several methods of fuzzy clustering have rapidly developed and many applications have been suggested in the literature[5,6]. These studies mainly focus on the mathematical research of optimality and convergence. A fuzzy k-modes algorithm as an extension of fuzzy k-means algorithm for clustering categorical data is described in Ref. [7]. It uses a simple matching dissimilarity measure for objects of categorical attributes, allowing the fuzzy k-means paradigm applicable on discrete, unordered categorical data. In this paper, we integrate the fuzzy paradigm on numeric and categorical attributes proposed in previous work and present a fuzzy k-prototype algorithm for mixed data. Fuzzy paradigm helps to improve the robustness of clustering against noise and terminate at a better result. Experimental results on some well-known real databases show that the accuracy of fuzzy clustering is better than that of the corresponding hard clustering in most cases. We give a detailed discussion about the impact of parameters on clustering accuracy and iteration number. We also discuss the evaluation of clustering quality and use it as a criterion for choosing cluster number.

The rest of the paper is organized as follows. In Section 1, we present a fuzzy k-prototype clustering algorithm for mixed data. Some experimental results are shown in Section 2 and Section 3 concludes the paper.

1 Problem Statement and Solution

Attributes can be categorized into two classes: categorical attribute and numeric attribute. Categorical attributes are fields that take on values from a limited and predetermined set of values. Usually, there is no particular order to the categories. Post code, car model, material status are examples of categorical attributes. Categorical attributes only represent to which of categories a thing belongs. Binary attributes can be regarded as special cases of categorical attributes having only two values. Numeric attributes that can be summed and sorted are very familiar in databases. Age, price, salary and temperature are examples of numeric attributes. A numeric database contains only numeric valued attributes, a categorical database contains only categorical valued attributes and a mixed database contains both numeric and categorical valued attributes.

1.1 Basic concepts

If D is the universal set from which values of a crisp set C are taken, then we can represent C as $C=\{x|x\in D, \text{and } \mathrm{Char}_C(x)=1\}$, where $\mathrm{Char}_C\rightarrow\{0,1\}$ is the characteristic function of C. For a fuzzy set F, its characteristic

function (or membership function) $\mathrm{Char}_F \to [0,1]$ takes values between 0 and 1 instead of the discrete values 0 or 1.

K-hard clustering partitions D into k disjoint groups. Each clustering corresponds to a $k \times n$ matrix U, satisfying:

(1) $w_{li} \in \{0,1\}$, i.e., the element of matrix is 0 or 1;

(2) $\sum_{l=1}^{k} w_{li} = 1$, i.e., only one 1-valued element in each column;

(3) $\sum_{i=1}^{n} w_{li} > 0$, i.e., at least one non-zero element in each row.

Since each hard clustering corresponds to a $\{0,1\}$ matrix, and each matrix satisfying the above conditions corresponds to a clustering, then

$M_{hk} = \left\{U \in V_{k \times n} \mid w_{li} \in \{0,1\},\ \forall\, l,i;\ \sum_{l=1}^{k} w_{li} = 1,\ \sum_{i=1}^{n} w_{li} > 0,\ \forall\, l \in [1,k],\ \forall\, i \in [1,n]\right\}$ is called k-hard clustering space on D.

K-fuzzy clustering partitions D into k clusters, each of which is a fuzzy set of D. Every clustering corresponds to a membership matrix U, satisfying:

(1) $w_{li} \in [0,1]$, i.e., the element of matrix is a value between 0 and 1;

(2) $\sum_{l=1}^{k} w_{li} = 1$, i.e., the sum of elements in each column equals 1;

(3) $\sum_{i=1}^{n} w_{li} > 0$, i.e., at least one non-zero element exists in each row.

Since each fuzzy clustering corresponds to a $[0,1]$ matrix, and each matrix satisfying the above conditions corresponds to a fuzzy clustering, then

$M_{fk} = \left\{U \in V_{k \times n} \mid w_{li} \in \{0,1\},\ \forall\, l,i;\ \sum_{l=1}^{k} w_{li} = 1,\ \sum_{i=1}^{n} w_{li} > 0,\ \forall\, l \in [1,k],\ \forall\, i \in [1,n]\right\}$ is called k-fuzzy clustering space on D.

In general, the process of fuzzy clustering can be divided into four phases:

1 Data normalization;

Normalization maps all the attributes to a common range (often $[0,1]$) to deal with the problem that different variables are measured in different units. After normalization, all attributes contribute equally to the distance between two objects. Some common ways of normalization include:

(1) Subtract the mean value from each attribute and then divide by the standard deviation;

(2) Divide each attribute by the mean of all the values it takes on;

(3) Divide each attribute by the range (the difference between the lowest and highest values it takes on) after subtracting the lowest value.

2 Define the dissimilarity between objects;

3 Use a clustering method to group similar objects;

4 Analyze the result.

Since the result of fuzzy clustering is a set of fuzzy sets, we usually transform them into crisp sets using some approaches such as λ-level cut and nearest maximum membership principle. The λ-level cut ($0 \leqslant \lambda \leqslant 1$) of a fuzzy set is a crisp set whose elements have a membership grade greater than or equal to λ, that is $w'_{li} = \begin{cases} 1 & w_{li} \geqslant \lambda \\ 0 & \text{otherwise} \end{cases}$. The nearest k-hard clustering of a k-fuzzy clustering is obtained by assigning each object to the cluster associated with maximum membership, that is $w'_{li} = \begin{cases} 1 & w_{li} = \max\limits_{1 \leqslant j \leqslant k} \{w_{ji}\} \\ 0 & \text{otherwise} \end{cases}$. For $\forall\, X_i \in D$, if $\max(w_{ji})$, $1 \leqslant j \leqslant k$, is not

unique, X_i is arbitrarily assigned to one cluster achieving the maximum. Thus, a fuzzy cluster C_l is transformed to a crisp cluster $C'_l=\{X_i|X_i\in D, w'_{li}=1, 1\leqslant i\leqslant n\}$. Since the first and last phases are simple, we focus on the second and third phases in this paper.

1.2 Dissimilarity measure

Let $A_1,A_2,\ldots,A_m$ be a set of attributes in a mixed database D in which $A_1,A_2,\ldots,A_p$ are numeric attributes and $A_{p+1},A_{p+2},\ldots,A_m$ are categorical attributes. The domain of A_j is denoted as $\mathrm{Dom}(A_j)$. The number of values for categorical attributes $A_j(p+1\leqslant j\leqslant m)$ is denoted as n_j. For simplicity, we assign an order to the values of categorical attributes. An object $X_i\in D$ can be represented as an m-dimensional vector $(x_{i1},x_{i2},\ldots,x_{im})$, where $x_{ij}\in \mathrm{Dom}(A_j)$.

Definition 1. Let $X_i, X_j\in D$, the dissimilarity between X_i and X_j is defined as

$$d(X_i,X_j)=d_r(X_i,X_j)+\gamma d_c(X_i,X_j)=\sum_{l=1}^{p}(x_{il}-x_{jl})^2+\gamma\sum_{l=p+1}^{m}\delta(x_{il},x_{jl}),$$

where $\delta(x_{il},x_{jl})=\begin{cases}0 & \text{if } x_{il}=x_{jl}\\ 1 & \text{otherwise}\end{cases}$.

In the above definition, $d_r(X_i,X_j)$ is the squared Euclidean distance on the numeric attributes between X_i and X_j, $d_c(X_i,X_j)$ is the simple matching dissimilarity on the categorical attributes between X_i and X_j, and γ is the weight parameter to avoid favoring either type of attributes. The dissimilarity satisfies the following properties:

(1) Reflex: $\forall X_i, X_j\in D$, $d(X_i,X_j)=0$ if and only if $x_{il}=x_{jl}$, $1\leqslant l\leqslant m$;

Proof. $d(X_i,X_j)=0\Leftrightarrow d_r(X_i,X_j)=0, d_c(X_i,X_j)=0\Leftrightarrow\sum_{l=1}^{p}(x_{il}-x_{jl})^2=0, \sum_{l=p+1}^{m}\delta(x_{il},x_{jl})=0\Leftrightarrow x_{il}=x_{jl}$, $1\leqslant l\leqslant m$.

(2) Symmetry: $\forall X_i,X_j\in D, d(X_i,X_j)=d(X_j,X_i)$;

(3) Metric Property ($p=0$): $\forall X_i,X_j,X_k\in D, d(X_i,X_j)+d(X_j,X_k)\geqslant d(X_i,X_k)$.

Proof. Assume $x_{il},x_{jl},x_{kl}\in \mathrm{Dom}(A_l)$, and A_l is a categorical attribute. If $\delta(x_{il},x_{jl})=\delta(x_{jl},x_{kl})<\delta(x_{il},x_{kl})$, then $\delta(x_{il},x_{jl})=0$, $\delta(x_{jl},x_{kl})=0$, and $\delta(x_{il},x_{kl})=1$ because the dissimilarity between two categorical values is either 0 or 1. That is, $x_{il}=x_{jl}, x_{jl}=x_{kl}$, and $x_{il}\neq x_{kl}$. On the other hand, the first two equalities imply $x_{il}=x_{kl}$ in contradiction to $x_{il}\neq x_{kl}$. So, we have $\delta(x_{il},x_{jl})+\delta(x_{jl},x_{kl})\geqslant\delta(x_{il},x_{kl})$. Thus, $d_c(X_i,X_j)+d_c(X_j,X_k)\geqslant d_c(X_i,X_k)$. Since $p=0$, i.e., all attributes are categorical attributes, $d(X_i,X_j)+d(X_j,X_k)=d_c(X_i,X_j)+d_c(X_j,X_k)\geqslant d_c(X_i,X_k)=d(X_i,X_k)$.

1.3 Fuzzy K-prototypes algorithm

Definition 2. Given a set of objects $X_1,X_2,\ldots,X_n$ described by numeric attributes $A_1,A_2,\ldots,A_p$ and categorical attributes $A_{p+1},A_{p+2},\ldots,A_m$, and the cluster number k, the problem of fuzzy clustering can be described as a mathematical program:

$$\text{Minimize } F(W,Z)=\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}^{\alpha}d(X_i,Z_l) \tag{1}$$

$$\text{Subject to } w_{li}\in[0,1],\ 1\leqslant l\leqslant k,\ 1\leqslant i\leqslant n \tag{2}$$

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

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

where $\alpha\geqslant 1$ is a fuzzy parameter, $W=\{w_{li}\}$ is a $k\times n$ membership matrix, and $Z=\{Z_1,Z_2,\ldots,Z_k\}$ is a set of k m-dimensional prototypes.

The state of clustering is expressed by the membership matrix W, implying the degree of belongingness of each object to a cluster. The fuzzy parameter α plays a role in determining the degree of fuzziness of the clusters.

$$\text{If } a>1\text{: } w_{li}=\begin{cases}1 & \text{if } X_i=Z_l^*\\ 0 & \text{if } X_i=Z_j^*,\ j\neq l\\ \dfrac{1}{\sum_{j=1}^{k}\left(\dfrac{d(X_i,Z_l^*)}{d(X_i,Z_j^*)}\right)^{\frac{1}{a-1}}} & \text{otherwise}\end{cases}$$

Theorem 3 can be proved using the similar method as that in Refs. [9,10] because the cost function only depends on W as Z^* is fixed.

Theorem 4. (Prototype update method) Consider the problem $F_2(W^*,Z)$ where W^* is fixed, then $F_2(W^*,Z)=\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}^{*a}d(X_i,Z_l)$ is minimized if and only if $Z_l(1\leqslant l\leqslant m)$ is assigned as follows:

For numeric attribute $A_j(1\leqslant j\leqslant p)$, $Z_{lj}=\dfrac{\sum_{i=1}^{n}w_{li}^{*a}X_{ij}}{\sum_{i=1}^{n}w_{li}^{*a}}$;

For categorical attribute $A_j(p+1\leqslant j\leqslant m)$, $Z_{lj}=a_j^{(r)}\in \mathrm{Dom}(A_j)$ where

$$\sum_{i=1}^{n}(w_{li}^{*a}\mid x_{ij}=a_j^{(r)})\geqslant\sum_{i=1}^{n}(w_{li}^{*a}\mid x_{ij}=a_j^{(t)}),\ 1\leqslant t\leqslant n_j.$$

Proof.

$$F_2(W^*,Z)=\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}^{*a}d(X_i,Z_l)=\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}^{*a}(d_r(X_i,Z_l)+\gamma d_c(X_i,Z_l))=$$
$$\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}^{*a}d_r(X_i,Z)+\gamma\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}^{*a}d_c(X_i,Z_l)$$

We denote $F_r(W^*,Z)=\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}^{*a}d_r(X_i,Z_l)$ and $F_c(W^*,Z)=\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}^{*a}d_c(X_i,Z_l)$. Thus, $F_2(W^*,Z)=F_r(W^*,Z)+F_c(W^*,Z)$. Since $F_r(W^*,Z)$, $F_c(W^*,Z)$ are nonnegative and independent of each other, minimizing $F_2(W^*,Z)$ is equivalent to minimizing $F_r(W^*,Z)$ and $F_c(W^*,Z)$ simultaneously. According to Theorem 2 and Theorem 4[7], the result is obvious.

If there is more than one value for a categorical attribute A_j having the maximum membership sum, we set Z_{lj} as the first one according to the order we assign to the domain of A_j. In general, fuzzy k-prototypes algorithm could be described as follows:

(1) Choose a set of initial cluster prototypes $Z^{(1)}$ and a controller of iteration ε;

(2) Determine $W^{(1)}$ that minimizes $F(W,Z^{(1)})$. Set $i=1$.

(3) Determine $Z^{(i+1)}$ that minimizes $F(W^{(i)},Z)$. If $|F(W^{(i)},Z^{(i+1)})-F(W^{(i)},Z^{(i)})|<\varepsilon$, then stop. Otherwise set $(W^*,Z^*)=(W^{(i)},Z^{(i+1)})$.

(4) Determine $W^{(i+1)}$ that minimizes $\mathbf{F}(\mathbf{W},Z^{(i+1)})$. If $|F(W^{(i+1)},Z^{(i+1)})-F(W^{(i)},Z^{(i+1)})|<\varepsilon$, then stop. Otherwise set $(W^*,Z^*)=(W^{(i+1)},Z^{(i+1)})$. Increase i by 1 and go to (3).

Let k be the number of clusters, m the number of attributes, and n the number of objects. The space complexity of fuzzy k-prototypes algorithm is $O(mn+km+kn+kM)$ to store $n\times m$ object matrix, $k\times n$ membership matrix, $k\times m$ prototype matrix and the sum of numeric values or membership sum of categorical values in memory, where $M=p+\sum_{i=p+1}^{m}n_i$. Hence, fuzzy k-prototypes algorithm is feasible for moderate databases with small number of diverse values on categorical attributes. If memory space is enough, the time complexity of fuzzy k-prototypes algorithm is $O(kmn)$ which is much faster than other clustering algorithms.

Theorem 5. Hard k-prototypes algorithm converges to a partial optimal solution in a finite number of iterations.

Proof. Assume $W^{(r)}$ is the membership matrix in iteration r, and $Z^{(r+1)}$ is the optimal solution obtained in iteration $r+1$. Similarly, $Z^{(r+1)}(t>r)$ is the optimal solution obtained in $t+1$ iteration at fixed $W^{(r)}$. If $W^{(r)}=W^{(t)}$, then $Z^{(r+1)}=Z^{(t+1)}$ according to the prototype update method in Theorem 4. Hence, $F(W^{(r)},Z^{(r+1)})=F(W^{(t)},Z^{(t+1)})$. On the other hand, the sequence $F(W,Z)$ generated by the algorithm is strictly decreasing. Hence, the algorithm tests each membership matrix at most once. In hard algorithm, there is a finite number of W since each element of W is either 0 or 1. Then the algorithm will reach a partial solution in a finite number of iterations.

Theorem 6. Assume (W^*,Z^*), (W',Z') are the global optimal solution of $F_{\alpha_1}(W,Z)$ and $F_{\alpha_2}(W,Z)$ respectively. If $\alpha_1>\alpha_2\geqslant 1$, then $F_{\alpha_2}(W',Z')>F_{\alpha_1}(W^*,Z^*)$.

Proof. Since $\forall\ 1\leqslant l\leqslant k$, $1\leqslant i\leqslant n$, $w_{il}\in[0,1]$, $w_{il}^{\alpha_2}\geqslant w_{il}^{\alpha_1}$ for $\alpha_1>\alpha_2\geqslant 1$ and $\exists\ w_{il}\in(0,1)$, such that $w_{il}^{\alpha_2}>w_{il}^{\alpha_1}$. $F_{\alpha_2}(W',Z')=\sum_{l=1}^{k}\sum_{i=1}^{n}w_{il}'^{\alpha_2}d(X_i,Z_l')>\sum_{l=1}^{k}\sum_{i=1}^{n}w_{il}'^{\alpha_1}d(X_i,Z_l')=F_{\alpha_1}(W',Z')$. Since (W^*,Z^*) is the global optimal of $F_{\alpha_1}(W,Z)$, $F_{\alpha_1}(W',Z')\geqslant F_{\alpha_1}(W^*,Z^*)$. Hence, $F_{\alpha_2}(W',Z')>F_{\alpha_1}(W^*,Z^*)$.

Lemma. If $\alpha_1>\alpha_2\geqslant 1$, then $\forall\ (W,Z)$, $F_{\alpha_2}(W,Z)>F_{\alpha_1}(W^*,Z^*)$, where (W^*,Z^*) is the global optimal solution of $F_{\alpha_1}(W,Z)$.

Proof. Let (W',Z') be the global optimal solution of $F_{\alpha_2}(W,Z)$. We have $F_{\alpha_2}(W',Z')>F_{\alpha_1}(W^*,Z^*)$ according to Theorem 6. Since (W',Z') is the global optimal of $F_{\alpha_2}(W,Z)$, we have $F_{\alpha_2}(W,Z)\geqslant F_{\alpha_2}(W',Z')$. Hence, $F_{\alpha_2}(W,Z)>F_{\alpha_1}(W^*,Z^*)$.

2 Experimental Results

To evaluate the effectiveness of fuzzy k-prototypes algorithm and discuss the impact of parameters, we perform some experiments on several real databases including one numeric database, one categorical database and two mixed databases[11].

2.1 Real databases

Soybean database (DS1) has 35 categorical-valued attributes and 47 instances. The instances are divided into 4 classes, containing 10, 10, 10, and 17 members respectively. Protein localization database (DS2) consists of 336 instances and 7 numeric valued attributes. 8 classes are labeled containing 143,77,52,35,20,4,2,2 instances respectively. Flag database (DS3) contains details of various nations and their flags. With these data we can try to predict the religion of a country from its size and the colors in its flag. There are 194 instances and 30 attributes: 10 attributes are numeric valued, and the remainder is categorical valued. Credit approval database (DS4) concerns credit card applications, described by a good mixture of attributes—numeric with small number of values, and categorical with large number of values. The database has 15 attributes and 690 instances classified into two classes. There are also 5% missing values on both numeric and categorical attributes in the database. DS1 and DS4 have been used in Refs. [1,2]. The characteristic of these databases is listed in Table 1.

Table 1 Description of datasets

	Type	Number of attributes		Number of objects	Number of clusters	Missing data
		Categorical	Numeric			
DS1	Categorical	35	0	47	4	No
DS2	Numeric	0	7	336	8	No
DS3	Mixed	20	10	194	8	No
DS4	Mixed	9	6	690	2	Yes

2.2 Accuracy of clustering results

Result table is used to test the accuracy of clustering with respect to databases in which the classifier of each object is predetermined. Assume the set of input clusters is $\{C_1,C_2,\ldots,C_k\}$, and the set of output clusters is

$\{C'_1, C'_2, \ldots, C'_k\}$. In result table, each row corresponds to one input cluster, and each column corresponds to one output cluster. The element a_{ij} corresponding to the ith row and the jth column represents the number of objects of C_i which are classified as C'_j. For the ith row, if a_{ij} is clearly much larger than a_{im}, $m \neq j$, it indicates that most objects of C_i get into C'_j, with limited number of exceptions distributed to other output clusters. Then we map C_i to C'_j, denoted as $f(i)=j$. After mapping all input and output clusters one by one, the accuracy of clustering result is defined as the fraction of correctly clustered points: $(\sum_{i=1}^{k} a_{i,f(i)})/n$. The larger the value is, the fitter the clustering result is with respect to the database.

2.3 Effectiveness

In the first experiment, we carry out a group of tests on DS1 using different values of α. For each value of α, the algorithm is run for 100 times, each randomly choosing k initial prototypes. The results demonstrate that fuzzy k-prototypes algorithm is better than hard algorithm in 67 runs. The results of 28 runs have no improvement and only 5% are worse than those of hard algorithm result. In the second case, the accuracy of 23 runs is at least 95% using hard algorithm so that fuzzy paradigm has no improvement on them. We also notice the decrease of accuracy in the third case is limited while the improvement in the first case is significant especially at bad initial choice. We list 5 examples belonging to three cases respectively in Table 2. It is obvious that fuzzy algorithm is superior to corresponding hard algorithm on clustering accuracy.

Table 2 Accuracy comparison

Decrease cases		Improvement cases		No-change cases	
Hard	Fuzzy	Hard	Fuzzy	Hard	Fuzzy
1	96%	64%	94%	1	1
98%	96%	30%	87%	96%	96%
55%	51%	55%	91%	98%	98%
70%	68%	51%	72%	68%	68%
74%	68%	34%	89%	72%	72%

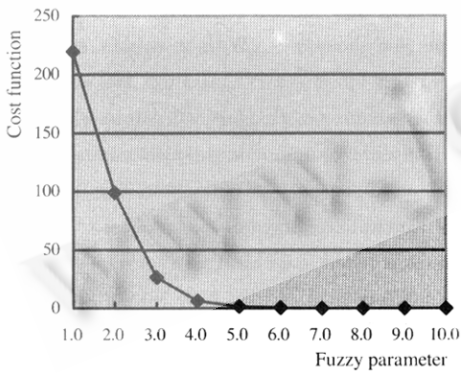

Fig. 1 Cost function vs fuzzy parameter (DS1)

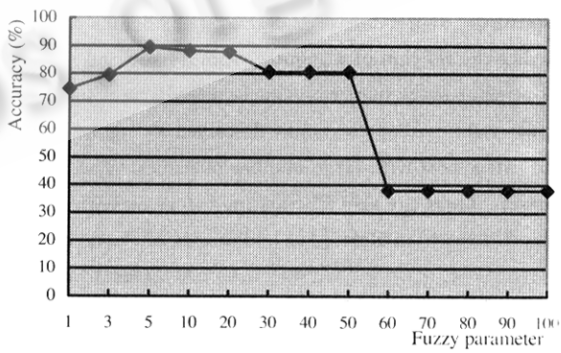

Fig. 2 Accuracy vs fuzzy parameter (DS1)

As proved in Theorem 6 and shown in Fig. 1, the cost function decreases dramatically as α increases. Figure 2 shows the average accuracy of clustering results with respect to variant α from 1.0 to 100. The average accuracy of clustering results increases as α increases at first until it reaches a peak point and then begins to decrease, but still remaining superior to hard algorithm. There is a dramatic drop at $\alpha=50$ and reaches the lowest point at $\alpha=60$ after which it remains unchanged. The reason is that $\lim_{\alpha \to +\infty} \frac{1}{\alpha-1}=0$ then $\lim_{\alpha \to +\infty} w_{li}=\frac{1}{k}$, i.e., the difference of member-

ship of an object to all clusters tends to become zero when α is large enough unless it is equal to one of the prototypes. In fact we find the maximum membership of most objects is very close to $1/k$ when α is more than 20. In such case, most objects have similar membership to all prototypes so that the dissimilarity loses its meaning. At large α, the algorithm converges rapidly to a local optimal solution and has no opportunity to find a better solution. An enlargement of Fig. 2 for α changing from 1.0 to 8.0 is shown in Fig. 3. It shows fuzzy algorithm reaches the best average accuracy at $\alpha=4.0$. We also notice from Fig. 4 the iteration number decreases as α increases. Fig. 5 depicts the distribution of accuracy at different fuzzy parameters for 100 runs on DS1. Fig. 5 describes the details of accuracy at different fuzzy parameters for 100 runs on DS3.

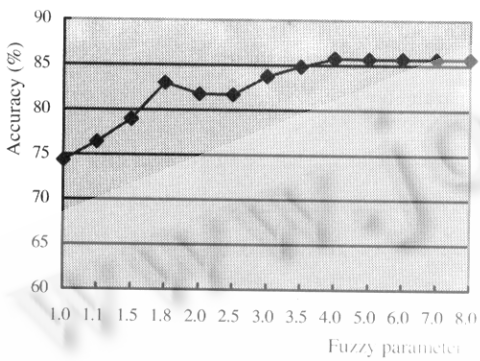

Fig. 3 Accuracy vs fuzzy parameter (DS1)

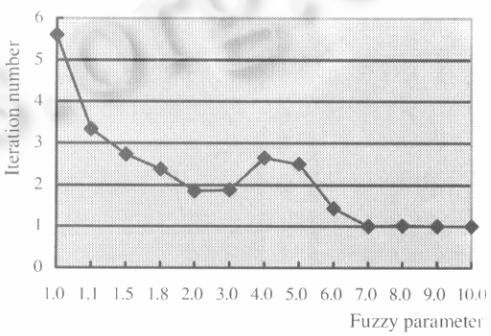

Fig. 4 Iteration number vs fuzzy parameter (DS1)

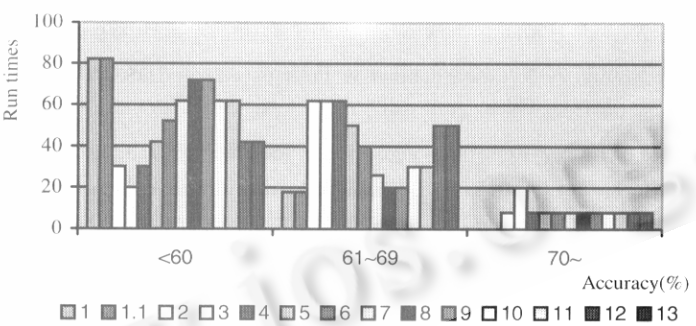

Fig. 5 Accuracy vs fuzzy parameter (DS3)

2.4 Effect of cluster size

As we know, k-means algorithms work well on clusters of spherical shape and similar size because they represent a cluster as its mean and assign an object to its nearest mean. For clusters of variant sizes, they may not distinguish small and big clusters by splitting the bigger one while merging the smaller ones, which decreases the accuracy of clustering. Is fuzzy k-prototypes algorithm sensitive to the variance of cluster size as k-means algorithms? In order to test the impact of cluster sizes on the accuracy of clustering, we generate a subset database by extracting 20 instances from 5 big clusters respectively from DS2. As the first experiment, we run 100 times on the original database and subset database, using a set of random initial prototypes in each run. The results in Fig. 6 show that the average accuracy of clustering on subset database is better than that of the original database at each parameter. Figure 7 shows the change of iteration number with respect to α.

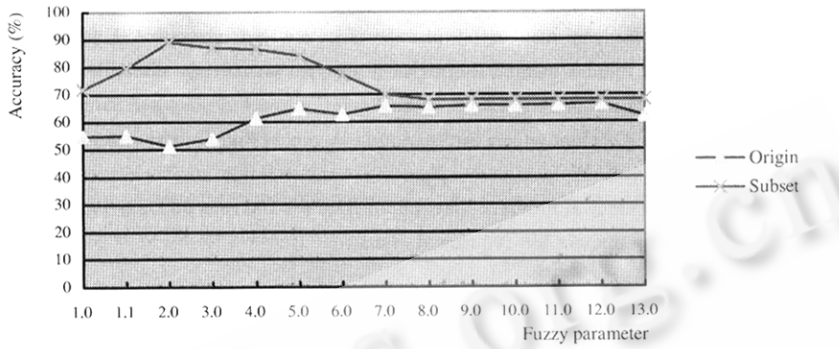

Fig. 6 Accuracy vs fuzzy parameter (DS2)

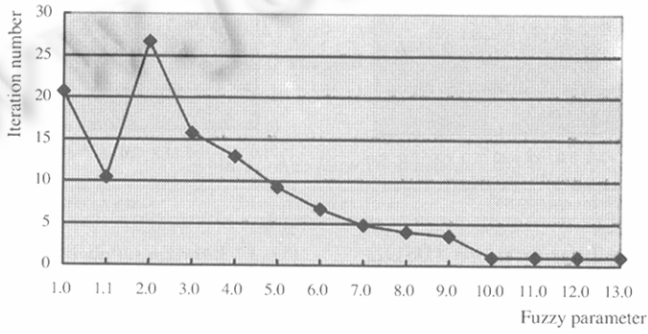

Fig. 7 Iteration number vs fuzzy parameter (DS2)

2.5 Effect of weight parameter

An important advantage of fuzzy k-prototypes algorithm is the ability of dealing with mixed numeric and categorical valued data. We perform the algorithm on DS3 and DS4 using different values of γ. For the two databases, we first normalize the numeric attributes to the range of $[0,1]$. From Figs. 8 and 9, we find the algorithm gets the best accuracy when $\gamma=1.1$ for both databases.

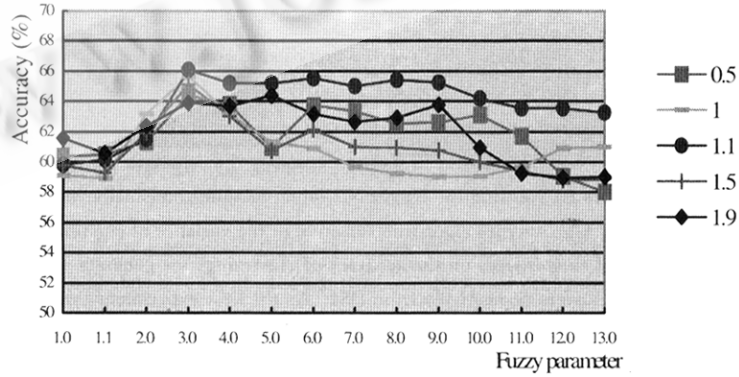

Fig. 8 Accuracy vs fuzzy parameter (DSs)

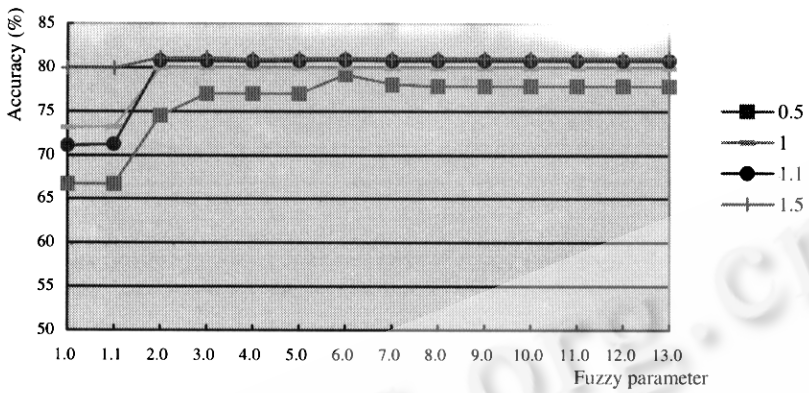

Fig. 9 Accuracy vs fuzzy parameter (DS4)

2.6 Dealing with missing data

In real database, there are usually some missing data because some data are not captured or available. In Ref. [1], the missing data on categorical attributes take part in the computation of membership matrix as a value of the domain, which is reasonable when the number of missing values on one attribute is much lower than other values. And it can not deal with missing data on numeric attributes. In our algorithm, we adopt three strategies to deal with missing data.

Strategy 1. Delete the objects containing missing values on at least one attribute and perform clustering on the remainder objects.

Strategy 2. Replace missing values on categorical attributes with the value which appears most frequently in the overall database, and replace missing values on numeric attributes with the average value.

Strategy 3. Regard the missing value as a special value. In the calculation of membership matrix, missing values represented by '?' are regarded as equal to any other value of the attribute, i. e., $\begin{cases}\delta(x,?)=0 & \forall\ x\in \mathrm{Dom}(A_j),\ p+1\leqslant j\leqslant m\\ d(x,?)=0, & \forall\ x\in \mathrm{Dom}(A_j),\ 1\leqslant j\leqslant p\end{cases}$. The objects containing missing values on A_j as well as their memberships do not contribute to the computation of prototypes on A_j.

The first two strategies can be accomplished in a pre-processing step before the clustering procedure, and the third strategy is implemented during the clustering procedure. We carry out our algorithm on DS4 using the three strategies respectively at $\gamma=1.1$ and compare their performances. In the first method, we get 653 instances after deleting the instances containing missing value. As shown in Fig. 10, the third strategy has the best accuracy compared with the other two strategies. Besides, it is the most efficient one because the extra preprocessing is saved. Strategy 1 gets the worst result because it keeps the objects out of the computation even if missing data exist in only one attribute.

2.7 Quality evaluation

The purpose of clustering is to group similar objects into clusters and separate dissimilar objects into different clusters. We give a criterion to evaluate the quality of clustering.

Definition 3. Let $C=\{C_1,C_2,\ldots,C_k\}$ be a clustering of a data set D. The quality of a cluster C_l is defined as the average membership of objects which belong to it with the maximum degree, that is $\mathrm{Qua}(C_l)=\frac{\sum_{i=1}^{n}(w_{li}\,|\,w_{li}\geqslant w_{hi},\ 1\leqslant h\leqslant k)}{|\{x_i\,|\,w_{li}\geqslant w_{hi},\ 1\leqslant h\leqslant k,\ 1\leqslant i\leqslant n\}|}$. Quality of C is defined as the average quality of all clusters, i.e. $\mathrm{Qua}(C)=$

$\sum_{i=1}^{k}\text{Qua}(C_i)/k$.

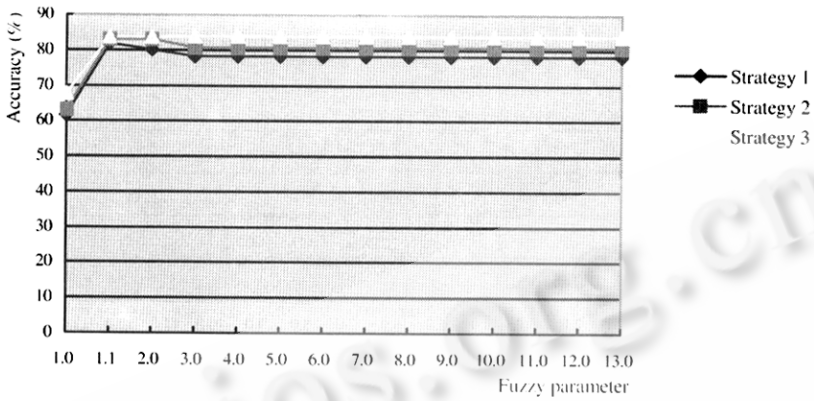

Fig. 10 Accuracy vs fuzzy parameter (DS4)

Since $\sum_{i=1}^{n}(w_{li}|w_{li}\geqslant w_{hi},\ 1\leqslant h\leqslant k)\leqslant|\{x_i|w_{li}\geqslant w_{hi},1\leqslant h\leqslant k,\ 1\leqslant i\leqslant n\}|$, we have $0\leqslant\text{Qua}(C_l)\leqslant 1$, then $0\leqslant\text{Qua}(C)\leqslant 1$. The larger the quality of a cluster is, the nearer is an object to the cluster prototype it belongs to, and farther to other prototypes. Similarly, the larger the quality of a clustering is, the more compact and separated the set of clusters is, indicating the better clustering result. The quality can be used as the criteria to determine the appropriate numbers of clusters.

3 Conclusion

Fuzzy set theory in particular is more and more frequently used in expert systems because of its simplicity and similarity to human reasoning. In this paper, we introduce a fuzzy k-prototypes algorithm combining the fuzzy paradigm on both numeric and categorical valued data. Experiments on real databases have shown the fuzzy algorithm is more effective on discovering clusters than hard algorithm. The contributions of the paper include:

- Present a fuzzy algorithm capable of dealing with mixed numeric and categorical valued data;
- Discuss the impact of fuzzy parameter and initial prototypes on clustering accuracy;
- Evaluate the quality of clustering results and discuss the relation between quality and cluster number;
- Extend the algorithm to handle missing data.

In future work, we will try to give a proof of the convergence in finite iterations for fuzzy k-prototypes algorithm. In our algorithm, γ acts equally on all categorical valued attributes. Since attributes in databases usually have different relevances, we also suggest to give different weights for each attribute and extend the fuzzy paradigm to weighted dissimilarity function.

Acknowledgments We would like to thank Dr. Zhexue Huang in the University of Hong Kong for his warm instruction and providing the code of k-prototypes algorithm.

References:

[1] Huang, Zhe-xue. Clustering large data sets with mixed numeric and categorical values. In: Lu Hong-jun, Motoda, Hiroshi, Liu Huan, eds. Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery & Data Mining. Singapore: World Scientific, 1997. 21~34.

[2] Huang, Zhe-xue. Extensions to the k-means algorithms for clustering large data sets with categorical values. Data Mining and Knowledge Discovery, 1998,2:283~304.

[3] Ruspini, E. H. A new approach to clustering. Information Control, 1969,(19):22~32.

[4] Bezedek, J. C. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, 1987.

[5] Dave, R. N., Bhaswan, K. Adaptive fuzzy C-shells clustering and detection of ellipses. IEEE Transactions on Neural Networks, 1992,3:643~662.

[6] Hathaway, R. J., Bezedek, J. C., Tucker, W. T. An improved convergence theory for the Fuzzy C-means clustering algorithm. In: Bezedek, J. C., ed. The Analysis of Fuzzy Information. Boca Raton: CRC Press, 1986.

[7] Huang, Zhe-xue, Ng, M. K. A fuzzy K-modes algorithm for clustering categorical data. IEEE Transactions on Fuzzy Systems, 1999,7(4):446~452.

[8] Bezedek, J. C. A convergence theorem for the fuzzy ISODATA clustering algorithms. IEEE Transactions on Pattern Analysis Machine Intelligence, 1980,PAMI-2:1~8.

[9] Ismail, M. A. Fuzzy C-means: optimality of solutions and effective termination of the algorithm. Pattern Recognition, 1986,19(6):481~485.

[10] Hathaway, R. J. Local convergence of the fuzzy C-means algorithms. Pattern Recognition, 1983,19(6):177~180.

[11] Blake, C. L., Merz, C. J. UCI Repository of machine learning databases. Department of Information and Computer Science. University of California, Irvine, CA, 1980, http://www.ics.uci.edu/~mlearn/MLRespository.html.

数值型和分类型混合数据的模糊 K-Prototypes 聚类算法

陈 宁[1], 陈 安[2,3], 周龙骧[1]

[1](中国科学院 数学与系统科学研究院,北京 100080);

[2](中国科学院 科技政策与管理科学研究所,北京 100080);

[3](中国科学院 软件研究所,北京 100080)

摘要: 由于数据库经常同时包含数值型和分类型的属性,因此研究能够处理混合型数据的聚类算法无疑是很重要的.讨论了混合型数据的聚类问题,提出了一种模糊 K-prototypes 算法.该算法融合了 K-means 和 K-modes 对数值型和分类型数据的处理方法,能够处理混合类型的数据.模糊技术体现聚类的边界特征,更适合处理含有噪声和缺失数据的数据库.实验结果显示,模糊算法比相应的确定算法得到的结果准确度高.

关键词: 数值型属性;分类型属性;确定聚类;模糊聚类

中图法分类号: TP311 **文献标识码**: A