结构化模糊 K-prototypes 聚类算法*)

汪加才[1] 文巨峰[1] 陈 奇[2] 俞瑞钊[2]

(南京审计学院计算机科学与技术系 南京210029)[1] (浙江大学人工智能研究所 杭州310027)[2]

摘 要 尽管综合了 K-means 和 K-modes 的 K-prototypes 算法已能有效地处理符号数据,但用聚类中的符号模(modes)来表示聚类中的数据均值将引起大量的信息丢失。为此,本文提出了一种适合于混合类型数据的结构化模糊 K-prototypes 算法(SFKP),在不增加时空开销的情况下提高聚类能力。实际数据集上的实验结果显示,SFKP 算法能够进行更加有效的聚类。

关键词 混合类型数据,模糊聚类算法,数据挖掘

A Structural Fuzzy K-prototypes Clustering Algorithm

WANG Jia-Cai[1] WEN Ju-Feng[1] CHEN Qi[2] YU Rui-Zhao[2]

(Department of Computer Science and Technology, Nanjing Audit University, Nanjing 210029)[1]

(Artificial Intelligence Institute, Zhejiang University, Hangzhou 310027)[2]

Abstract Although K-prototypes algorithm integrating K-means and K-modes algorithms has removed the numeric-only limitation of the K-means algorithm and enable it to be used to efficiently cluster large categorical data sets, the fact that replacing the means of clusters with the frequency-based modes will cause the lose of information in clusters. In this paper, a structural fuzzy K-prototypes algorithm for clustering mixed-type databases is presented and can enhance the clustering ability without increasing computational cost and memory storage. Experiments on several real databases show that the structural K-prototypes algorithm can get better clustering result than the corresponding non-structural algorithm.

Keywords Mixed numeric and nominal data, Fuzzy clustering algorithm, Data mining

1 前言

作为统计学的一个分支,聚类分析已有多年的研究历史。这些研究主要集中于基于距离的聚类分析方面,典型代表是 K-means[1]、模糊 K-means(FCM)[2]、ISODATA[1,3]等算法。在这类算法中,相似或不相似的度量是由数据对象描述属性的取值来确定的,通常利用各对象间的距离来描述。K-means 系列算法以其实现简单、效率高、适合于大数据集而广受欢迎,但它们的不足是假设描述事物的属性为有序的数值类型。

在实际应用领域中,人们经常需要处理如性别、颜色、形状、疾病类型等无顺序的符号类型数据,或者是既有数值型又有符号型的混合数据。相对于存在大量数值属性聚类方法,能有效处理符号属性数据或数值和符号混合型数据的聚类算法则较少。

一般地,可以将面向符号型数据的聚类方法分为三类。第一种是直接将符号型数据编码为有序的整数值并应用于对象间的距离计算中。多数情况下,这种转换是不合理的,所得到的距离值也难于解释。第二种是对数值型数据进行离散化,将混合类型数据统一为符号数据后再使用符号聚类算法[4]。这种方法的缺点是在离散化过程中容易造成重要信息的丢失。第三种是设计一种能适合于数值型数据和符号型数据的基于概率分布函数的评价函数[5]。对于符号属性数据,可以通过简单的计数方式来估计概率分布,而对于数值型数据则十分困难。

为了能够对符号数据进行 K-means 聚类, Ralambondrainy[6]提出了一种称为概念 K-means 聚类算法,即先将符号属性转化为多个取值为0或1的二值属性,然后将这些0或1看作一般的数值数据进行聚类。该方法尽管可以在转化的数据集上用 K-means 算法进行聚类,但对于具有较多取值可能的属性而言,因会产生大量的二值属性而使计算代价和存储代价过大,同时还加剧了"维度灾难"问题。

为此,Huang 提出了 K-modes 算法[7]和模糊 K-modes 算法[8]。K-modes 算法用模(mode)来替换聚类中心,采用符号匹配的差异性计算方法来处理符号量,以及利用基于频率方法对各聚类模进行更新。K-prototypes(KP)算法[7,8]是 K-means 和 K-modes 算法的结合,可以对采用数值量和符号量混合描述的对象进行聚类分析。虽然这些算法既解决了符号型数据的聚类问题,又不会增加计算和存储复杂度,但其缺点也是明显的。对于一个类内的每一符号属性,有时无法用单一模来表示类内所有对象在该属性上的统计信息,K-modes 算法是以丢失其它符号值的统计信息作为代价的。

本文提出的结构化模糊 KP(SFKP)算法是 KP 算法的扩展,利用了作者在文[10]中所提出的结构向量和广义重叠距离(generalized overlap distance)的概念,可在不增加时空开销的情况下提高聚类能力。本文第2节介绍基本概念及 SFKP 算法,第3、4节是算法的实现、复杂度分析及实验结果,最后是结论。

2 基本概念与结构化模糊 K-prototypes 算法

定义1 数据库表 T 是由 n 个属性 $A_1,\cdots,A_n$ 描述的一组待聚类对象集 X,$X=\{x_1,x_2,\cdots,x_m\}$。对象 x_i 表示为$(x_{i,1},\cdots,x_{i,n})$。将 n 个属性根据其取值的不同分为数值型和符号型两类。不失一般性,设前者有 p 个,后者有 $n-p$ 个。属性 A_j 的值域记为 $DOM(A_j)$。对于$1\leqslant j\leqslant p$,$DOM(A_j)=[0,1]$;对

*)基金项目:江苏省高校自然科学研究计划项目(编号:03KJB520054)。汪加才 副教授,博士,主要研究数据挖掘,商业智能等。

于 $p+1\leqslant j\leqslant n$，$DOM(A_j)=\{1,2,\cdots,n_j\}$，其中 $n_j=|DOM(A_j)|$，为属性 A_j 符号值的个数。

定义2 对象 $x\in X$ 所对应的结构向量 Sx 为将 x 中的符号属性 $A_j(q+1\leqslant j\leqslant n, p+1\leqslant q\leqslant n)$ 替换成长度为 n_j 的0/1向量，$Sx(Sx_1,\cdots,Sx_p,\cdots,Sx_q,\cdots,Sx_n)=(Sx_1,\cdots,\{Sx_{q+1}^1,\cdots,Sx_{q+1}^{n_{q+1}}\},\cdots,\{Sx_n^1,\cdots Sx_n^{n_n},\})$。即：$\forall\ j, 1\leqslant j\leqslant q: Sx_j=x_j$；$\forall\ j,t,q+1\leqslant j\leqslant n, 1\leqslant t\leqslant n_j$：若 $t\neq x_j$ 则 $Sx_j^t=0$，否则 $Sx_j^t=1$。

定义3 对象集 X 的第 $k(1\leqslant k\leqslant K)$ 个聚类的结构中心向量 $SZ_k(SZ_{k,1},SZ_{k,2},\cdots,SZ_{k,q},\cdots SZ_{k,n})=(SZ_{k,1},\cdots,SZ_{k,q},\{SZ_{k,q+1}^1,\cdots,SZ_{k,q+1}^{n_{q+1}}\},\cdots,\{SZ_{k,n}^1,\cdots,SZ_{k,n}^{n_n}\})$，其中：$\forall\ j, 1\leqslant j\leqslant q: SZ_{k,j}\in DOM(A_j)$；$\forall\ j,t,q+1\leqslant j\leqslant n, 1\leqslant t\leqslant n_j: SZ_{k,j}^t\in[0,1]$ 并且满足 $\sum_{t=1\cdots n_j}SZ_{k,j}^t=1$。

结构中心向量 SZ 中允许存在未经转换的符号属性 $A_{p+1},\cdots,A_q$，这可以看作为对 KP 算法的兼容。

定义4 结构向量(包括结构中心向量) Su,Sv 间的距离函数定义为：$d(Su,Sv)=d_r(Su,Sv)+\gamma d_o(Su,Sv)+\rho d_t(Su,Sv)=\sqrt{\sum_{j=1}^{p}(Su_j-Sv_j)^2}+\gamma\sum_{j=p+1}^{q}\delta_o(Su_j,Sv_j)+\rho\sum_{j=q+1}^{n}\delta_t(Su_j,Sv_j)$。其中：$\delta_t(Su_j,Sv_j)=\max_{t=1,\cdots n_j}\{|Su_j^t-Sv_j^t|\}$，$q+1\leqslant j\leqslant n$。

定义中，$d_r(\cdot)$ 为向量间在数值属性上的欧氏距离；$d_o(\cdot)$ 为在符号属性上的重叠(overlap)距离[9]：对于符号属性 A_j，当 $Su_j=Sv_j$ 时 $\delta_o(Su_j,Sv_j)=1$，否则为0；$d_t(\cdot)$ 为在符号属性上的结构距离分量；ρ 的作用与 γ 类似，为一调节结构距离分量的权值因子。

性质1[10] 对象 $x\in X$ 与结构中心向量 SZ_k 间的结构距离分量 $d_t(Sx,SZ_k)=\sum_{j=q+1}^{n}(1-SZ_{k,j}^{x_j})$。

性质1说明在计算各对象与聚类中心的距离时，对每个符号属性，不管符号值的多少，仅需一次减法运算即可求出其距离分量；同时，在聚类时也不需要真的将对象的符号属性转化为0/1向量。这有效地避免了传统概念 K-means 算法时空开销大的缺点。

性质2[10] $d_t(Su,Sv)$ 满足如下性质：

(1) $d_t(Su,Sv)\geqslant 0$

(2) $d_t(Su,Sv)=d_t(Sv,Su)$

(3) $d_t(Su,Sv)+d_t(Sv,Sw)\geqslant d_t(Su,Sw)$

定义5 设 X 为定义1中的待聚类对象集，结构化 KP 的硬性聚类算法和模糊聚类算法(SHKP/SFKP)是将 X 划分为 K 个分类，并使(1)式的成本函数最小。

$$F(W,SZ)=\sum_{k=1}^{K}\sum_{i=1}^{m}W_{k,i}^{\alpha}d(SZ_k,x_i) \tag{1}$$

满足：$0\leqslant W_{k,i}\leqslant 1, 1\leqslant k\leqslant K, 1\leqslant i\leqslant m$ (2)

$$\sum_{k=1}^{K}W_{k,i}=1, 1\leqslant i\leqslant m \tag{3}$$

$$0<\sum_{i=1}^{m}W_{k,i}<m, 1\leqslant k\leqslant K \tag{4}$$

式中，$K(\leqslant m)$ 是已知的目标聚类数；$\alpha\in[1,\infty]$ 为一权指数；SZ 为聚类中心矩阵，$SZ=[SZ_1,SZ_2,\cdots,SZ_K]$，$\forall\ k: 1\leqslant k\leqslant K$，$j: 1\leqslant j\leqslant n, Z_{k,j}\in DOM(A_j)$；$W=[W_{k,i}]$ 是一 $K\times m$ 的实数关联矩阵，表示对象 x_i 对聚类中心 SZ_k 的隶属度。

定理1 设成本函数 $F(W,SZ)$ 中的 W 固定为 W^*，则满足(2)，(3)，(4)式的 SZ：$SZ^*=\min_{SZ}F(W^*,SZ)$。SZ^* 的计算方法是，$\forall\ k, 1\leqslant k\leqslant K$：

(a) $\forall\ j, 1\leqslant j\leqslant p: SZ_{k,j}^*=\sum_{i=1}^{m}W_{k,i}^{\alpha}Sx_{i,j}/\sum_{i=1}^{m}W_{k,i}^{\alpha}$

(b) $\forall\ j, p+1\leqslant j\leqslant q: SZ_{k,j}^*=r\in DOM(A_j)$，$r$ 满足：

$$\sum_{i=1}^{m}(W_{k,i}^{\alpha}|Sx_{i,j}=r)\geqslant\sum_{i=1}^{m}((W_{k,i}^{\alpha}|Sx_{i,j}=t), 1\leqslant t\leqslant n_j;$$

(c) $\forall\ j, q+1\leqslant j\leqslant n, S=\{\mathrm{argmax}_{t\in DOM(A_j)}\sum_{i,Sx_{i,j}=t}W_{k,i}^{\alpha}\}$：$SZ_{k,j}^r=1/|S|$，$r\in S$；$SZ_{k,j}^t=0, t\in DOM(A_j)-S$。

证明：成本函数 $F(W^*,SZ)=\sum_{k=1}^{K}\sum_{i=1}^{m}W_{k,i}^{\alpha}d(SZ_k,Sx_i)=\sum_{k=1}^{K}\sum_{i=1}^{m}W_{k,i}^{\alpha}(d_r(SZ_k,Sx_i)+\gamma d_o(SZ_k,Sx_i)+\rho d_t(SZ_k,Sx_i))=F_r(W^*,SZ)+\gamma F_o(W^*,SZ)+\rho F_t(W^*,SZ)$。由于 $F_r(W^*,SZ)$，$F_o(W^*,SZ)$，$F_t(W^*,SZ)$ 均为非负且相互独立的分量，对 $F(W^*,SZ)$ 的最小化也就是同时对 $F_r(W^*,SZ)$，$F_o(W^*,SZ)$，$F_t(W^*,SZ)$ 的最小化。由于 $\forall\ i,j, 1\leqslant j\leqslant q, 1\leqslant i\leqslant m$：$Sx_{i,j}=x_{i,j}$，因此，(a)、(b)式分别对应传统 FCM 和 FKP 算法的计算模式。

关于 $F_r(W^*,SZ)=\sum_{k=1}^{K}\sum_{i=1}^{m}W_{k,i}^{\alpha}d_r(SZ_k,Sx_i)$，对于给定的 W^*，右式的各内部求和分量是非负且相互独立的。因此，$F_r(W^*,SZ)$ 的最小化也就是各内部求和分量 $F_r(W^*,SZ_k)=\sum_{i=1}^{m}W_{k,i}^{\alpha}d_r(SZ_k,Sx_i)$ 的最小化。

$F_r(W^*,SZ_k)=\sum_{i=1}^{m}W_{k,i}^{\alpha}\sum_{j=q+1}^{n}\delta_r(SZ_{k,j},Sx_{i,j})=\sum_{i=1}^{m}W_{k,i}^{\alpha}\sum_{j=q+1}^{n}(1-SZ_{k,j}^{x_{i,j}})=\sum_{j=q+1}^{n}\sum_{i=1}^{m}W_{k,i}^{\alpha}-\sum_{j=q+1}^{n}\sum_{i=1}^{m}W_{k,i}^{\alpha}SZ_{k,j}^{x_{i,j}}$。显然，$F_r(W^*,SZ_k)$ 的最小化就是 $\sum_{j=q+1}^{n}\sum_{i=1}^{m}W_{k,i}^{\alpha}SZ_{k,j}^{x_{i,j}}=\sum_{j=q+1}^{n}\sum_{t=1}^{n_j}SZ_{k,j}^t\sum_{i,x_{i,j}=t}W_{k,i}^{\alpha}$ 项的最大化。因为 $\sum_{t=1\cdots n_j}SZ_{k,j}^t=1$ 并且 $\sum_{i=1}^{m}W_{k,i}^{\alpha}$ 固定，当 $\{\sum_{i,x_{i,j}=t}W_{k,i}^{\alpha}\}$ 中有唯一最大值时，存在唯一解；反之则存在无穷组解。定理的(3)式就是一组使 $F_r(W^*,SZ_k)$ 最小化的解。定理得证。

实际上，(b)式是(c)式的一种极端情况，在 $|S|=1$ 时等价。为提高算法效率以及保留更多的聚类信息，可采用更直接的近似最优计算方法：$\forall\ j,t,q+1\leqslant j\leqslant n, 1\leqslant t\leqslant n_j: SZ_{k,j}^t=\sum_{i,x_{i,j}=t}W_{k,i}^{\alpha}/\sum_{i=1}^{m}W_{k,i}^{\alpha}$。

定理2 设成本函数 $F(W,SZ)$ 中的 SZ 固定为 SZ^*，则满足(2)，(3)，(4)的 W：$W^*=\min_{W}F(W,SZ^*)$。W^* 的计算方法是：

(a)对于 $\alpha=1$(对应 SHKP 算法)：

$$W_{k,i}^*=\begin{cases}1, \text{若 } d(SZ_k^*,x_i)\leqslant d(SZ_l^*,x_i), 1\leqslant l\leqslant K\\ 0, \text{否则}\end{cases}$$

(b)对于 $\alpha>1$(对应 SFKP 算法)：

$$W_{k,i}^*=\begin{cases}1, \text{若 } SZ_k^*=Sx_i\\ 0, \text{若 } SZ_l^*=Sx_i, l\neq k\\ 1/\sum_{l=1}^{K}[\frac{d(SZ_k^*,x_i)}{d(SZ_l^*,x_i)}]^{1/(\alpha-1)}, \text{若 } SZ_l^*\neq Sx_i, 1\leqslant l\leqslant K\end{cases}$$

3 算法实现与复杂性分析

为与算法语言语法一致，本节的控制变量初始值由前文中的1一律改变为0。

3.1 数据预处理

除可按需要选择做通常的数据清洗、数据集成与转换、数

据缩减[11]之外，本阶段的核心任务是：

(1)对数值型数据进行规范化处理；对符号型数据将其映射成以0为基数的连续自然数；

(2)建立属性描述表 A：struct {char *Type*；int *Pos*；}*A*[*n*+1]；其中：*n* 为数据集维数；分量 *Type* 为属性类型，取 'N'-Numeric、'O'- Overlap、'S'-Structual 三种。*Pos* 为该属性在结构中心向量中的起始位置，可建立宏：#define *START A*[*j*].*Pos*。多设的最后一个表元素 *A*[*n*]，其 *Pos* 值表示最后一个属性的结束位置。显然，$\forall\ j, A[j].Type='S': n_j=A[j+1].Pos-START$。

3.2 结构化模糊聚类

将 SFKP 分解为距离计算、隶属矩阵 W 及中心矩阵 SZ 的更新等三个主要操作。

(1)主要数据结构：float $X[m][n]$，SZ[K][N]，$w[K][m+1]$；其中 N 为结构中心向量的维数，$N=A[n].Pos=n+\sum_{j,A[j].Type='S'}(n_j-1)$；除 w 与上文中的 W 稍有不同，其它各标识符的含义不变。

(2)距离计算：float *Dist*(float $x[n]$，int k)；返回向量 x 与 $SZ[k]$ 间的距离，其重叠分量和结构化分量为：$\sum_{j,A[j].Type='O'}\delta_0(x[j],SZ[k][\text{START}]$和$\sum_{j,A[j].Type='S'}(1-SZ[k][\text{START}+x[j]])$。计算复杂度为 $O(n)$。

(3)W 的计算。为方便 SZ 的计算，$w[k][i]$中所存储的实际是 $W_{i,k}^2$，最后一列用于保存计算过程中相应行的累加和，即：$\forall\ k, 0\leqslant k<K$：$w[k][m]=\sum_{i=0..m-1}w[k][i]$。$W$ 的计算复杂度为 $O(mnK)$。

(4)SZ 的计算。对符号属性 j，先计算：$\forall\ t, 0\leqslant t<n_j$，$WS[t]=\sum_{i,X[i][j]=t}w[k][i]$；然后再求 $SZ[k]$：若类型为 'O'，则 $SZ[k][START]=\arg\min_{t=0..n_j-1}\{WS[t]\}$；若类型为 'S'，则 $\forall\ t, 0\leqslant t<n_j$，$SZ[k][\text{START}+t]=WS[t]/w[k][m]$。$SZ$ 的计算复杂度为 $O(N_cKm+N_tK(m+N-N_c))=O(mnK+N_tK(N-N_c))$，其中 N_t、N_c 分别为数值属性和符号属性数($n=N_t+N_c$)。最坏情况下($N_t=0$)，复杂度为 $O(mnK+nNK)$。

在计算复杂度方面，符号属性采用重叠类型或结构类型并没有差别。在存储需求上，后者比前者额外增加的存储量为 $K*\sum_{j,A[j].Type='S'}(n_j-1)$。由于 $K<<m$，这不会造成严重的存储负担。

3.3 聚类结果分析与评价

对聚类效果进行度量，这就是聚类的有效性问题。在下面的实验中，除采用文[7～9]中的分类正确率指标(Ac)外，另使用文[3]给出的两个聚类有效性指标。

定义6 聚类的划分系数 $PC=\frac{1}{m}\sum_{k=1}^{K}\sum_{i=1}^{m}W_{i,k}^2$；划分熵 $PE=-\frac{1}{m}\sum_{k=1}^{K}\sum_{i=1}^{m}W_{i,k}\log W_{i,k}$。

划分系数(Partition Coefficient)反映了所有输入对象相对于聚类中心的接近程度。如果每个对象仅属于一类，且此时的 $W_{i,k}$ 较大，则数据的不确定性就较小。划分熵(Partition Entropy)反映了聚类结果的好坏：若所有的 $W_{i,k}$ 接近0或1，则熵就小，所给出的聚类结果就好；若 $W_{i,k}$ 接近0.5，则聚类块的模糊程度高，从而熵就大，相应的聚类结果就差。

性质3 权指数 α 为1时，PC=1，PE=0。

4 实验

实验1将以 UCI[12]中的9个含符号属性的实际数据集(描述见表1)为聚类对象来验证结构化模糊 K-prototypes 聚类算法的有效性；实验2则主要分析算法参数 K、α 对聚类结果的影响。

4.1 实验1 有效性实验

表2列出了四种聚类算法对表1数据集的聚类结果。对于硬性算法(HKP/ SHKP)，$\alpha=1$；对于模糊算法(FKP/SFKP)，$\alpha=1.1$。各算法各自运行50次，并且在每次运行前各算法被设置为相同的初始聚类中心(随机产生)，表中的结果为50次运行的平均值。

表1 数据集描述

数据集名称	对象数	属性数		类别数
		数值	符号	
Soybean-small	47	0	35	4
Soybean	683	0	35	19
Mushroom	8124	0	22	2
Tic-tac-toe	958	0	9	2
Vote	300	0	16	2
Zoo	101	1	15	7
Annealing	798	9	29	5
Crx	490	6	9	2
Labor-neg	40	8	8	2

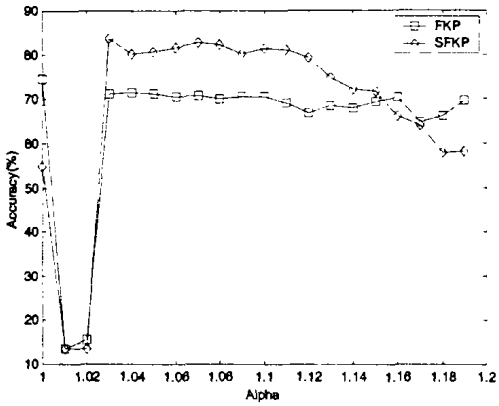
从表2的运行结果可见，SFKP 的分类正确率普遍好于其它两种非结构化算法，多数情况下划分系数和划分熵也优于后者，即使在从划分系数和划分熵角度看聚类结果没有改善的 Mushroom、Tic-tac-toe、Annealing、Labor-neg 等数据集上也能取得最高的分类正确率。

表2 聚类分类结果($\rho=1.0,\gamma=1.0,\varepsilon=10^{-6}$)

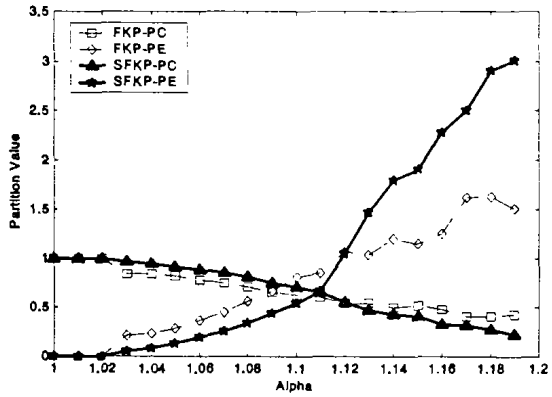
数据集名称	K	HKP	FKP			SHKP	SFKP		
		Ac	Ac	PC	PE	Ac	Ac	PC	PE
Soybean-small	4	77.1	77.1	0.877	0.183	78.7	**86.8**	0.960	0.065
Soybean	30	70.8	69.4	0.655	0.713	73.9	**80.4**	0.730	0.500
Mushroom	2	78.2	72.8	0.788	0.270	80.0	**88.8**	0.757	0.339
Tic-tac-toe	10	67.0	67.3	0.464	1.020	73.0	**80.9**	0.138	2.113
Vote	4	88.8	87.6	0.714	0.416	88.7	**89.0**	0.809	0.287
Zoo	10	87.3	87.7	0.866	0.295	83.9	**90.5**	0.932	0.103
Annealing	6	79.4	78.1	0.760	0.400	**81.7**	80.6	0.742	0.433
Crx	10	82.2	81.2	0.551	0.771	83.0	**83.5**	0.629	0.660
Labor-neg	4	84.3	82.0	0.712	0.444	83.1	**85.0**	0.646	0.582
平均值		79.1	78.1	0.710	0.501	80.67	**85.1**	0.705	0.565

4.2 实验2 参数选择实验

以表1中的Soybean为实验对象，观察 K、α 对聚类结果

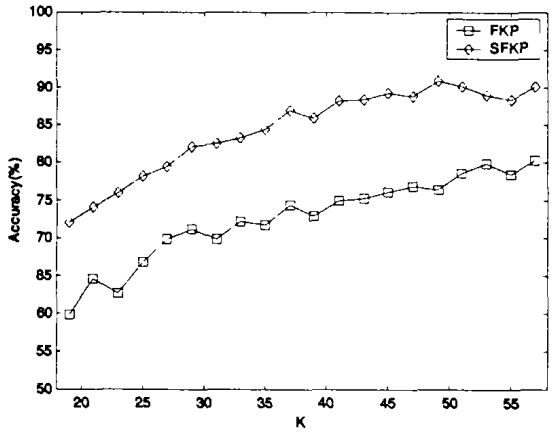


(a) 参数α与分类正确率

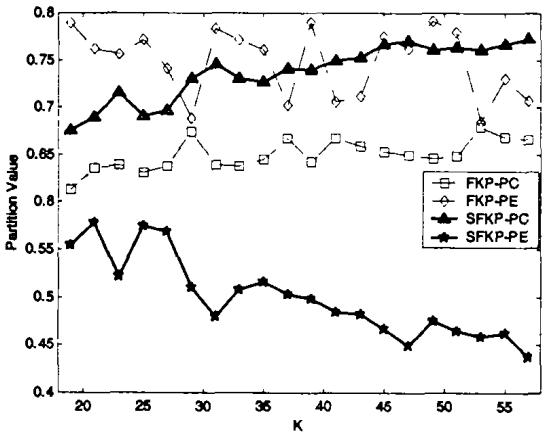


(b) 参数α与划分值

图1 参数α与分类正确率、划分值的关系



(a) 参数K与分类正确率



(b) 参数K与划分值

图2 参数K与分类正确率、划分值的关系

的影响。图1是K固定为30、α由1.0按0.01的步长递增时的运行结果。对于各α值，重复实验1的过程。可见：(1)随着α的增加，划分系数由初始的1逐步递减，而划分熵则由0逐步递增。(2)图1(a)中各算法的分类正确率在$\alpha=1.01$和1.02处极差。实际上，当$\alpha=1.01$时，$1/(\alpha-1)=100$，导致$1/d(Z_i,x_i)^{1/(\alpha-1)}$项的溢出。若要避免，将$W$定义为精度更高的类型即可。图1(b)在此两点处无实验值，为方便，人为定为1或0。(3)在$1.03\leqslant\alpha\leqslant1.17$范围内SFKP的分类正确率要远好于FKP。

图2是当α固定为1.1、K由19按2的步长递增时的运行结果。对于各K值，同样重复实验1的过程。可见：(1)各算法的分类正确率随着K的增加而提高，同时SFKP总是位于FKP之上。(2)SFKP的划分系数和划分熵均优于FKP。(3)随着K的递增，各算法的划分系数呈缓慢增加、划分熵缓慢递减的趋势，但FKP的划分熵不稳定且下降幅度不明显。

上述实验中没有考虑定义4中的ρ、γ参数，均设为1。从形式上看，它们实际上是在计算距离时施加于各属性的权重因子，应该与领域相关，而不应该人为指定。可以利用属性重要性分析的过滤途径(Filter)[13]显式地选中相关属性而丢弃不相关属性；也可利用如Relief-F算法[14]来设置各属性的权值，相关深入研究留待以后。

结论 本文所提出的SFKP算法采用结构化向量的概念，既解决了信息丢失问题，又不致于因聚类中心维数的增加而增加距离计算成本。聚类中心所增加的结构分量为我们分析各聚类的特征提供了充足的信息，这些额外的存储开销仍是值得的。

由定理1所支持的SFKP算法在算法结构上和中心向量及关联矩阵的计算上与以前FKP版本保持了一致。在计算中心向量时，可以将结构分量看作为无结构的普通数值向量；而在距离计算时，又类似于K-modes算法中重叠距离的计算，每个符号属性仅参与计算一次。实验结果显示，SFKP算法是一种有效的混合类型数据的聚类算法。

参考文献

1 Jain A K, Dubes R C. Algorithm for clustering data. Prentice-Hall, 1988

2 Ruspini E R. A new approach to clustering. Information & Control, 1969, 19: 22～32

3 史忠植. 知识发现. 清华大学出版社, 2002

4 Biswas G, Weinberg J, Li C. ITERATE: A conceptual clustering method for knowledge discovery in database. Artificial Intelligence in the Petroleum Industry. In: Braunschweig B and Day R eds. 1995. 111～139

5 Cheesman P, Stutz J. Bayesian classification (AUTOCLASS): theory and results. Advances in Knowledge Discovery and Data Mining, 1995

6 Ralambondrainy H. A conceptual version of the k-means algorithm. Pattern Recognition Letter, 1995, 16: 1147～1157

7 Huang Z. Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining Knowldge Discovery, 1998, 2(3): 283～304

8 Huang Z, Ng M K. A fuzzy k-modes algorithm for clustering categorical data. IEEE Transaction on fuzzy systems, 1999, 7(4): 446～452

9 Chen N, Chen A, Zhou L X. Fuzzy K-prototypes algorithm for clustering mixed numeric and categorical valued data. Journal of Software, 2001, 12(8): 1107～1119

10 汪加才，陈奇，俞瑞钊. 面向分类数据的自组织神经网络. 计算机工程与应用, 2003, 39(5): 96～98

11 Han J W, Kamber M. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000

12 Blake C, Keogh E, Merz C J. UCI repository of machine learning database, 1998

13 Blum A L, Langley P. Selection of relevant features and examples in machine learning. Artificial Intelligence, 1997, 97: 245～271

14 Kononenko L. Estimating attributes: analysis and extensions of relief. In: Proc. of the 1994 European Conf. on Machine Learning, Amsterdam. Springer Verlag, 171～182