第24卷 第5期 控 制 与 决 策 2009年5月
Vol. 24 No. 5 Control and Decision May 2009

文章编号: 1001-0920(2009)05-0697-04

一种面向混合属性数据聚类的新算法

廖志芳[a,b], 罗 浩[b], 樊晓平[b], 刘克准[b]
(中南大学 a. 软件学院, b. 信息科学与工程学院, 长沙 410075)

摘 要: 在分析传统聚类算法的基础上,提出一种针对混合属性数据的聚类算法. 该算法利用格论中简单元组及超级元组将对象属性转化为格模型建立,以对象间格覆盖数来衡量类间相似度,根据高覆盖数高相似度的原则选择聚类中心进行聚类. 在公共数据集上的实验结果表明,该算法在不增加空间复杂度的基础上,有效地提高了混合属性数据聚类的质量.

关键词: 格; 混合属性数据; 覆盖格; 相似度
中图分类号: TP301 **文献标识码**: A

New hybrid data orientation cluster algorithm

LIAO Zhi-fang[a,b], *LUO Hao*[b], *FAN Xiao-ping*[b], *LIU Ke-zhun*[b]
(a. School of Software, b. School of Information Science and Engineering, Central South University, Changsha 410075, China. Correspondent: FAN Xiao-ping, E-mail: xpfan@csu.edu.cn)

Abstract: Based on the analysis of the traditional clustering algorithms, an algorithm is presented to cluster the hybrid data. The method changes the object's attributes to lattice based on the conception of simple tuples and hyper tuples, uses the numbers of covers to measure the similarity between labels, and chooses the clustering mean-point according to the rule of high covers to high similarity. Experiment results based on public data set show that the proposed algorithm can improve the quality of hybrid data clustering, and doesn't increase the space complexity.

Key words: Lattice; Hybrid data; Covers; Similarity

1 引 言

聚类就是将物理或抽象对象的集合分为由类似对象组成的多个类的过程[1],在同一个聚类形成簇中的对象具有较高相似度,不同簇中的对象具有较低相似度. 当前的聚类算法大多用于处理单种属性的数据,如数值型属性的数据处理,或者符号型属性的数据处理,但在实际数据聚类中,需要处理的数据除数值型属性外,还包括文本、图像等符号型属性的混合型数据. 这些算法在进行不同属性的数据处理时,都要进行相关转换,或者全部转换成数值型数据,或者全部转换成符号型数据进行处理,数据聚类精度在进行转换时受到影响[2-8].

为更好地解决混合数据聚类问题,本文提出一种新的面向混合型数据的聚类方法: CBL (Clustering base on lattice)算法. 该方法以格论[9]为基础,利用格论中简单元组和超级元组的概念,对数据进行格的划分,以非距离的格关系作为聚类相似度的衡量方法,在不增加空间复杂度的条件下,该算法有效地提高了聚类精确度.

2 格论基础

2.1 格的相关定义

设 R 是非空集合 A 上的一个二元关系,若 R 满足自反性、反对称性、传递性,则称 R 为 A 上的偏序关系,通常记作 $\leq$. 假设 $B \subseteq A$, C 是由 B 的上界组成的集合,即 $c = \{y \mid y \in A \wedge \forall x \in B, x \leq y\}$,若 C 的最小元 y_0 存在,则 y_0 称作 B 的最小上界或上确界(记为 $\mathrm{lub}(B)$),记为 $x \vee y$;反之,设 C 是由 B 的下界组成的集合,即 $c = \{y \mid y \in A \wedge \forall x \in B, y \leq x\}$,若 C 的最大元 y_0 存在,则 y_0 称作 B 的最大下界或下确界,记为 $x \wedge y$. 设 $(L, \trianglelefteq)$ 是一个偏序集,若对于任意的非空子集都有最小上界和最大下界,则称 $(L, \trianglelefteq)$ 构成一个格,记为 $(L, \wedge, \vee)$[9]. 若对于

收稿日期: 2008-05-21; **修回日期**: 2008-09-04.
基金项目: 国家自然科学基金项目(60774023); 国家863计划项目(2005AA1Z2330); 湖南省自然科学基金项目(06JJ50143).
作者简介: 廖志芳(1968—),女,湖南桂阳人,副教授,从事数据挖掘的研究; 樊晓平(1961—),男,浙江绍兴人,教授,博士生导师,从事智能信息处理等研究.

任意的 $x,y\in L$,$\{x,y\}$ 都有最小上界,即 $x+y$ 存在;或者对于任意的 $x,y\in L$,$\{x,y\}$ 都有最大下界,则称 $(L,\leqslant)$ 构成一个半格,记为 $(L,\vee)$ 或 $(L,\wedge)$. 本文中,半格特指 $\{x,y\}$ 都有最小上界 $(L,\wedge)$ 的情况.

根据格的定义,可以定义一维及多维空间格. 在一维格中,对于数值型属性,沿用原定义来描述格,例如一个自然数对 $L\langle 1,5\rangle$ 组成的区间就是一个格. 对于符号型属性,定义一个集合为一个格,如一个集合 $L\{a,b,e,g,h,x\}$ 就是一个格. 将一维空间内的格扩展到多维空间,则所有属性上的各个数值区间/属性值集合就组成一个多维的格. 如包含二维属性的数据空间,其中一个属性为数值型,另一个属性为符号型,则 $L\langle(0,5),\{a,b,e,g\}\rangle$ 就为一个二维混合属性格.

设 D 是 $\Omega=\{x_1,x_2,\cdots,x_T\}$ 的关系属性空间,V_x 是属性 $x\in\Omega$ 的值域空间. 如果

$$L=\prod_{x\in\Omega}2^{V^x},$$

则 L 为半格. L 按以下关系排序:

$$\forall t,s\in L,t\leqslant s\Leftrightarrow t(x)\subseteq s(x),\quad\forall x\in\Omega. \tag{1}$$

其数据总合为

$$t+s\stackrel{\text{def}}{=}\langle t(x)\cup s(x)\rangle_{x\in\Omega}. \tag{2}$$

此时 L 定义为 D 的格域空间,其中 L 中的元素被称为超级元组,一个超级元组的集合称为超元组关系. V_x 称作简单元组,L 与 D 的关系可表示如下:

$$\Omega(a)\mapsto\langle\{x_0(a)\},\{x_1(a)\},\cdots,\{x_T(a)\}\rangle,$$

且 $D\subseteq V\subseteq L$.

2.2 基于格的聚类度量方法

设 D 是一个关系数据空间,$\Omega=\{x_1,x_2,\cdots,x_T\}$ 为关系属性空间,L 是 D 的域空间,$\{D_1,D_2,\cdots,D_k\}$ 是 D 的聚类类别,对于每一个 D_i,既是数据集 D 中的子集,同时也是 D 中的一个类别对象简单元组集合. 为获得 D_i 中的聚类对象,将 $\{D_i\}$ 中所有简单元组按式(2) 形成对应的超元组关系 $H=\{h_1,h_2,\cdots,h_k\}$,将聚类问题变成将简单元组关系 (D) 转换成超级元组关系 (H) 的问题,即 D 的聚类就是找出 $H\subseteq[D]$ 的超级关系.

假设存在两个格:$L_1=\{[1,9],\{a,e,g,h,i\}\}$,$L_2=\{[2,5],\{e,g,h\}\}$. 对于数值部分的元素,显然有 $\min(1,9)<\min(2,5)$,$\max(1,9)>\max(2,5)$;对于符号型部分的元素,有 $\{e,g,h\}\subseteq\{a,e,g,h,i\}$,可得 $L_2\leqslant L_1$.

对于一个对象 $a(3,h)$,$3\in[1,9]$,$h\in\{e,g,h\}$,可得出 $a\in L_1$,同理可得 $a\in L_2$,本文将 L_1 和 L_2 称为 a 的覆盖格. 若有 b 同时被 L_1 和 L_2 覆盖,则称 a 和 b 为近邻,覆盖数越多,说明 a,b 关系越密切. 因此,为判定各 D_i 超级关系,本文采用超级元组的覆盖数方法对其进行衡量,于是度量方法就转变为超级元组覆盖数的计算,即对数据样本 D 中的每个简单元组 x,试图找到覆盖 x 和 t 的覆盖数. 本文称这个数字为 x 对 t 的覆盖次数,记作 $\operatorname{cov}(t,x)$.

对于 $c\in C$,定义 $D_c\stackrel{\text{def}}{=}\{x\in D:x$ 在聚类 c 中$\}$. 如果 D 中每个元素都有不相同的类标签,则 D 可分割为所有 D_C 组成的集合. 定义

$$K(t,c)=\sum_{x\in D}\operatorname{cov}(t,x),$$

$K(t,c)$ 是 c 类中所有简单元组的总覆盖数. 从而,得到下列聚类规则:

规则 1 如果 c_0 类得到最大的 $K(t,c)$ 值,则 t 被聚类到 c_0 类.

3 聚类算法

根据上述格论的基本原理以及基于格的聚类度量方法,提出一种新的面向混合数据的聚类新算法. 该算法不再根据对象间的距离划分对象的聚类,而是通过计算能同时覆盖两个对象的格数来确定其相似度,也就是用覆盖关系来度量对象间的相似度. 对于一个对象 x,考虑能够覆盖 x 的所有格 N,对于任意其他数据对象 t,计算能同时覆盖这两个对象的格的数量 $\operatorname{cov}(x,t)$,用 $\operatorname{cov}(x,t)$ 来描述它们的相似度,$\operatorname{cov}(x,t)$ 值越大,对象 (x,t) 相似度就越大. 通过这种方法可以计算各对象 $x_i(i=1,2,\cdots,n)$ 到聚类中心 $c_j(j=1,2,\cdots,k)$ 的覆盖数 $\operatorname{cov}(x_i,c_j)$,从而将对象划分到不同簇. 实验表明,该方法对于符号型数据、数值型数据及混合型数据的处理都十分有效.

3.1 算法的基本思想

该算法首先任选 k 个对象作为初始的聚类中心 c_i,剩余的对象根据其与各中心的相似度,也就是 $\operatorname{cov}(x_j,c_i)$ 值的大小分配到最相似的簇中;然后对每个簇进行更新:计算簇类每个对象 $x_i\in c_i$ 与其他所有对象 $x_j\in c_i$ 之间的覆盖格数 $\operatorname{cov}(x_i,x_j)$ 并求和,选取拥有最大 $\sum_{i=1}^{n_l}\operatorname{cov}(x_i,x_j)$ 值的对象 x_i 作为新的簇中心,并重新聚类,直至稳定.

为获取对象的覆盖数,首先需由数据对象构成超级元组. 一个超元组可从每个属性中提取一个集合产生. 假设 a_i 为一属性,$i=1,2,\cdots,n$,S_i 为 $\mathrm{Dom}(a_i)$ 所有子集的集合,定义 $N_i'\stackrel{\text{def}}{=}|S_i|$,那么超元组 h 为 $\prod S_i$ 的一个元素. 因此,所有超元组的个数为 $\prod N_i'$. 归纳而言,所有的超级元组个数为 $\prod N_i'$,其中

$$N_i' = \begin{cases} 2^{m_i}, & a_i \text{ 为文本型数据；} \\ \dfrac{m_i(m_i+1)}{2}, & a_i \text{ 为数值型数据.} \end{cases} \tag{3}$$

对于任意数据对象 $x \in V$，并非所有的格 L 生成的超级元组都能覆盖 x. 根据前述理论，若 $X \leq L$，则在每一维属性 x_i 上，都必须有 L_i 覆盖 x_i，即 $x_i \in L_i$，L 覆盖 x.

若 a_i 为符号型属性，则能同时覆盖 x_i，c 的格即为 S_i 中包含 x_i，c 的元素. 因此，当 $x_i = c_i$ 时，$N_i = 2^{m_i - 1}$；当 $x_i \neq c_i$ 时，$N_i = 2^{m_i - 2}$.

若 a_i 为数值型属性，则能同时覆盖 x_i，c 的格即为上界大于 $\max\{x_i, c_i\}$，下界小于 $\min(x_i, c_i)$ 的区间. 因此有

$$N_i = (\max(a_i) - \max(c_i, x_i) + 1) \times (\min(c_i - x_i) - \min(a_i) + 1).$$

根据上述分析，可以求得两个或多个对象间的覆盖格数 N，并通过对覆盖格的计数，可以将对象分配到不同的类中去.

3.2　算法描述

整个算法可以大致描述如下：

输入：簇的数目 k 和包含 n 个对象的数据库；

输出：k 个簇，簇内对象相异度总和最小.

```
For i = 1 to k{
    choose ith objects as initial clustering center c_i
}
Repeat {
  For i = 1 to n{
     For j = 1 to k{
        count the cover number cov(object_i, c_j) of each object_i to clustering center c_j and compare with current maximum cover number cov(object_i, c_j')
      if cov(object_i, c_j) > cov(object_i, c_j') then
        dispatch object_i to c_j
     } // End for
  } // End for
  For i = 1 to k{
    // Renew clustering center c_i
    For j = 1 to n_i{
       For r = 1 to n_i( r ≠ j){
         count each object's total cover number sum_j
    sum j = sum_j + cov(x_r, c_j)
         } // End for
      choose the object that have the max total cover number and set it as the new center
      Set c_i = obj_j
  } // End for
Untill clustering result do not change}
```

3.3　算法分析

3.3.1　K 值的选取

在大部分聚类算法中，分类数目是需要用户预先设置的参数. 对于同一个数据库，用同样的聚类算法聚类多次，每次设置不同的类数目，可以得到一系列聚类结果. 因此，选择合适的类数目是正确聚类的前提，本算法利用遗传算法对聚类数目 k 进行选择.

定义类间覆盖格数

$$\mathrm{CBC} = \sum_{i=0}^{k}\sum_{j=i+1}^{k}\mathrm{cov}(C_i, C_j). \tag{4}$$

其中：C_i 表示第 i 个簇的中心代表点，$\mathrm{cov}(C_i, C_j)$ 为两个簇中心代表点之间的覆盖格数.

类内覆盖格数

$$\mathrm{CIC} = \sum_{i=0}^{k}\sum_{P_j \in C_i}\mathrm{cov}(P_j, C_i), \tag{5}$$

其中 p_j 为簇 C_i 中数据对象.

根据以上规则，定义适应度函数

$$S = \omega_1 \frac{\mathrm{CBC}}{\mathrm{CIC}} + \omega_2 \frac{1}{nd}. \tag{6}$$

其中：nd 为各簇间样本差别数；ω_1 和 ω_2 为权重，用来调节均匀度对函数的影响.

通过对适应度函数的计算，可以确定合适的 k 值，$K \leq \sqrt{n}$. 由于 k_r 值的范围必须落在区间 $[1 - k_{\max}]$ 内，如果经过变异、交叉等操作后，k_r 值落在该区间之外，则可以取消该操作以减少计算.

3.3.2　聚类质量分析

本文算法采用覆盖格的数量来衡量对象间的相似度，因此不仅能有效地处理数值型数据，对于符号型属性更是比其他算法有效. 在 K-mode 算法中，符号型数据的距离采用相等即为 1，不等则为 0，然后乘以权值的方式，而权值的确定往往需由用户确定. CBL 算法中符号型数据的距离受属性值域的影响，根据属性本身的特性来确定距离，无需用户干预，因而减少了权值确定这一环节. 在更新聚类中心时采用了 K-means 算法中的随机替换法，有效地克服了 K-prototype 算法选取出现最高频率的属性值为新聚类中心而导致的信息损失问题.

3.3.3　算法复杂度

设 n 为对象数量，k 为簇数量，$k \ll n$，t 为迭代次数. 在计算过程中，首先选取 k 个聚类中心，时间复

杂度为 $O(k)$;将 n 个元素分配到不同簇,执行 n 次操作,时间复杂度为 $O(n)$;接着更新聚类中心,更新次数为 $\sum^{k} n_i \times (n_i - 1) \approx n^2 / k$,时间复杂度为 $O(n^2)$,其中 n_i 为各簇元素数量. 因此,整个算法平均时间复杂度为 $T(n) = O(k) + t(O(n) + O(n^2)) = O(n^2)$. 从时间复杂度上看,该算法在处理小数据量数据集时其有效性更高.

4 实验结果

实验环境为 Pentium IV 351 CPU ,512 M DDR SDRAM ,80 G HDD ,操作系统 Windows XP (sp2) 专业版,开发工具为 VC ++ 6.0(sp6). 选取一组纯数值型、一组纯文本型和两组混合型数据集进行实验.

4.1 数据集说明

本文从 UCI 公共数据库选用了纯数值型、纯符号型以及混合型数据集进行实验. 这些数据集分别为 Breast cancer wisconsin(BCW) , Ecoli ,Japanese Credit Approval(JCA) 和 House Votes(HV) ,数据集的详细描述如下表 1 所示.

表 1 数据集的说明

	Type	Number of attributes		Category	Instance	Missing data
		Nominal	Numeric			
BCW	Numeric	0	10	2	699	Yes
Ecoli	Mixed	2	6	8	336	No
JCA	Mixed	8	8	2	690	Yes
HV	Nominal	17	0	2	435	Yes

这 4 个数据集涵盖了 3 种数据类型,其中 BCW 为纯数值型数据,Ecoli 和 JCA 为数值与符号型混合数据,HV 为纯符号型数据.

4.2 实验分析

4 个数据集均在相同环境下利用 4 种算法 K-prototype ,CBL ,EM 以及 DBScan 分别独立进行实验,利用图示以及可视化表示方法分别对数据聚类结果进行比较. 聚类结果的精确度比较见图 1.

从图1可以看出,CBL 算法的聚类精度在处理

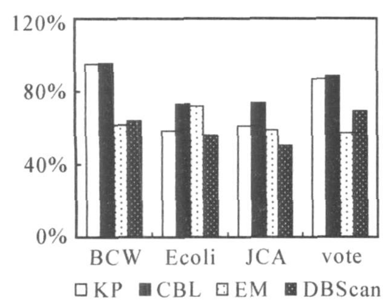

图 1 聚类精度比较

BCW 和 HV 数据集时,与 K-prototype 算法的聚类精度相似,但优于其他 2 种算法;而在处理 Ecoli 和 JCA 混合数据集时,算法精度明显高于其他算法. 这说明 CBL 算法在处理混合型数据时有效性更高,而在处理单一属性的数据集时,有效性与其他算法区别不大.

下面继续利用可视化表示方法来说明算法的有效性. 本可视化表示仅以 Ecoli 数据集为例.

4.3 实验结果的可视化表示

在图 2 中,对象被随机分配到以簇中心为圆心、对象到簇中心的距离(与覆盖数成反比) 为半径的圆上,C 代表簇标号,方形标志为正确聚类对象,叉形标志为未正确对象. 通过图示可以直观地看到聚类分布的情况,对象距离图中心越近,说明其与中心点之间的覆盖数越大,相似度就越大;反之越远的相似度就越小.

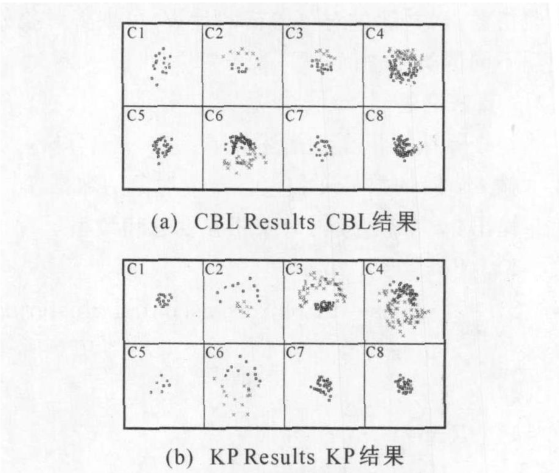

图 2 Ecoli 数据集可视化结果

数据来源于日本大阪大学细胞和分子生物学研究所,包含 8 个属性(其中 2 个二值型,6 个数值型) ,共 336 条记录,数据记录用来预测生物蛋白在原核细胞中的位置. 图 2 给出了两种方法聚类的可视化结果.

5 结 论

本文提出了一种新的面向混合型数据的聚类算法,采用覆盖格的数量来衡量对象间的相似度. 这种算法对于混合型数据处理非常有效,不仅能处理数值型数据,而且能处理符号型数据,与传统处理符号型数据算法(如 K-prototype , K-mode) 相比,聚类效果更加有效. 但本文算法在初始聚类中心的选取上采取随机选取的方法,同时在聚类类别的选取上也有赖于先验经验,在混合数据处理上未考虑数据权重的处理. 因此,在进一步的研究中将就这些方面进行更深入的研究,以期达到更优的聚类结果.

(下转第 705 页)

判[J]. 系统工程理论与实践, 2002, 22(4): 141-144.
(Bu G Z, Zhang Y W. Grey fuzzy comprehensive evaluation based on the theory of grey fuzzy relation[J]. Systems Engineering-Theory & Practice, 2002, 22(4): 141-144.)
[7] 朱绍强, 孟科, 张恒喜. 区间数灰色模糊综合评判及其应用[J]. 电光与控制, 2006, 13(5): 36-38.
(Zhu S Q, Meng K, Zhang H X. Interval numbers grey fuzzy comprehensive evaluation and its application[J]. Electronics Optics & Control, 2006, 13(5): 36-38.)
[8] 寇进忠. 随机性方法与灰色性方法的互补问题[J]. 系统辩证学报, 1998, 6(4): 81-83.
(Kou J Z. On the mutual complementarity between random methods and grey methods[J]. J of Systemic Dialectics, 1998, 6(4): 81-83.)
[9] 于义彬, 王本德, 柳澎, 等. 具有不确定信息的风险型多目标决策理论及应用[J]. 中国管理科学, 2003, 11(12): 9-13.
(Yu Y B, Wang B D, Liu P, et al. Risky multiobjective decision-making theory and its application[J]. Chinese J of Management Science, 2003, 11(12): 9-13.)
[10] 张席敬, 喻建华, 周其华. 灰色随机变量的熵和信息量[J]. 南昌大学学报, 2005, 29(8): 337-341.
(Zhang X J, Yu J H, Zhou Q H. The entropy and the quantity of information of the grey random variable[J]. J of Nanchang University, 2005, 29(8): 337-341.)
[11] 罗党, 刘思峰. 灰色多指标风险型决策方法研究[J]. 系统工程与电子技术, 2004, 26(8): 1057-1060.
(Luo D, Liu S F. Research on grey multi-criteria risk decision-making method[J]. Systems Engineering and Electronics, 2004, 26(8): 1057-1060.)
[12] 王坚强, 任世昶. 基于期望值的灰色随机多准则决策方法[J]. 控制与决策, 2009, 24(1): 39-43.
(Wang J Q, Ren S C. Grey random multi-criteria decision-making approach based on expected value[J]. Control and Decision, 2009, 24(1): 39-43.)
[13] 刘思锋, 郭天榜, 党耀国, 等. 灰色系统理论及其应用[M]. 北京: 科学出版社, 1999.
(Liu S F, Guo T B, Dang Y G, et al. Grey system theory and its application[M]. Beijing: Science Press, 1999.)
[14] 岳超源. 决策理论和方法[M]. 北京: 科学出版社, 2003.
(Yue C Y. Decision-making theory and method[M]. Beijing: Science Press, 2003.)
[15] Goumas M, Lygerou V. An extension of the PROMETHEE method for decision making in fuzzy environment: Ranking of alternative energy exploitation[J]. European J of Operational Research, 2000, 123(3): 606-613.
[16] 王坚强. 信息不完全的多准则决策的 SIR 方法[J]. 系统工程与电子技术, 2004, 26(9): 1205-1208.
(Wang J Q. Superiority and inferiority ranking methods for multiple criteria decision making with incomplete information on weights [J]. Systems Engineering and Electronics, 2004, 26 (9): 1205-1208.)

(上接第 700 页)

参考文献(References)

[1] Berkhin P. A survey of clustering data mining techniques[R]. Berlin: Springer Heidelberg, 2006.
[2] Johannes Grabmeier, Andreas Rudolph. Techniques of cluster algorithms in data mining[J]. Data Mining and Knowledge Discovery J, 2002, 6(4): 303-360.
[3] Ralambondrainy H. A conceptual version of the *k*-means algorithm[J]. Pattern Recognition Letters, 1995, 16 (11): 1147-1157.
[4] Hui Wang, Werner Dubitzky. A flexible and robust similarity measure based on contextual probability[C]. Int Joint Conf on Artificial Intelligence, 2005: 27-34.
[5] Ma Shuai, Wang T J, Tang S W, et al. A fast clustering algorithm based on reference and density[J]. J of Software, 2003, 14(6): 1089-1095.
[6] Jiang Shengyi, Song Xiaoyu, Wang Hui, et al. A clustering-based method for unsupervised intrusion detections[J]. Pattern Recognition Letters, 2006, 27 (7): 802-810.
[7] Steinbach M, Karypis G, Kumar V. A comparison of document clustering techniques [C]. Proc of the 6th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. Boston, 2000: 1023-1030.
[8] 胡长流, 宋振明. 格论基础[M]. 郑州: 河南大学出版社, 1990: 64-65.
(Hu C L, Song Z M. Foundation of lattice theory[M]. Zhengzhou: He'nan University Press, 1996: 64-65.)
[9] 胡瑞飞, 殷国富, 谭颖. 一种混合聚类算法及其应用[J]. 四川大学学报, 2006, 38(5): 156-161.
(Hu R F, Yin G F, Tan Y. A hybrid clustering algorithm and its application [J]. J of Sichuan University, 2006, 38(5): 156-161.)