ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.9, September 2009, pp.2387−2396
doi: 10.3724/SP.J.1001.2009.03412

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种基于概念的数据聚类模型*

张明卫[1,2+], 刘 莹[2], 张 斌[1], 朱志良[2]

[1](东北大学 信息科学与工程学院,辽宁 沈阳 110004)

[2](东北大学 软件学院,辽宁 沈阳 110004)

Concept-Based Data Clustering Model

ZHANG Ming-Wei[1,2+], LIU Ying[2], ZHANG Bin[1], ZHU Zhi-Liang[2]

[1](School of Information Science and Engineering, Northeastern University, Shenyang 110004, China)

[2](School of Software, Northeastern University, Shenyang 110004, China)

+ Corresponding author: E-mail: neuzmw@163.com

Zhang MW, Liu Y, Zhang B, Zhu ZL. Concept-Based data clustering model. *Journal of Software*, 2009,20(9):2387−2396. http://www.jos.org.cn/1000-9825/3412.htm

Abstract: In data mining, lots of clustering algorithms have been developed, and most of them are limited by scalability and interpretability. To solve this problem, a concept-based data clustering model is presented. From the perspective of the metadata describing samples, some basic concepts are extracted from the preprocessed dataset firstly in this model, and then generalizes, higher level concepts representing clustering results. Finally, the samples are classified into different final concepts and the clustering process is completed. On the premise of ensuring the accuracy of the clustering results, this model can greatly decrease the number of tuples needing to be processed, improving the data scalability of clustering algorithms. In addition, to discover and analyze knowledge based on concepts, this model can improve the interpretability of clustering results, and facilitate to interact with users. Experimental results show that the proposed model is more useful to the algorithms with higher computation cost and better results.

Key words: data mining; clustering; concept; concept tuple; model

摘 要: 在数据挖掘研究领域,现有的大多数聚类算法都受到数据可伸缩性和结果可解释性的限制.为了解决这一难题,提出了一种基于概念的数据聚类模型.该模型从描述数据样本的数据本身出发,首先在预处理后的数据集上提取基本概念,再对这些概念进行概化,形成表示聚类结果的高层概念,最后基于这些高层概念进行样本划分,从而完成整个聚类过程.该模型能够在保证聚类准确性的基础上,很大程度地减少要处理的数据量,提高原算法的可伸缩性.另外,该模型基于概念进行知识的发现与分析,能够提高聚类结果的可解释性,便于与用户交互.实验结果表明,该模型对于聚类结果较好且复杂度较高的算法尤为有效.

关键词: 数据挖掘;聚类;概念;概念元组;模型

---

* Supported by the National Natural Science Foundation of China under Grant No.60773218 (国家自然科学基金); the Key Project of the National 'Tenth Five-Year-Plan' of the Ministry of Science and Technology of China under Grant No.2004BA721A05 (国家科技部"十五"攻关项目)

Received 2007-10-28; Revised 2008-02-20; Accepted 2008-07-02

中图法分类号: TP18 文献标识码: A

数据挖掘(data mining)是知识发现的一种手段,能够从巨量的数据集合中抽取隐含的、先前未知的、对决策有潜在价值的规则[1].当挖掘任务面临缺少领域知识或领域知识不完整的数据集合时,可以采用聚类分析技术.所谓聚类,就是按照相似程度把大量的数据样本(n 个)聚集成 k 个类($k<n$),使得同一类内样本的相似性较大,而不同类间样本的相似性较小.目前聚类算法有很多,如基于划分的算法(k-means[2], FREM[3])、基于密度的算法(ST-DBSCAN[4],DENCLUE[5],OPTICS[6])、基于层次的算法(BIRCH[7],CURE[8]),以及基于网格和子空间的算法(STING[9])等等.虽然这些算法在聚类性能上各有优劣,但数据可伸缩性和结果可解释性始终是聚类研究中的热点和难点问题.

对于数据可伸缩性,各种基本算法几乎都有自己的改进模型,然而它们大部分都受到时间和空间复杂度的限制,随着数据量的增加,可能使任何一种聚类算法都无法正常运行.对于结果可解释性,概念聚类是对一般聚类方法的改进[10,11].和已往方法不同,概念聚类将聚类过程分为两步:(1) 发现合适的簇;(2) 形成对每个簇的概念描述.概念聚类的优点主要体现在能对聚类结果给予合理的解释.

从方法论上讲,聚类分析有以下两种做法.第 1 种是从数据对象的角度出发,通过计算它们之间的相似程度(距离),形成合理的对象簇,然后再对这些簇进行概念分析和知识展示.现有的聚类算法大多采用该方法.第 2 种方法则是从描述数据对象的数据本身出发,首先从这些数据中提取出基本概念,然后再对这些概念进行概化,以形成更高层的概念,最后把数据对象分到不同的高层概念中,从而产生最终的聚类结果.

基于方法 2,本文提出了一种基于概念的数据聚类模型 CBCM(concept based clustering model).首先对德国 Wille 教授提出的形式概念分析中的概念加以改进,定义了用于聚类分析的概念.在此基础上,给出了利用模型 CBCM 进行聚类的整个过程,并描述了相应的算法.最后,本文在模拟生成的数据集和中医小儿肺炎病例数据集上进行了详细的实验分析,证明了该模型的有效性.

利用模型 CBCM 进行聚类有以下优点.首先,由于数据对象的个数通常要远远大于描述其概念的个数,因此,该模型可在很大程度上提高算法的可伸缩性.其次,由于模型 CBCM 基于概念进行聚类,所以能够提高聚类结果的可解释性.最后,利用该模型进行聚类便于与用户交互,因为模型 CBCM 先对概念进行概化,再对数据对象进行分类,如果用户不满意最终生成的概念,则不必对数据对象进行分类,直接进行再次概念概化即可.然而,模型 CBCM 也存在缺点,即较难确定模型中的参数.

1 概念与概念元组

给定一个有限 m 维离散向量空间 $U=D_1\times D_2\times\ldots\times D_m$,其中 D_j 是有限符号集($j=1,2,\ldots,m$).$u=\langle v_1,v_2,\ldots,v_m\rangle$是以符号向量形式描述的对象,属性值 $v_j\in D_j,\forall u\in U$ 称为 U 的实例.称$|U|$为集合 U 的元素的个数,表示了集合 U 的尺度.

定义 1(值分布矩阵). 用 $Dom(D_j)=\{v_{j1},v_{j2},\ldots,v_{jk}\}$表示向量 D_j 的值域,其中,$k=|Dom(D_j)|$.存在矩阵:

$$m(D_j)=\begin{bmatrix} v_{j1},v_{j2},\ldots,v_{jk} \\ p_{j1},p_{j2},\ldots,p_{jk}\end{bmatrix} \tag{1}$$

其中,$0\le p_{ji}\le 1$,($i=1,2,\ldots,k$),表示值 v_{ji} 出现的概率,且 $\sum_{i=1}^{k}p_{ji}=1$.则称矩阵 $m(D_j)$为向量 D_j 的一个值分布矩阵.

定义 2(概念). 给定一个有限 m 维离散向量空间 $U=D_1\times D_2\times\ldots\times D_m$,存在向量 $c(U)=\langle m(D_1),m(D_2),\ldots,m(D_m)\rangle$,其中,$m(D_j)$是向量 D_j 的一个值分布矩阵($j=1,2,\ldots,m$),则称向量 $c(U)$为向量空间 U 上的一个概念.

定理 1. 离散向量空间 U 上的任一非空子集 $T\subseteq U$,能够唯一确定一个 U 上的概念 $c(T)$.

证明:给定离散向量空间 U 上的任一非空子集 $T\subseteq U$,对于空间 U 的任一维向量 D_j,根据集合 T 中元组在 D_j 上的取值概率分布,能唯一确定该集合在向量 D_j 上的值分布矩阵.因此,非空子集 T 能够唯一确定概念 $c(T)$. □

定义 3(元值分布矩阵). 给定值域 $Dom(D_j)=\{v_{j1},v_{j2},\ldots,v_{jk}\}$的向量 D_j,如果向量 D_j 的值分布矩阵 $m(D_j)$中,存在 $i\in\{1,2,\ldots,k\}$,使 $p_{ji}=1$,则称矩阵 $m(D_j)$为向量 D_j 的元值分布矩阵.

元值分布矩阵表示向量 D_j 确定取单一值 v_{ji},而取其他值的概率为 0.根据定义 3 可知,向量 D_j 共有$|Dom(D_j)|$个元值分布矩阵.

定义 4(元概念). 给定一个有限 m 维离散向量空间 $U=D_1\times D_2\times...\times D_m$,$c$ 为向量空间 U 上的一个概念,如果概念 c 中的每个元素都是元值分布矩阵,则称概念 c 为向量空间 U 上的元概念.

根据元概念的定义可知,m 维离散向量空间 $U=D_1\times D_2\times...\times D_m$ 上共有 $\prod_{j=1}^{m}|Dom(D_j)|$ 个元概念.

定理 2. 离散向量空间 U 上的一个元概念 c 和 U 上的一个元组 $u\in U$ 等价.

证明: (1) 给定离散向量空间 U 上的一个元概念 c,对于空间 U 的任一维向量 D_j,c 在 D_j 上能够唯一确定一个值 v_{ji},使得 p_{ji}=1.将这些各维上的值合在一起,即组成元组 u.因此,元概念 c 能够唯一确定 U 上的一个元组 $u\in U$.

(2) 给定离散向量空间 U 上的一个元组 $u\in U$,对于任一维向量 D_j,u 在 D_j 上的取值能够唯一确定一个元值分布矩阵 $m(D_j)$,将各维上的元值分布矩阵合在一起,即组成元概念 c.因此,元组 u 能够唯一确定 U 上的一个元概念 c.

由上述证明可知,离散向量空间 U 上的一个元概念 c 和其对应的元组 $u\in U$ 等价. □

定义 5(概念元组). 给定离散向量空间 U 上的一个元概念 c 和它所唯一确定元组 $u\in U$,称 u 为向量空间 U 上的一个概念元组.

由定理 2 可知,m 维的离散向量空间 $U=D_1\times D_2\times...\times D_m$ 上共有 $\prod_{j=1}^{m}|Dom(D_j)|$ 个概念元组.

2 基于概念的数据聚类模型

聚类是按照某种相似程度的度量把大量数据样本进行分类,使得同一类内样本的相似性较大,而不同类间样本的相似性较小.现有聚类算法大多从数据对象(即数据元组)的角度出发,通过数据元组之间几何距离或语义距离的度量来确定它们之间的相似性,从而将不同的元组归于不同的类.然而,数据元组是由分布在各个属性上的数据值进行描述的.各属性不同数据值的组合形成一系列的元组,每一个元组都代表一种数据对象可能的取值情况,根据上一节的定义,把这些元组称为概念元组.

在聚类过程中,用概念元组代替原有样本,可以取得效果相当的聚类结果.举例来说,假设待聚类样本集中存储了世界上所有的东北虎、大熊猫、蚂蚁和蝗虫这 4 种动物,这些样本通过诸如“长有腿的数量”、“繁殖方式”、“食物种类”等合适的属性进行描述.因为样本集庞大,可能没有任何一种聚类算法能够有效地对该样本集进行聚类.而从描述样本对象的数据本身出发,可以生成数量有限的概念元组.在聚类中,一个代表“澳大利亚小红蚁”的概念元组可以代替几亿亿只的小红蚁个体,而不会导致该类信息的丢失.利用概念元组进行聚类,可以生成相应的概念,再根据生成的概念对原数据样本进行分类,会产生最终的聚类结果.图 1 给出了基于概念的数据聚类模型.利用该模型进行聚类,可以很好地解决聚类中存在的数据可伸缩性和结果可解释性等问题.

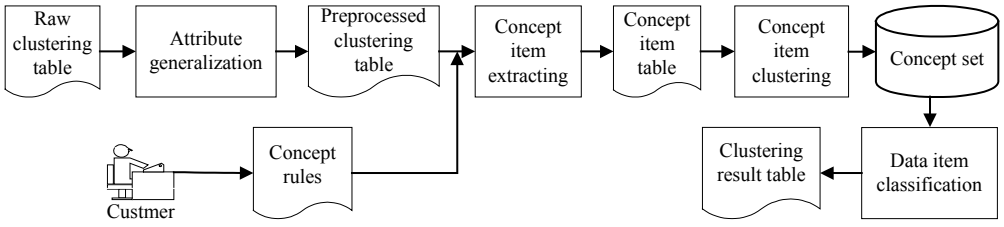


Fig.1 Concept based data clustering model

图 1 基于概念的数据聚类模型

为了提取概念元组,首先需要对原始聚类表的数值属性进行概念化分段.把分段后的聚类表看成离散向量空间,根据一定规则在该向量空间中提取概念元组.挖掘用户可以根据自身经验提出一些约束性知识作为概念

规则,并根据这些规则对提取的概念元组进行过滤,生成相应的概念元组表.接下来可以把概念元组当作一般的数据库元组,利用现有的聚类算法对其进行聚类,从而生成一些概念元组的簇.根据定理 1,每个簇唯一确定一个概念,所以通过概念元组的聚类,会得到概化的概念,并生成相应的概念库.最后,需要将原始聚类表中的元组划分到不同的概念中,从而产生最终的聚类结果.

2.1 数值属性的概念化分段

为了产生离散向量空间,并有效缩减聚类空间,需要将原始聚类表中的数值属性进行概念划分,以形成有序的概念值.对数据属性进行离散化的方法有许多,本文针对有原始分类信息的数据集,给出了一种基于基尼系数的离散化方法,用于递归地划分数值属性的值.

给定一个数据元组的集合 S,基于基尼系数对数值属性 A 进行离散化的算法可简要描述如下:

1) A 的每个值可以认为是一个潜在的区间边界或阈值 T.例如,A 的值 v 可以将样本 S 划分成分别满足条件 $A<v$ 和 $A\geq v$ 的两个子集,这样就创建了一个二元离散化.

2) 给定 S,所选择的阈值是这样的值,它使划分得到的基尼系数最小.基尼系数的度量如下:

$$gini_{split}(S,T)=\frac{|S_1|}{|S|}gini(S_1)+\frac{|S_2|}{|S|}gini(S_2) \tag{2}$$

其中,S_1 和 S_2 分别对应 S 中满足 $A<T$ 和 $A\geq T$ 的样本.对于给定集合,它的基尼系数根据集合中样本的类分布来计算.例如,给定 m 个类,S 的基尼系数为

$$gini(S)=1-\sum_{j=1}^{m}P_j^2 \tag{3}$$

其中,p_j 是类 j 在 S 中的概率,等于 S 中类 j 的样本数除以 S 的样本总数.

3) 确定阈值的过程递归地用于所得到的每个划分,直到满足用户给定的某个终止条件.

基于基尼系数的离散化方法使用数据分布反映出的信息量特征,可以尽可能地将区间边界定义在准确的位置上,有助于提高概念划分的准确性.

2.2 概念元组的提取

设经过离散化得到的关系表 T 含有 n 个对象,即 $T=\{s_1,s_2,...,s_n\}$,元组 s_i 由 m 个属性 A_j 描述(j=1,2,...,m).属性 A_j 的值域取自有限个可以互相区别的符号组成的域,用 $Dom(A_j)=\{v_{j1},v_{j2},...,v_{jk}\}$ 表示属性 A_j 的值域.可把关系表 T 看作 m 维离散向量空间 T,向量空间 T 上的概念元组实际上是笛卡尔积 $A_1\times A_2\times...\times A_m$ 上的一个 m 元组.

向量空间 T 上的概念元组总数 $sum_c=\prod_{j=1}^{m}|Dom(A_j)|$,由此可见,$sum_c$ 与 m 成指数关系.随着维数的增加,空间 T 上的概念元组总数会成倍增加.为了防止生成过多无关的概念元组,在提取过程中,只生成那些在关系表 T 中有对应对象的概念元组.这样,无论关系表 T 的维数有多大,生成的概念元组总数肯定不会超过表 T 原有的元组数 n.

虽然可以非常简单地生成向量空间 U 上的所有概念元组,然而并不是任意概念元组都有意义,有些就根本不可能在现实世界中存在.因此,挖掘用户可以在概念元组的提取过程中定义一些约束条件,过滤掉那些不满足条件的概念元组.本文主要考虑以下 3 种约束条件.

1) 必要值约束:如果一个概念元组被认为是有意义的,则该元组在某个或某几个属性上取值是确定的.考虑中医小儿肺炎病例数据,如果要有意义,属性“咳嗽程度”上的取值就不能是“无咳”.

2) 并存值约束:在一条合理的数据中,有一些属性值之间存在着共生关系,即同时存在或同时不存在.并存值约束用于指定概念元组在各属性上必须同时出现的值.如果一个概念元组只包含这样一组值中的一个或几个,则该元组不满足并存值约束.

3) 互斥值约束:在一条合理的数据中,有一些属性值之间存在着互斥关系,即两个属性上的值只能出现其中一个.互斥值约束用于指定概念元组在各属性上不能同时出现的值.如果一个概念元组同时包含了这样两个

属性上的互斥值,则该元组不满足互斥值约束.

必要值约束规定了概念元组在一个属性上取值的约束,而并存值和互斥值约束定义了概念元组在多个属性间取值的约束.本文把这些约束称为概念规则,通过用户定义的概念规则,不但可以缩减生成概念元组的规模,而且能够检测原数据集中的孤立点.

概念元组提取算法 CIEA(Concept item extracting algorithm)可以描述为:顺序选择元组 $s_i \in T$,判断 s_i 是否符合概念规则,如果不符合,作为孤立点报告给用户,如果符合,判断 s_i 是否已提取,如果没有,添加到概念元组表中,如果已提取,则选择关系表 T 中的下一个元组进行处理.算法 CIEA 的详细描述如下:

输入:经过概念分段的数据表 T,概念规则集 R.

输出:数据表 T 对应的概念元组表 TC,表 T 中的孤立点集 TO.

方法:

```
Initiate(TC);                          // 初始化概念元组表 TC,将 TC 清空
for(T 中的每个元组 s_i) {
    if(Mismatch (s_i,R)) {             // 如果元组 s_i 不符合概念规则集 R
    Outlier(s_i, TO); //将元组 s_i 添加到孤立点集 TO 中
    }
    else if( No_Item(TC,s_i)) {        // 如果概念元组表 TC 中没有元组 s_i
            Insert_Item(TC,s_i));      // 将元组 s_i 添加到概念元组表 TC 中
    }
}
```

该算法在给定关系表 T 和概念规则集 R 的条件下,至多在 $O(n)$的时间内计算出表 T 对应的概念元组表 TC 和孤立点集 TO,并且$|TC| \le |T|$.

2.3 概念元组聚类

在离散化得到关系表 T 的基础上,经过概念元组的提取,会产生关系表 T 对应的概念元组表 TC.TC 表存储了从 T 表提取出的所有符合规则的概念元组,两个表的结构完全一致.因此,可以把概念元组当作一般的数据库元组进行聚类.概念元组的聚类过程实际上是一个概念概化的过程.聚类得到的概念元组的簇可以看作是一个概化的概念.

2.3.1 概念元组的语义距离度量

设感兴趣的概念元组表 TC 含有 n 个对象,即 $T=\{s_1,s_2,...,s_n\}$,元组 s_i 由 m 个属性 A_j 描述(j=1,2,...,m).属性 A_j 的属性域中元素之间的结构关系因描述对象的不同而不同,可以分为标称(nominal)和序数(ordinal).

D_n 表示无序概念域,$A_j \subseteq D_n$ 表示属性值取自无序概念集合.如地图颜色 D_n={‘红色’,‘黄色’,‘绿色’,‘粉红色’,‘蓝色’}.

D_o 表示有序概念域,$A_j \subseteq D_o$ 表示属性值取自有序概念集合.如比赛排名 D_o={‘冠军’,‘亚军’,‘季军’}.

元组 $s_i \in TC$(i=1,2,...,n)之间的相似性用语义距离来度量,对于不同类型的属性,有不同的度量公式.

1) 标称属性的语义距离:

$$d_n(a_{ik},a_{jk})=\begin{cases}0, & a_{ik}=a_{jk}\\ 1, & a_{ik}\neq a_{jk}\end{cases},\quad i\neq j \tag{4}$$

a_{ik} 和 a_{jk} 是元组 s_i 和 s_j 在第 k 个属性(为标称属性)上的取值,$d_n(a_{ik},a_{jk})$表示元组 s_i 和 s_j 在第 k 个属性上的语义距离,属性值相同时,$d_n(a_{ik}, a_{jk})$为 0,不同为 1.

2) 序数属性的语义距离:

$$d_o(a_{ik},a_{jk})=|a_{ik}-a_{jk}| \tag{5}$$

$d_o(a_{ik},a_{jk})$表示元组 s_i 和 s_j 在第 k 个属性(为序数属性)上语义距离.考虑比赛排名,如果两个元组在该属性上

的取值分别为冠军和季军,则语义距离为 2,如果并列冠军,则语义距离为 0.

3) 两个元组 $s_i,s_j \in TC$ 之间的语义距离度量如下:

$$d(s_i,s_j)=\sum_{k=1}^{ko}\beta_k \mid a_{ik}-a_{jk} \mid +\sum_{k=1}^{kn}\beta_k d_n(a_{ik},a_{jk}) \tag{6}$$

其中,k_o 和 k_n 分别为表 TC 序数属性和标称属性的个数.$\sum_{k=1}^{ko}\beta_k \mid a_{ik}-a_{jk} \mid$ 是元组 s_i 和 s_j 在序数属性上的语义距离的累加和,$\sum_{k=1}^{kn}\beta_k d_n(a_{ik},a_{jk})$ 是元组 s_i 和 s_j 在标称属性上的语义距离累加和,β_k 是用户指定的权重因子向量.

2.3.2 概念元组聚类与概念库的生成

给定概念元组间的语义距离度量公式后,可以使用现有的聚类算法对概念元组表进行聚类.由于提取出的概念元组个数一般要远小于原数据集的个数,所以可以选择那些效果好而时间复杂度较高的算法,以生成良好的聚类结果.

概念元组的聚类结果是一些表示特定概念的簇,每个簇是关系表 T 对应向量空间的子集,根据定理 1 可知,每个簇能唯一确定向量空间 T 上的一个概念.由簇生成概念的方法比较简单,可以通过统计,计算出该簇在各个属性上的值分布矩阵,即可得到它所对应的概念.

2.4 元组分类

在概念元组表的基础上,通过聚类分析与统计,会生成关系表 T 所对应的概念库.最后,需要将表 T 中的元组分配到不同的概念上,以产生最终的聚类结果.

在表 T 中,除了孤立点之外的数据对象,其他对象都有自己对应的概念元组,因此,元组分类的过程实际上就是数据对象向概念元组进而向更高层概念的映射过程.实际做法可有很多,本文给出了一种较简单的方法.

设关系表 T 由 m 个属性 A_j 描述(j=1,2,...,m),给定关系表 T 上的元组 s_i 和概念 c,度量元组 s_i 和概念 c 匹配度的公式如下:

$$match(s_i,c)=\prod_{j=1}^{m}p(a_{ij}) \tag{7}$$

其中,a_{ij} 是元组 s_i 在第 j 个属性上的取值,$p(a_{ij})$表示在概念 c 中,值 a_{ij} 在第 j 个属性上出现的概率.

对关系表 T 中的元组 s_i 进行分类时,首先计算 s_i 与概念库中各个概念的匹配度,将 s_i 划分到与之匹配度最大的概念 c 中.通过该方法,可以简单地将元组进行分类.

3 实验分析

为了客观地验证基于概念的数据聚类模型 CBCM 的有效性,本文分别在模拟生成的数据集和中医小儿肺炎病例数据集上,对模型的概念元组个数、聚类效果、算法执行效率等方面进行了实验分析.

3.1 实验数据集的生成

模式数据集的生成参考文献[12]中的方法,首先在 20 维向量空间中随机产生 10 个聚类点,每点有 20 个坐标,然后在各聚类点的每个坐标点 μ 处分别按正态分布产生[μ–1000,μ+1000]上的整数,形成各聚类点周围随机的记录.实验共生成 100K 个样本数据,其中为各聚类点生成的记录数以及各聚类点所采用的维平均 σ 参数,见表 1.

Table 1 Simulation dataset

表 1 模拟数据集

Cluster	1	2	3	4	5	6	7	8	9	10
Number (k)	30	20	15	10	8	7	5	2.5	2	0.5
Average (σ)	0.5	1.0	1.25	1.75	2.0	1.5	1.0	0.75	0.3	0.2

中医小儿肺炎病例数据集来源于科技部“十五”攻关课题,该课题共采集了 1 072 人的 10K 条小儿肺炎病例数据.为了有效地对聚类模型 CBCM 进行验证,本文采用了基于遗传算法的数据生成技术,含已采集病例共生成 100K 条病例数据.该数据集的特点是维数大(共含 76 维),各维均为标称属性或序数属性,且取值个数少(2~5 个).

3.2 概念元组个数分析

聚类模型 CBCM 从数据集中提取的概念元组个数直接影响着模型的聚类效果和执行效率.因此,分析概念元组数随维数和概念分段数的变化关系,可为模型中参数的选择提供参考.在进行该实验时,采用了属性划分方法:即数据集按属性分成 k 个没有交叉列的子集,所有子集的属性个数相同.测试共进行 k 次,每一次针对一个特定的子集,把所有得到的概念元组数的平均值作为估计的概念元组数.

该实验使用了原始生成的 100K 模拟数据集和经扩充得到的 100K 中医小儿肺炎病例数据集.与模拟数据集不同,中医小儿肺炎病例数据集所采用的概念分段数固定.表 2 和表 3 分别列出了在以上两个数据集上的测试结果.

Table 2 Number of concept tuples on simulation dataset

表 2 模拟数据集上的概念元组数

Dimension number		3	5	7	10	15	20
Segment number	5	79	373	1 280	4 426	15 737	31 155
	10	343	2 547	10 107	28 507	57 866	76 484
	15	838	7 273	23 049	47 146	74 231	89 098
	20	1 659	12 651	32 440	56 498	81 340	93 540

Table 3 Number of concept tuples on pneumonia dataset

表 3 肺炎数据集上的概念元组数

Dimension number	3	5	7	10	20	76
Item number	34	372	5 281	80 165	81 487	81 490

由表 2 和表 3 可以看出,随着维数和每维上概念分段数的增加,可提取的概念元组个数也随之增加.当维数和概念分段数增加到一定程度时,生成的概念元组数已接近于原始记录数.在这种情况下,采用模型 CBCM 进行聚类,不但不能提高算法的执行效率,加上数据预处理、概念元组提取和元组分类等时间,算法的执行效率反而会降低.模型 CBCM 是在保证尽量少丢失信息量的前提下,通过减少要处理的数据量来提高聚类算法的可伸缩性.因此,数据预处理时设定的概念分段数和概念元组提取时采用的维数,要在保证准确率的基础上尽量采用较小的值,以提高算法的执行效率.

3.3 聚类效果分析

使用聚类模型 CBCM 的前提是要保证聚类算法的准确性.为了测试原始聚类和使用模型聚类之间的结果一致性,首先定义聚类匹配度的概念:

$$m(C,C')=\sum_{i=1}^{n} m_{C_i}\, p(C_i) \tag{8}$$

其中,C 和 C'分别表示原始聚类结果和使用模型的聚类结果,C_i 是原始聚类结果中的一个簇,n 为原始聚类结果中的簇个数,$p(C_i)$代表簇 C_i 发生的概率. m_{C_i} 为 C_i 的簇匹配度,可按式(9)计算:

$$m_{C_i}=\max\left(\frac{|C_i\cap C'_j|}{|C_i\cup C'_j|}\right) \tag{9}$$

其中,C'_j 为模型聚类结果中的一个簇(j=1,2,...,m),m 为模型聚类结果的簇个数,$|C_i\cap C'_j|$表示簇 C_i 与 C'_j 中相同元素的个数.由式(9)可知, m_{C_i} 表示原始聚类结果中的簇 C_i 与模型聚类结果中对应的簇 C'_j 之间的一致程度,且 $m_{C_i}\in[0,1]$.据此,由式(8)可知,$m(C,C')$反映了原始聚类结果 C 和模型聚类结果 C'的一致程度,且 $m(C,C')\in[0,1]$,1 代表完全一致,0 代表完全不一致.

实验首先采用 100K 的模拟数据集,在表 3 列出的各概念分段数和各维数形成的点上进行了测试.在进行测试时,采用属性划分方法,并使用软件 SPSS13.0 提供的快速聚类算法,设定聚类数目为 10,最大迭代次数为 800.图 2 给出了模型 CBCM 在各段各维处的聚类匹配度,图 3 给出了 5 段 10 维处聚类结果的各簇匹配度.

图 2 中的横、纵及 z 轴坐标分别表示维数、聚类匹配度和概念分段数.从中可以看出,当维数大于 10,概念分段数大于 5 后,模型的聚类匹配度已接近于 1,再对维数和分段数进行增加,聚类匹配度变化不大.另外,图 2 还显示,维数对聚类匹配度的影响要大于概念分段数.因此,使用模型 CBCM 进行预处理时可以生成较少的概念分段,但概念元组提取时要采用恰当的维数.图 3 显示了 5 段 10 维处模型聚类结果的各簇匹配度,其值分布较均匀,整体聚类匹配度为 0.874 3,和原始聚类结果的一致性较高.

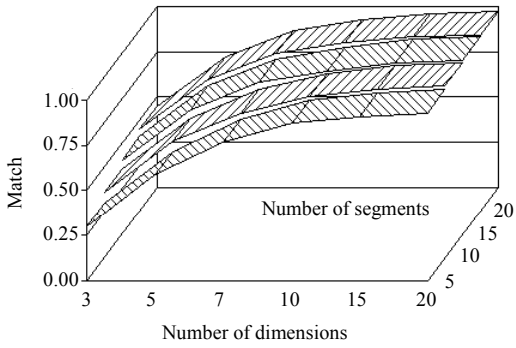


Fig.2 Clustering match
图 2 聚类匹配度

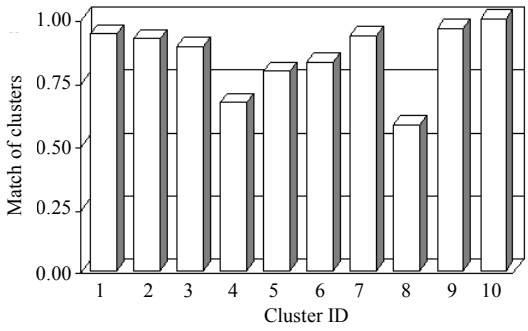
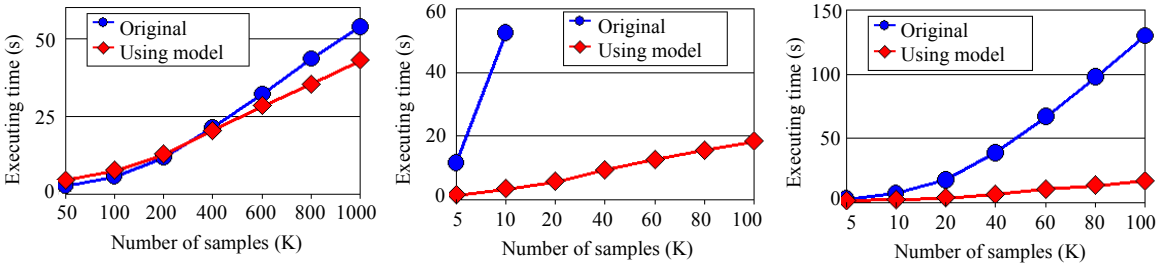


Fig.3 Match of clusters on 5 segments and 10 dimensions
图 3 5 段 10 维的簇匹配度

另外,实验还在原始采集的 10K 中医小儿肺炎病例数据集上进行了测试.聚类时采用数据集固有的概念分段数,当模型使用 7 维属性时,聚类结果匹配度已达到 0.891.从以上结果可以看出,如果模型 CBCM 采用较小的维数和概念分段数,也能达到较理想的聚类效果.

3.4 算法执行效率分析

为了测试聚类模型 CBCM 对聚类算法性能的改进,本文对 SPSS13.0 提供的快速聚类算法和层次聚类算法以及文献[4]中基于密度的 ST-DBCSAN 算法进行了使用模型前后执行时间上的对比分析.使用模型时,采用的概念分段数为 5,维数为 10.实验在模拟数据集上进行测试,以平均随机抽样的方法,快速聚类算法抽取了 50K,100K,200K,400K,600K,800K 和 1M 记录作为数据子样,层次聚类算法和 ST-DBCSAN 算法抽取了 5K,10K,20K,40K,60K,80K 和 100K 记录作为数据子样.分别对数据子样各执行算法 5 遍,计算算法的平均执行时间.图 4 给出了以上 3 种算法使用模型前后的执行时间对比.



(a) Quick clustering algorithm
(a) 快速聚类算法

(b) Hierarchical clustering algorithm
(b) 层次聚类算法

(c) ST-DBSCAN algorithm
(c) 基于密度的 ST-DBSCAN 算法

Fig.4 Contrastive analysis of the clustering model CBCM on executing time
图 4 聚类模型 CBCM 的执行时间对比分析

从图 4(a)可以看出,就基于划分的快速聚类算法来说,针对小数据量,使用模型的执行时间反而要高于不使用模型的聚类时间,然而随着数据量的增大,模型 CBCM 在执行时间上要优于原始聚类算法,但其改进效果并不是很明显.图 4(b)针对 SPSS 提供的层次聚类算法给出了执行时间上的对比.如果不采用模型 CBCM 进行聚类,当数据量大于 10K 后,算法会因内存不足而不能运行.由图 4(b)和图 4 (c)可知,使用模型 CBCM 可以明显降低算法的时间和空间复杂度,从而提高算法的可伸缩性.

由以上实验结果可以看出,模型 CBCM 能够在保证准确性的基础上明显提高聚类算法的可伸缩性,增加算法对庞大数据集的处理能力.然而,如何能够恰当地选择概念分段数和维数等参数,是模型使用的一个难点.

4 结论与未来工作

迄今为止,聚类算法的研究已有很长一段时间,但数据可伸缩性和结果可解释性一直是两个重要的研究方面.本文在形式概念分析的启发下,提出了一种基于概念的数据聚类模型.该模型不同于以往聚类分析方法,主要从描述数据对象的数据本身出发,提取出概念元组作为基本概念,再通过对概念元组的聚类进行概念概化.由于概念元组数一般要远小于原有的数据样本数,所以该模型能够在很大程度上提高数据的可伸缩性.另外,由于该模型基于概念进行分析,这样做可以提高聚类结果的可解释性,同时,也方便与用户进行交互.

尽管模型 CBCM 在诸多方面性能优越,然而如何恰当地选择维数和概念分段数等参数,使模型能够在保证准确率的基础上最大程度地提高算法效率,是模型使用上的一个难点.如何自动或辅助地确定这些模型参数是今后的研究内容.

References:

[1] Chen MS, Han JW, Yu PS. Data mining: An overview from a database perspective. IEEE Trans. on Knowledge and Data Engineering, 1996,8(6):866−883.

[2] Han JW, Kamber M, Wrote; Fan M, Meng XF, Trans. Data Mining Concepts and Techniques. Beijing: China Machine Press, 2001. 232−236 (in Chinese).

[3] Ordonez C, Omiecinski E. FREM: Fast and robust EM clustering for large data sets. In: Kalpakis K, Goharian N, Grossman D, eds. Proc. of the 2002 ACM CIKM Int'l Conf. on Information and Knowledge Management. McLean: ACM Press, 2002. 590−599.

[4] Birant D, Kut A. ST-DBSCAN: An algorithm for clustering spatial-temporal data. Data & Knowledge Engineering, 2007,60(1): 208−221.

[5] Hinneburg A, Keim D. An efficient approach to clustering in large multimedia databases with noise. In: Agrawal R, Stolorz PE, Piatetsky-Shapiro G, eds. Proc. of the 4th Int'l Conf. on Knowledge Discovery and Data Mining. New York: AAAI Press, 1998. 58−65.

[6] Ankerst M, Breunig MM, Kriegel HP, Sander J. OPTICS: Ordering points to identify the clustering structure. In: Delis A, Faloutsos C, Ghandeharizadeh S, eds. Proc. ACM SIGMOD Int'l Conf. on Management of Data. Philadelphia: ACM Press, 1999. 49−60.

[7] Zhang T, Ramakrishnan R, Livny M. BIRCH: An efficient data clustering method for very large databases. In: Jagadish HV, Mumick IS, eds. Proc. of the 1996 ACM SIGMOD Int'l Conf. on Management of Data. Montreal: ACM Press, 1996. 103−114.

[8] Guha S, Rastogi R, Shim K. CURE: An efficient clustering algorithm for large databases. In: Haas LM, Tiwary A, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Seattle: ACM Press, 1998. 73−84.

[9] Wang W, Yang J, Muntz RR. STING: A statistical information grid approach to spatial data mining. In: Jarke M, Carey MJ, Dittrich KR, eds. Proc. of the 23rd Int'l Conf. on Very Large Data Bases. Athens: Morgan Kaufmann Publishers, 1997. 186−195.

[10] Fisher D. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 1987,2(2):461−465.

[11] Bai S. Concept clustering under insufficient knowledge. Chinese Journal of Computers, 1995,18(6):409−416 (in Chinese with English abstract).

[12] Guo JS, Zhao Y, Shi PF. An efficient dynamic conceptual clustering algorithm for data mining. Journal of Software, 2001,12(4):582−591 (in Chinese with English abstract). http://www.jos.org.cn/ch/reader/view_abstract.aspx?flag=1&file_no=20010414&journal_id=jos

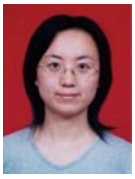
附中文参考文献:

[2] Han JW, Kamber M,著;范明,孟小峰,译.数据挖掘:概念与技术.北京:机械工业出版社,2001.232−236.

[11] 白硕.不完全知识下的概念聚类.计算机学报,1995,18(6):409−416.

[12] 郭建生,赵奕,施鹏飞.一种有效的用于数据挖掘的动态概念聚类算法.软件学报,2001,12(4):582−591. http://www.jos.org.cn/ch/reader/view_abstract.aspx?flag=1&file_no= 20010414&journal_id=jos

张明卫(1979－),男,山东胶州人,博士生,助教,CCF 学生会员,主要研究领域为数据挖掘,服务计算.



刘莹(1981－),女,博士生,助教,主要研究领域为服务计算.

张斌(1964－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为服务计算,数据挖掘,Web 信息处理.

朱志良(1962－),男,博士,教授,博士生导师,主要研究领域为计算机网络与通信,软件架构与信息整合技术,混沌分形与图像处理.