聚类初始中心点选取研究

杨天霞, 王治和, 王 华, 王凌云

(西北师范大学数学与信息科学学院, 甘肃 兰州 730070)

[摘要] 研究了利用已发现的频繁序列模式对序列数据库进行再聚类再发现的问题, 针对已有的 K-均值聚类算法随机选取初始中心点而导致聚类结果不稳定性的缺点, 提出了一种基于 Huffman思想的初始中心点选取算法——K-SPAM (K-means algorithm of sequence pattern mining based on the Huffman Method)算法. 该算法能够在一定程度上减少陷入局部最优的可能, 而且对序列间相似度的计算采用一种高效的"与"、"或"运算, 可极大提高算法的执行效率.

[关键词] K-均值, 序列模式, Huffman树, 聚类, 初始中心

[中图分类号] TP391 [文献标识码] A [文章编号] 1001-4616(2010)04-0161-05

Research of Clustering Initial Center Selection

Yang Tianxia, Wang Zhihe, Wang Hua, Wang Lingyun

(College of Mathematics and Information Science, Northwest Normal University, Lanzhou 730070, China)

Abstract The paper studied the problem of reclustering and rediscovering in the sequence database on the basis of the results of sequential pattern mining. Aiming at this shortcoming that it could lead to the instability of clustering results to select randomly the initial focal points in the existing K-means clustering algorithm, an initial center selection algorithm named K-SPAM (K-means algorithm of sequence pattern mining based on the Huffman Method) algorithm was proposed. It was based on Huffman idea. The algorithm could reduce probability of local optimum to a certain extent. Moreover, a highly efficient "and" and "or" operators were adopted to calculate similarity between pairs of sequences. To do so could greatly improve the execution efficiency of the algorithm.

Key words K-means; sequential patterns; Huffman tree; clustering; initial center

自 1995年 Agrawal和 Srikant两位学者提出序列模式的概念以来[1], 有关序列模式的挖掘得到了广泛的关注, 国内外很多学者对其展开了研究, 使得序列模式的挖掘效率得到了极大的提高. 序列模式挖掘的主要任务是找出随着时间或是特定顺序经常发生的模式, 关于其挖掘算法已经有很多研究. 序列模式发现作为重要的 KDD分支, 在交易数据分析、疾病分析、Web日志分析等领域已经展开了较为广泛的研究和应用.

MacQueen J B在 1967年提出了 K-均值聚类算法[2], 它是到目前为止应用于科学和工业领域中诸多算法中一种极有影响力的技术. 算法首先随机选取 k 个点作为初始聚类中心, 然后计算各个数据对象到各聚类中心的距离, 把数据对象归到离它最近的那个聚类中心所在的类; 再对调整后的新类计算新的聚类中心, 如果相邻两次的聚类中心没有明显变化, 聚类准则函数收敛, 说明数据对象调整结束. 该算法的一个特点是在每次迭代中都要考察每个样本的分类是否正确, 若不正确, 就要调整. 在全部数据调整完后, 再修改聚类中心, 进入下一次迭代. 如果在一次迭代算法中, 所有的数据对象被正确分类, 则不会有调整, 聚类中心也不会有任何变化, 这标志着准则函数已经收敛, 至此算法结束.

然而 k 个初始聚类中心点的选取对聚类结果具有较大的影响, 因为在该算法中是随机地任意选取 k 个点作为初始聚类中心, 初始地代表一个簇. k 个中心点选取的不同将直接影响算法的迭代次数和执行效率. 本文提出了一种基于 Huffman树构造思想的初始中心点选取方法, 该方法解决了随机选取初始中心点

收稿日期: 2010-06-10

基金项目: 西北师范大学 2006- 2010年度重点学科基金 (2007C04).

通讯联系人: 杨天霞, 硕士研究生, 研究方向: 数据挖掘. E-mail: ytxluck@163.com

容易陷入局部最优,以及 K-均值聚类算法对初始聚类中心有严重的依赖性,随机选择初始中心点容易导致聚类结果不稳定的问题.同时,本文对序列间相似度的计算采用相似性度量 Jaccard系数定义序列间的相似性,使得算法执行效率大大提高.

1　基本知识

定义 1[1]　序列(Sequence):是项集的有序表,记作 $< S_1, S_2, \cdots, S_n >$,其中 S_k $(1 \leqslant k \leqslant n)$ 是项集.

定义 2[1]　序列包含(Sequence Contain):给定两个序列 $A = \{a_1, a_2, \cdots, a_m\}$ 和 $B = \{b_1, b_2, \cdots, b_n\}$,如果存在一组整数 $1 \leqslant i_1 < i_2 < \cdots < i_m \leqslant n$,使得 $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i2}$, $\cdots$, $a_m \subseteq b_{im}$,则称序列 A 被序列 B 包含.不包含在任何其它序列中的序列称为最大序(maximal sequence).

定义 3[1]　序列 S的支持数是包含 S的客户序列数目.如果序列的支持数大于等于用户给定的最小支持度阈值(minsup),则称 S是一个频繁序列.

定义 4[3]　序列 S_1和 S_2之间的相似度定义

$$f(S_1, S_2) = \frac{| S_1 \cdot p \cap S_2 \cdot p |}{| S_1 \cdot p \cup S_2 \cdot p |}, \tag{1}$$

其中, $S_1.p$ 表示序列 S_1所支持的序列模式的集合, $S_2.p$ 表示序列 S_2所支持的序列模式的集合.显然, $0 \leqslant f(S_1, S_2) \leqslant 1$.当 $f(S_1, S_2) = 0$时,表示序列 S_1和序列 S_2支持完全不同的序列模式,可以认为 S_1和 S_2之间没有任何的相似性;当 $f(S_1, S_2) = 1$时,表示序列 S_1和序列 S_2支持完全相同的序列模式,可认为 S_1和 S_2非常相似.

2　基于 Huffman树构造思想的序列聚类挖掘算法

本文提出的基于 Huffman树构造思想的 K-均值序列模式聚类挖掘算法是在已挖掘出的频繁序列模式的基础上对序列数据库中的序列再次聚类.由于序列数据的特殊性以及经典的 K-均值聚类算法存在的一些不足,本文对 K-均值算法如何选取 k个初始中心点做了改进,对序列间相似性的计算采用了位图中的"与"和"或"运算,大大提高了算法的执行效率.

2.1　预处理阶段

对原始序列数据库用 SPAM 算法[4]挖掘出全部频繁序列模式,再根据序列-模式的支持关系构造如表 1所示的序列-模式支持关系表.其中, $S_1, S_2, \cdots, S_n$ 表示数据序列, $P_1, P_2, \cdots, P_m$ 表示序列模式,若 S_i支持(包含) P_j,则对应的属性值为 1,否则为 0.每个数据序列可由一个 m 维向量来描述它对序列模式的支持信息.

表 1　序列-模式支持关系

Table 1　Sequence-pattern support relationship

	P_1	P_2	…	P_m
S_1	1	1		0
S_2	0	1		1
…			…	
S_n	0	1		0

2.2　相异度

使用 Huffman思想来选取 k个初始中心是基于对象间相似性的,本文采用序列间的相似度函数 $f(S_1, S_2) = \frac{| S_1 \cdot p \cap S_2 \cdot p |}{| S_1 \cdot p \cup S_2 \cdot p |}$,用相异度矩阵来存放对象之间的相似性,相异度矩阵是一个对象-对象结构.

由序列模式挖掘结果得到序列-模式支持关系表后,要计算序列 S_i和 S_j之间的相似度,此时只需将 S_i和 S_j的 m 个属性分别做传统的"与"、"或"操作即可,可极大提高算法的运行效率.

对应的相异度函数和相异度矩阵定义为:

$$d(S_1, S_2) = 1 - f(S_1, S_2),$$

和

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ d(n,1) & d(n,2) & d(n,3) & \cdots & 0 \end{bmatrix},$$

其中 $d(i, j)$ 表示对象 i 和对象 j 之间的差异, 通常 $d(i, j)$ 为一个非负数, 且有 $d(i, j) = d(j, i)$ 以及 $d(i, i) = 0$ 当对象 i 和对象 j 彼此非常"接近"或非常相似时, 该数据接近 0, 相反该数值越大, 就表示对象 i 和对象 j 越不相似.

2.2 初始中心点的选取

2.2.1 *Huffman* 算法思想[5]

步骤 **1** 根据给定的 n 个权值 $\{W_1, W_2, \cdots, W_n\}$ 构造 n 棵二叉树的集合 $F = \{T_1, T_2, \cdots, T_n\}$, 其中每颗二叉树 T_i 中只有一个带权为 W_i 的根结点, 其左右子树均空;

步骤 **2** 在 F 中选取两棵根结点权值最小的树作为左右子树构造一棵新的二叉树, 且置新的二叉树的根节点的权值为其左右子树上根结点的权值之和;

步骤 **3** 在 F 中删除这两棵树, 同时将新得到的二叉树加入 F 中;

步骤 **4** 重复步骤 2和步骤 3, 直到 F 中只含有一棵树为止, 这棵树便是 Huffman树.

2.2.2 *K-SPAM* 算法初始中心点选取

由于序列数据本身的特殊性, 在对序列数据库使用 K-均值聚类算法进行再聚类时, 对初始中心点的选取上不能直接采用 Huffman树的构造思想, 需要对 Huffman思想做一些改动.

① 在本文的算法中, w_j 指的是前一阶段所构造的序列 - 模式表中描述 S_j 对应的 m 维向量. 根据 Huffman思想, 基于数据相异度, 将数据样本构造成一棵树. 根据算法的实际需要, 在构造树的时候作了改变: 在构造树时, 不用左右子树根结点权值之和作为新的二叉树根结点权值, 而是用左右两个根结点 w-left 和 w-right的按位与的结果 w-result作为新二叉树的根结点权值, 将 w-left和 w-right两个结点删除即可. 由于 w-result代表了 w-left和 w-right共同支持的序列模式, 能够表示它们的相似性, 所以用 w-result来表示新树的根结点很有意义.

② 将构造出来的 Huffman树, 按构造结点的逆序找到 $k-1$ 个结点, 根据图论知识可知, 去掉这 $k-1$ 个结点可将该树分为 k 个子树, 这 k 个子树的根结点即为初始的 k 个聚类中心点.

2.3 K-SPAM 算法描述

设序列数据库中有 n 个数据对象 (序列数据), 对其按 SPAM算法进行频繁序列模式挖掘, 假设挖掘出了 m 个频繁序列模式, 那么每个原序列数据对象就是 m 维, 现要对该 n 个数据对象进行聚类, 聚类数为 k.

K-SPAM 算法伪代码描述:

输入: 数据预处理后的序列 - 模式支持关系矩阵 $\boldsymbol{M}$, 以及用户要求的聚类数数目 k

输出: 聚类的集合.

Step 1: do

Step 2: 对矩阵 $\boldsymbol{M}$ 中的 n 个对象 $S_n = \{s_1, s_2, \cdots, s_n\}$, 选取相异度最小的两点 p, q, $d_{pq} = \min\{d_{ij}, i, j \in 1, 2, \cdots, n\}$, 计算 p 和 q 按位与的结果作为新点放入 S_n.

Step 3: 从 S_n 中删除 p 和 q 得到 $S_{n-1} = \{s_1, s_2, \cdots, s_{n-1}\}$.

Step 4: 计算 S_{n-1} 中序列间的相异度, 得到相异度矩阵.

Step 5: while S_i 中的对象个数大于 1.

Step 6: 至此构造出了关于 n 个对象的一棵树, 记为 G_n. delete G_n 中逆序构造出来的 $k-1$ 个点, 得到 k 个子树, k 个子树的根节点 $c_1, c_2, \cdots, c_k$ 即为算法的 k 个初始中心点.

Step 7: do

Step 8: 分别计算 n 个数据对象与中心点的距离, 并赋给最近的簇.

Step 9: 计算 k 个簇的中心值.

Step 10: while各簇中心值仍发生明显变化.

3 实验分析

实验环境为 PentiumⅣ/Intel 1.73 GHz PC, 512 MB 内存, Windows XP 操作系统和 Microsoft Visual C++ 6.0. 实验数据集采用:

(1) UCI数据库中的 3个数据集 Iris Plants数据集、Wine recognition数据集和 Balance Scale Weight&

Distance数据集[6];

(2) 使用 IBM 数据生成器产生交易数据库 D1C5T3N0105[7].

3.1 *K-SPAM* 算法稳定性实验及结果分析

本节所选的 3个数据集类别个数都是 3, 将 K-均值聚类结果与 K-SPAM 算法聚类结果进行对比, 以下是聚类结果的正确率计算公式:

$$f = \frac{1}{m}\sum_{j=1}^{m}\frac{\frac{n_{j1}}{N_{j1}} + \frac{n_{j2}}{N_{j2}} + \frac{n_{j3}}{N_{j3}}}{3},$$

其中 m 是测试的次数, $\frac{n_{j1}}{N_{j1}}$ 表示第 j次测试中类 1的正确率, 类似地$\frac{n_{j2}}{N_{j2}}$ 表示第 j次测试中类 2的正确率, 同样$\frac{n_{j3}}{N_{j3}}$表示第 j次测试中类 3的正确率. 当一次测试时这 3个值越大, 那么这次测试的平均正确率也就越大. 相反, 如果这 3个值越小, 那么平均正确率也就越小.

在上式的求和计算中对 3个类的正确率分别进行计算, 然后求出每次聚类的平均值. 限于篇幅, 文中只给出 20次聚类测试后平均正确率对比表(见表 2).

表 2　聚类结果对比

Table 2　Comparison of clustering results

Data Set	Mean Accuracy Rate of K-均值	Mean Accuracy Rate of K-SPAM
IRIS Plants	79.19%	89.00%
wine recognion	72.57%	80.10%
Balance Scale Weight & Distance	80.12%	83.45%

从表 2可以明显看出, K-SPAM 算法在聚类结果正确性方面要优于 K-均值聚类算法, 而正确性的提高在一定程度也会提高算法的稳定性. K-SPAM 算法在选取初始中心点时采用 Huffman 方法, 所有每次选取的中心点都比较稳定, 不会出现大的偏差. K-均值聚类算法由于采用随机方法选取初始中心点, 所以每次聚类初始中心点都有可能不同, 还可能会差别很大, 从而导致这种算法聚类结果的正确性会受很大影响, 进而影响到算法的稳定性.

3.2 K-SPAM 算法运行效率实验及分析

实验数据集采用 IBM 数据生成器产生交易数据库 D1C5T3N0105, 数据库包含 5000条交易记录, 1024条数据序列, 然后利用现有的经典序列模式挖掘算法 SPAM 针对不同的支持度挖掘出频繁序列模式如表 3所示.

表 3　SPAM 算法挖掘序列模式结果

Table 3　SPAM algorithm for mining sequential pattern results

最小支持度	0.05	0.08	0.10	0.12	0.15	0.20
序列模式数	426	235	141	95	52	34

实验一: 分析聚类数目对算法迭代次数的影响, 以最小支持度 0.10时的挖掘结果作为输入, 聚类数目分别取 k= 3、6、9、12、15, 如图 1所示. 从实验一可以明显看出, K-SPAM 算法由于采用了 Huffman的思想选取了初始聚类中心点, 所以算法很快收敛于最优值, 迭代次数明显少于 K-均值聚类算法.

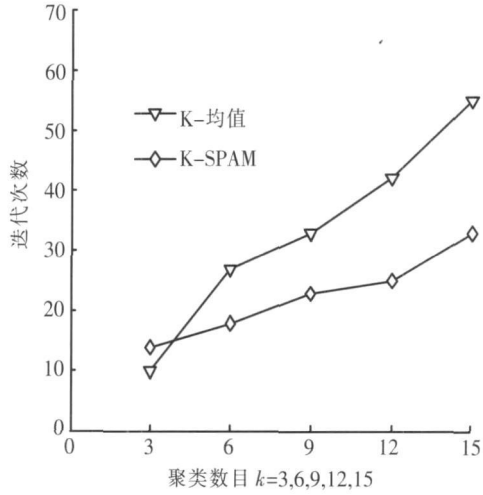

图 1　不同聚类数目下迭代次数比较

Fig.1　Comparison of iterations under different clustering number

文献[3]最先提出了基于已发现序列模式对数据库中序列进行聚类的 POPC (Pattern-Oriented Partial Clustering)算法, 它采用的是一种层次聚类方法.

实验二: 分析序列模式数目对算法性能的影响, 固定聚类数目 k= 9, 以不同最小支持度下的挖掘结果作为输入, 实验结果如图 2所示.

实验三: 分析聚类数目对算法性能的影响, 以最小支持度 0.10时的挖掘结果作为输入, 分别取聚类数目 k= 3、6、9、12、15, 如图 3所示.

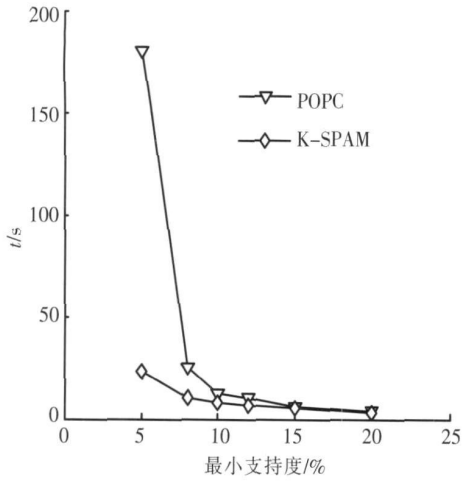

图 2 不同最小支持度下两种算法执行时间比较

Fig.2 Comparison of the execution time of two algorithms under different minimum support

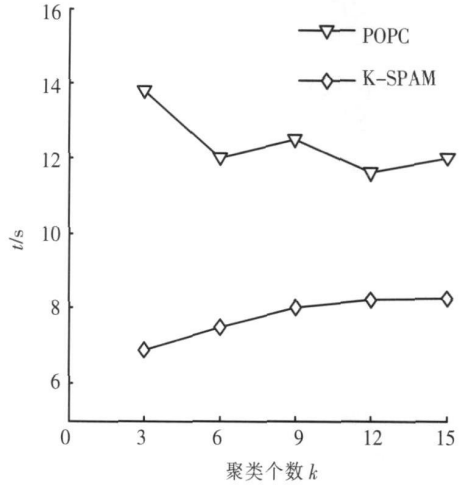

图 3 不同聚类数目下两种算法执行效率比较

Fig.3 Comparison of the efficiency of two algorithms under different clustering number

从实验二和实验三可看出, 本文的 K-SPAM 算法的执行效率要明显优于 POPC 算法. 在实验二中, 当最小支持度很小时, POPC 算法的运行时间明显大于 K-SPAM 算法, 这是由于最小支持度小, 挖掘出的序列模式数相反会很大, 从而导致 POPC 算法的初始聚类数目增多, 使相应的迭代次数也急剧增加, 以及在每次迭代中相异度矩阵的计算量增大, 最终导致执行时间明显高于 K-SPAM 算法的执行时间. 而 K-SPAM 算法由于是 K-均值算法的改进, 继承了它简单易行的优点, 其次是采用了 Huffman 思想能够使算法在执行过程中很快收敛于最优值, 减少迭代次数, 再次是算法中对序列间相异度的计算采用了“与”、“或”运算, 极大地提高了算法的执行效率.

4 结语

本文重点研究了在已挖掘的频繁序列模式的基础上, 再利用划分聚类的 K-均值算法对序列数据进行聚类研究. 文中利用 Huffman 树的构造思想, 对 K-均值算法随机选取初始中心点会导致聚类结果的不稳定性缺点提出了一种新的解决算法 K-SPAM. K-SPAM 算法实现了对包含相似模式的序列数据进行聚类, 通过对聚类初始中心点的选取采用 Huffman 思想, 减少了 K-均值算法的迭代次数, 提高了聚类的稳定性. 并通过实验对 K-SPAM 和 K-均值算法的聚类结果进行比较, 进一步证实了 K-SPAM 算法的优点.

[参考文献]

[1] Agrawal A, Srikant R. Mining sequential patterns[C]//Taipei: Proc of the 11st Int Conf on Data Engineering, 1995: 3-14.

[2] Kaufman L, Roueeeuw P J. Finding Groups in Data: An Introduction to Cluster Analysis[M]. New York: John Wiley & Sons, 1990.

[3] Morzy T, Wojciechowski M, Zakrzewicz M. Scalable hierar-chical clustering method for sequences of categorical values [C]//Proc of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Lecture Notes in Computer Science 2035. New York: Springer-Verlag, 2001: 282-293.

[4] Ayres J, Gehrkeetal J. Sequential pattern mining using a bitmap representation[C]// Proc of the 8th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. Edmonton, 2002: 429-435.

[5] 严蔚敏, 吴伟民. 数据结构[M]. 北京: 清华大学出版社, 2007: 144-145.

[6] UCI数据集[DB/OL]. [2008-03-13]. http://download.csdn.net/source/378926.

[7] IBM Almaden Research Center Quest Data Mining Project[DB/OL]. (1996-03-12) [2007-05-26]. http://www.almaden.ibm.com/cs/quest/syndata.html.

[责任编辑: 丁 蓉]