文章编号:1005-0523(2021)03-0137-05

一种基于图论与最大路径的关联规则挖掘算法

涂晓斌,郭 力,刘晨宁,周 婷,左黎明

(华东交通大学理学院,江西 南昌 330013)

摘要:关联规则的挖掘目标是发现数据项集之间的关联关系或相关关系,是数据挖掘中的一个重要课题。对于超大数据集,传统算法效率较低,对其加以改进,给出了一种基于图论与最大路径的关联规则挖掘算法。该算法将事务集构造成布尔矩阵,经矩阵清理后,将其转换为图的形式,根据关联规则图生成邻接矩阵。当取步长为 k 且 $k>2$ 时,按行从第一个非0元素开始遍历,寻找最大权值路径,此时连接所得元素的行列索引即频繁 $k+2$ 项集。实验结果表明该算法减少了对数据集的扫描次数,针对大数据集,相较于传统的 Apriori 算法能够显著缩短时间,大大提高效率。

关键词:图论;最大路径;布尔矩阵;邻接矩阵;Apriori 算法

中图分类号:TP311 **文献标志码**:A

本文引用格式:涂晓斌,郭力,刘晨宁,等. 一种基于图论与最大路径的关联规则挖掘算法[J]. 华东交通大学学报,2021,38(3):137-141.

DOI:10.16749/j.cnki.jecjtu.20210706.002

An Algorithm for Mining Association Rules Based on Graph Theory and Maximum Path

Tu Xiaobin, Guo Li, Liu Chenning, Zhou Ting, Zuo Liming

(School of Science, East China Jiaotong University, Nanchang 330013, China)

Abstract: The goal of association rule mining is to discover the association or correlation between data item sets, which is an important topic in data mining. For very large data sets, traditional algorithms are inefficient. This paper improves them and gives an association rule mining algorithm based on graph theory and maximum path. The algorithm first constructs the transaction sets into a Boolean matrix. After the matrix is cleaned, the transaction set is converted into the form of a graph, and then an adjacency matrix is generated according to the association rule graph. When the step size is k and $k>2$, traverse from the first non-zero element by line to find the path with the largest weight, and the row and column index of the connected elements is the frequent $k+2$ item set. Experimental results show that the algorithm firstly reduces the number of scans of the data set. Secondly, for large data sets, compared with the traditional Apriori algorithm, it can significantly shorten the time and greatly improve the efficiency.

Key words: graph theory; maximum path; Boolean matrix; adjacency matrix; Apriori algorithm

Citation format: TU X B, GUO L, LIU C N, et al. An algorithm for mining association rules based on graph theory and maximum path[J]. Journal of East China Jiaotong University, 2021, 38(3): 137-141.

收稿日期:2021-03-24

基金项目:国家自然科学基金项目(11761033);江西省科技厅科技项目(20192BBHL80004)

作者简介:涂晓斌(1967—),男,教授,研究方向为工程制图。E-mail:769283941@qq.com。

通信作者:郭力(1998—),女,硕士研究生,研究方向为信息安全。E-mail:1374951417@qq.com。

最初提出关联规则的动机是针对购物篮分析问题,除了应用于顾客模式的挖掘,在其它领域也得到了应用,包括工程、医疗保健、金融证券分析、电信和保险业的错误校验等。Agrawal[1]首先在1993年提出了关联规则概念,随后在1994年提出了经典的Apriori算法,该算法第一次实现了在大数据集上的关联规则提取,利用逐层搜索的迭代方法找出数据库中项集的关系,形成规则。汪曦曦[2]根据事务数据集生成一个布尔矩阵,并得到关联规则图,通过判断节点之间是否存在通路,产生频繁项集,该过程需删除无用的通路。罗丹等[3]在基于矩阵的改进算法中,删除了不能连接的项集和非频繁项集,使矩阵更加简化,但计算过程避免不了新矩阵的生成。宋文慧等[4]利用一个上三角矩阵表示事务之间的关系,但在挖掘频繁项集的过程中生成大量的候选项集。边根庆等[5]扫描一次数据库,生成布尔矩阵,赋予项和事务权重,定义了一个权重支持度,通过大量计算得到频繁项集。杨秋翔等[6]在布尔矩阵的下方添加一行,用于计算项目支持数,删除统计值等于0的事务在矩阵中所对应的行,挖掘过程需要反复进行排序和计算操作。廖纪勇等[7]在生成关联规则的布尔矩阵后,将各列按照支持度计数进行排序,但每生成一组频繁 k–项集,需要重新排序得到新的矩阵。田建勇等[8]利用矩阵表示事务间的关系,但在生成频繁项集的过程中仍然产生候选项集。

这些算法基于矩阵挖掘频繁项集[9-10],在一定程度上有效减少了对数据集扫描的次数,但在挖掘过程中出现了候选项集或新的矩阵,占用大量内存。在超大数据集下,规则生成的效率较低。本文以布尔矩阵为基础,将图论与最大路径问题[11-13]相结合,通过增加步长搜索频繁项集,能有效提高算法效率。

1 Apriori算法流程

Apriori算法的主要思想是找出存在于事务数据集中最大的频繁项集,再利用得到的最大频繁项集与预先设定的最小置信度阈值生成强关联规则。Apriori算法的步骤如下。

1)扫描全部数据集,产生候选1–项集的集合;

2)根据最小支持度,由候选1–项集的集合产生频繁1–项集;

3)当 k>1,重复执行步骤(4)(5)(6);

4)由 L_k 执行连接和剪枝操作,产生候选 k+1–项集的集合 C_{k+1};

5)根据最小支持度,由候选 k+1–项集的集合 C_{k+1},产生频繁 k+1–项集的集合 L_{k+1};

6)若 L 不等于空,则 k=k+1,跳往步骤(4);否则,跳往步骤(7);

7)根据最小置信度,由频繁项集产生强关联规则,结束。

2 基于图和最大路径的优化算法

定义1 最长路径问题。指给定图中找到一条最长长度的简单路径问题。路径的长度可以通过其边数来测量,而在加权图中通过各边的权重之和来测量。

定义2 k–path问题。当规定长度或步长时,找出一条长度为 k 的简单路径,或是在固定步长时,找到一条长度最长的简单路径。

这种问题应用在关联规则中,可以将事务作为结点,当两个事务同时出现时就会相连接而产生一条边,二者同时出现的频率作为边的权值,从而构成一张无向带权图。当规定步数为 k 且 k>2时,所经过的结点数为 k+1,此时,挖掘频繁 k+2–项集的任务即可转变成寻找一条最大权值的 k–path。

2.1 改进的算法

假设数据集 D 中包含 m 个事务 T 和 n 个项目 t,即 $D=\{T_1,T_2,\cdots,T_m\}$,其中 $I=\{i_1,i_2,\cdots,i_k\}$ 为 k–项集,设置一个最小支持度min_support。

Step1 构造关联规则矩阵。将事务集看作列向量,项目集看作行向量,则按以下规则可生成一个 m 行 n 列的事务集布尔矩阵 $\boldsymbol{R}$。

$$a_{ij}=\begin{cases}1, a_{ij}\in T_i\\ 0, a_{ij}\notin T_i\end{cases},(i=1,2,\cdots,n,\ j=1,2,\cdots,m)\rightarrow$$

$$\boldsymbol{R}=\begin{pmatrix}a_{11} & a_{12} & \cdots & a_{1n}\\ a_{21} & a_{22} & \cdots & a_{2n}\\ \vdots & \vdots & \vdots & \vdots\\ a_{m1} & a_{m2} & \cdots & a_{mn}\end{pmatrix}。$$

Step2　矩阵的清理。对于矩阵 $\boldsymbol{R}$ 按列求和,得出各个项目在整个数据库的支持度计数,删去列和小于最小支持度计数的列,剩下的列所对应的项目名即频繁 1–项集 L_1,整理后的矩阵记作 $\boldsymbol{R}'$。

Step3　构造关联规则图。根据矩阵 $\boldsymbol{R}'$ 创建一个简单无向图 $G=(V,E)$,顶点为频繁 1–项集中的所有项目,即 $V=L_1=\{i_1,i_2,\cdots,i_p\},p\leqslant n$。对于 L_1 中的项目,两两组合,在矩阵 $\boldsymbol{R}'$ 中进行按位与运算计数,记的数为 $W(i_si_t)$,即有 $i_si_t=W(i_si_t)$,其中 $s,t=1,2,\cdots,p$ 且 $s\neq t$。对图 G,连接 $i_si_t=W(i_si_t)\neq 0$ 的顶点,则边 i_si_t 的权重为 $W(i_si_t)$,关联规则图构建完毕。

Step4　构造待挖掘矩阵。将关联规则图转换成邻接矩阵 $\boldsymbol{M}_{p\times p}$,此处的图为无向带权图,邻接矩阵 $\boldsymbol{M}_{p\times p}$ 的主对角线一定为 0。在遍历过程中,为避免元素被重复计算,对矩阵 $\boldsymbol{M}_{p\times p}$ 进行约简,将其下三角区域元素用 0 替换,对于小于最小支持度计数的元素也以 0 替换,得到待挖掘的矩阵 $\boldsymbol{M}'_{p\times p}$。

$$\boldsymbol{M}_{p\times p}=\begin{pmatrix}0 & W(i_1i_2) & \cdots & W(i_1i_p)\\ W(i_1i_2) & 0 & \cdots & W(i_2i_p)\\ \vdots & \vdots & \vdots & \vdots\\ W(i_1i_p) & W(i_2i_p) & \cdots & 0\end{pmatrix}\rightarrow$$

$$\boldsymbol{M}'_{p\times p}=\begin{pmatrix}0 & W(i_1i_2) & \cdots & W(i_1i_p)\\ 0 & 0 & \cdots & W(i_2i_p)\\ \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \cdots & 0\end{pmatrix}$$

Step5　频繁 k–项集的生成。对于频繁 2–项集,筛选图 G 中大于 min_support 的边即可。去除待挖掘矩阵的全 0 行,全 0 列。以矩阵的左上角非 0 元素为起点,频繁 k–项集(k>2)所对应的步长为 k–2,因该矩阵为上三角矩阵,在列方向上向下移动,行方向上左右移动,找出权值最大的一条路径。连接该路径中元素所对应的项目,生成频繁 k–项集。当步长等于 max|T|时,停止搜索。

3　算法分析与实验结果

3.1　实例分析

为了进一步说明算法,给出一个简单的算例。已知数据库 D 中包含 10 个事务,$D=\{T_1,T_2,T_3,\cdots,T_{10}\}$,项目集合为 $I=\{a,b,c,d,e,f\}$,最小支持度为 0.2,事务数据集如表 1。

表 1　事务数据集

Tab.1　Transactional datasets

事务	项目集	事务	项目集
T_1	a,c,e	T_6	b,c
T_2	b,d,f	T_7	a,b
T_3	b,c	T_7	a,b,c,e,f
T_4	a,b,c,d,f	T_9	a,b,c
T_5	a,b,e,f	T_{10}	a,c,e,f

1）扫描数据库 D,可构造一个 10 行 6 列的 0–1 事务矩阵 $\boldsymbol{R}$。根据所给条件计算出最小支持度计数:min sup_*count*=min_*support*×|D|=0.2×10=2。计算矩阵各列的列和,删去小于最小支持度计数的列,得到一个新的矩阵 $\boldsymbol{R}'$,也就是频繁 1–项集所对应的矩阵。

$$\boldsymbol{R}=\begin{array}{c}\begin{matrix}a & b & c & d & e & f\end{matrix}\\ \begin{pmatrix}1&0&1&0&1&0\\0&1&0&1&0&1\\0&1&1&0&0&0\\1&1&1&1&0&1\\1&1&0&0&1&1\\0&1&1&0&0&0\\1&1&0&0&0&0\\1&1&1&0&1&1\\1&1&1&0&0&0\\1&0&1&0&1&1\end{pmatrix}\end{array}\rightarrow\boldsymbol{R}'=\begin{array}{c}\begin{matrix}a & b & c & e & f\end{matrix}\\ \begin{pmatrix}1&0&1&1&0\\0&1&0&0&1\\0&1&1&0&0\\1&1&1&0&1\\1&1&0&1&1\\0&1&1&0&0\\1&1&0&0&0\\1&1&1&1&1\\1&1&1&0&0\\1&0&1&1&1\end{pmatrix}\end{array}$$

由此得到频繁 1–项集 $L_1=\{\{a\},\{b\},\{c\},\{e\},\{f\}\}$。

2）根据矩阵 $\boldsymbol{R}'$ 构造一个关联规则图 $G=(V,E)$,其中 $V=\{a,b,c,e,f\}$;对矩阵 $\boldsymbol{R}'$ 各列按位采用“与运算”,得出边集为 $E=\{ab,ac,ae,af,bc,be,bf,ce,cf,ef\}$,对应的权重如表 2。

表 2　边与权重

Tab.2　Edges and weight

边集 E	权重 W	边集 E	权重 W
ab	5	be	2
ac	5	bf	4
ae	4	ce	3
af	4	cf	3
bc	5	ef	3

画出关联规则图如图1。

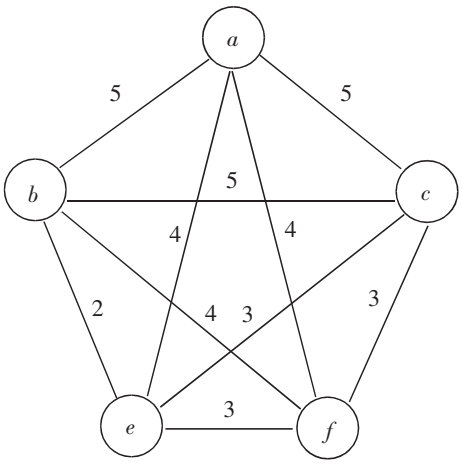


图1 关联规则图

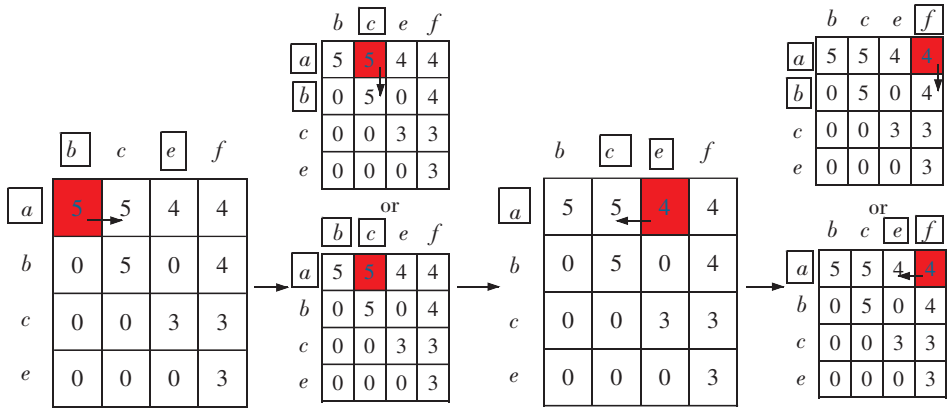
Fig.1 Association rule graph

3）根据关联图生成邻接矩阵 $\boldsymbol{M}$，并将小于最小支持度计数的元素用0替换。

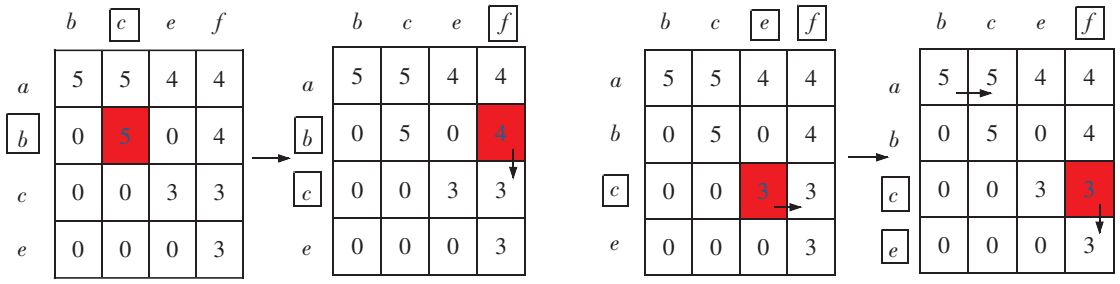
同时可得到频繁2–项集：L_2={{ab},{ac},{ae},{af},{bc},{bf},{ce},{cf},{ef}}。

$$\boldsymbol{M}=\begin{array}{c} \\ a\\ b\\ c\\ e\\ f\end{array}\begin{array}{c}\begin{array}{ccccc}a & b & c & e & f\end{array}\\ \begin{pmatrix}0&5&5&4&4\\5&0&5&2&4\\5&5&0&3&3\\4&2&3&0&3\\4&4&3&3&0\end{pmatrix}\end{array}\rightarrow\boldsymbol{M}'=\begin{array}{c} \\ a\\ b\\ c\\ e\\ f\end{array}\begin{array}{c}\begin{array}{ccccc}a & b & c & e & f\end{array}\\ \begin{pmatrix}0&5&5&4&4\\0&0&5&0&4\\0&0&0&3&3\\0&0&0&0&3\\0&0&0&0&0\end{pmatrix}\end{array}$$

4）频繁 k–项集的生成 k>2。搜寻频繁3–项集，取步长为1。在去除全0行、全0列后，从第1行开始遍历，选择非0出发点 ab，向矩阵的右下角搜索最大路径，移至点 ac，该路径所对应的行、列索引名组合成频繁项集，即{abc}。同理对于下一个点 ac，有两种移动情况，都得到频繁项集{abc}。如图2(A)，最终得频繁项集：{abc}，{ace}，{abf}，{aef}；遍历第2行，出发点为 bc，无法移动，对下一个点进行操作，如图2(B)，得到频繁项集：{bcf}；遍历第3行，如图2(C)，得到频繁项集：{cef}。



(A) 第1行搜索过程



(B) 第2行搜索过程　　(C) 第3行搜索过程

图2 频繁 k–项集的生成过程

Fig.2 The process of generating frequent k–itemsets

最终可得频繁 3-项集 L_3={{*abc*},{*ace*},{*abf*},{*aef*},{*bcf*},{ce*f*}}。当步长等于 max|*T*|时，停止搜索。

3.2　实验结果分析

为测试改进后的算法性能，将本文算法与 Apriori 算法进行关联规则挖掘实验。硬件环境为 Intel(R)Core(TM)i7-10 750 Hz@2.60 GHz，软件环境为 Windows10 操作系统，使用 Python 编程实现。本实验选择了某商品零售企业提供的购物篮数据集(https://download.csdn.net/download/qq_42878458/16601884?spm=1001.2014.3001.5503)，该数据包含 9 835个购物篮数据，169 个商品类别。

考虑当事务数增加时，将改进的算法与传统的 Apriori 算法进行时间的对比，所得实验结果见图 3。

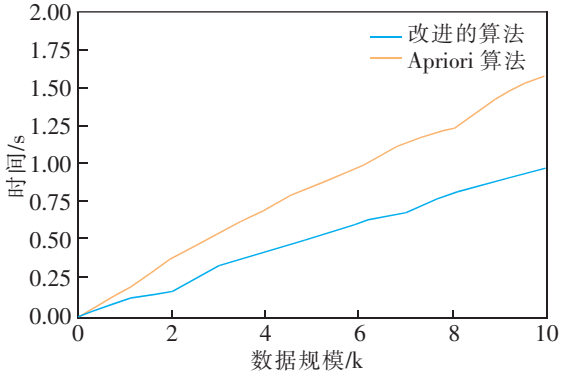


图 3　改进的算法与 Apriori 算法

Fig.3　Improved algorithm and Apriori algorithm

由图 3 可知，与 Apriori 算法相比，随着数据规模的增大，改进的算法挖掘频繁项集的时间在不断减少，有效改进了传统算法挖掘效率低的问题。

4　结论

1）Apriori 算法基于多次扫描事务数据集来执行，在改进的算法中，把事务集看作列向量，项目集看作行向量，只扫描一次数据集，并构造出 0-1 关联规则矩阵，减少了时间和空间消耗。

2）对关联规则矩阵进行约简，删去列和小于最小支持度计数的列；对待挖掘矩阵进行约简，将下三角区域和小于最小支持度计数的元素以 0 替换，避免重复计算。

3）结合最大路径问题，以待挖掘矩阵的左上角非 0 元素为起点，结合图论中的最大路径问题，向右下角搜索权值最大的一条路径，并根据步长确定频繁项集，在时间复杂度上要优于传统的算法，在海量数据的处理方面有一定优势。

参考文献：

[1] AGRAWAL R. Mining association rules between sets of items in large databases，SIGMOD conference[J]. ACM SIGMOD Record，1993，22(2)：207-216.

[2] 汪曦曦. 基于图和矩阵的关联规则挖掘算法[D]. 山东：山东大学，2009.

[3] 罗丹，李陶深. 一种基于压缩矩阵的 Apriori 算法改进研究[J]. 计算机科学，2013，40(12)：75-80.

[4] 宋文慧，高建瓴. 基于矩阵的 Apriori 算法改进[J]. 计算机技术与发展，2016，26(6)：62-64.

[5] 边根庆，王月. 一种基于矩阵和权重改进的 Apriori 算法[J]. 微电子学与计算机，2017，34(1)：136-140.

[6] 杨秋翔，孙涵. 基于权值向量矩阵约简的 Apriori 算法[J]. 计算机工程与设计，2018，39(3)：690-693.

[7] 廖纪勇，吴晟，刘爱莲. 基于布尔矩阵约简的 Apriori 算法改进研究[J]. 计算机工程与科学，2019，41(12)：2231-2238.

[8] 田建勇，石林江. 融合布尔矩阵和项目特性的关联规则挖掘算法[J]. 控制工程，2020，27(6)：1004-1011.

[9] SUN L N. An improved apriori algorithm based on support weight matrix for data mining in transaction database[J]. Journal of Ambient Intelligence and Humanized Computing，2020，11(2)：495-501.

[10] 朱嘉宏，谷岩，胡勇军. Research on Improved association rules algorithm base on matrix[J]. Operations Research and Fuzziology，2019，9(2)：147-155.

[11] 王建新，杨志彪，陈建二. 最长路径问题研究进展[J]. 计算机科学，2009，36(12)：1-4.

[12] 许卫卫，张启钱，邹依原，等. 改进 A~* 算法的物流无人机运输路径规划[J]. 华东交通大学学报，2019，36(6)：39-46.

[13] 刘二根，谭茹涵，陈艺琳，等. 基于改进人工蚁群的智能巡线机器人路径规划[J]. 华东交通大学学报，2020，37(6)：103-107.