基于加权有向图的权频繁模式挖掘算法

封 军，郑 诚，郑小波，肖 云

（安徽大学 计算智能与信号处理教育部重点实验室，安徽 合肥 230039）

摘 要：为解决加权图遍历模式的挖掘问题，提出了一种从加权有向图中挖掘加权频繁模式算法。在该算法中，利用图全局拓扑结构和顶点权值信息评估遍历模式的权支持度，从而将剪枝问题转化成模式可扩展性问题，再利用可扩展模式产生候选模式集。本算法把图顶点权值融合进来，提高了挖掘结果的准确度。实验结果表明，该算法可以有效地进行基于加权向图的权频繁模式挖掘。

关键词：加权有向图；权支持度；关联规则；权频繁模式

中图分类号：TP311 **文献标识码**：A **文章编号**：1674-7720(2010)20-0004-04

Mining weighted frequent patterns algorithm based on weighted graph

FENG Jun, ZHENG Cheng, ZHENG Xiao Bo, XIAO Yun

(Educational Department Key Laboratory of Intelligent Computing & Signal Processing, Anhui University, Hefei 230039, China)

Abstract: In order to better solve the problem about mining weighted frequent patterns based on the weighted graph, the related algorithm is proposed, which is to mine weighted frequent patterns form graph with weight on its vertexes. In this algorithm, we use global topological structure and information to assess the weight support of traversal patterns, and then pruned problem is transformed into patterns scalability problems, and to utilize the extensible patterns to produce the candidate patterns. This algorithm mainly combines with the weight of vertexes to enhance the accuracy of the results. The experimental result shows that it is efficient to mine weighted frequent patterns from the weighted graph.

Key words: weighted graph; weight support; association rule; weight frequent patterns

在现实世界中，许多复杂问题可以被描述成有向图及在其上的遍历问题，其中具有代表性的实例是 WWW 的站点访问[1]，可以把每个网站站点模拟成一个有向图，图中的顶点代表 Web 页面，并代表一个页面到另一个页面的链接，用户在网站上访问的页面记录可以看成在一张图上的遍历，一旦所有的遍历都给出，即可从中挖掘出有用的信息知识。这些信息知识的表现之一就是频繁模式的挖掘发现。而通过这些频繁模式的发现，不仅可以改进网站系统的性能，而且还可以相应地制定出市场决策(例如在用户经常访问的 Web 页面做产品推销和广告发布等)。此外，如超文本访问、人际关系网络构成等，都可以通过图来模拟其基本特征。所以，图遍历模式挖掘一直是一个研究热点。

针对带权值的有向图的挖掘已不再有效，例如 Apriori、FP-增长算法等[4]传统挖掘算法。本文为了解决加权有向图遍历挖掘的问题，提出一种基于有向加权图权频繁模式挖掘算法，简称 GTWF 算法。对于加权图遍历模式的挖掘，传统的图挖掘没有考虑到带有权值的图遍历模式挖掘[5]，对于此项研究，先前的工作大多是与挖掘加权关联规则和其子问题——加权频繁项集有关的，而没有考虑到带有权值的图遍历问题。

1 基本概念

定义 1 加权有向图：给某个有向图的每个顶点加上相应权值，这些权值各自代表相应顶点的重要程度，这样的图称作加权有向图。

例如图 1，如果把这个有向图看作一个 Web 站点的结构图，则它有 7 个结点{A,B,C,D,E,F,G}，每个顶点分

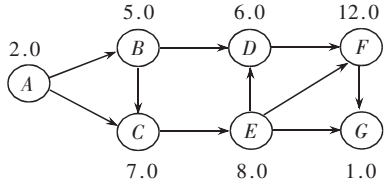


图 1 加权有向图

别代表一个网页，还有 9 个边，他们代表两网页之间的链接，并且在每个结点有一个相应的权值，这个权值可以代表用户对这些 Web 页面的兴趣度[6]，这样的权值可根据用户在 Web 页面上的停留时间、用户对 Web 页面访问行为[2](网页滚动条的次数、被收藏的次数等)或访问 Web 页面的用户数进行分析和评估得出。

定义 2 路径遍历数据库 (Path Traversal Database)，如表 1，把所有关于有向图上的遍历路径综合到一个数据库 D 中，这个的数据库称为路径遍历数据库。例如，用户对某一 Web 站点访问时，把他们各自访问所有 Web 页面按访问的先后顺序进行排列后，得到一系列访问路径，这样的路径称为在站点结构图上的遍历路径。如路径遍历模式 P:<B,C,E,F>，即为图 1 中的一个遍历路径。

表 1 路径遍历数据库 D

id	Traversal
1	<A,B>
2	<B,C,E,F>
3	<A,C>
4	<B,C,E>
5	<A>
6	<A,C,E,D>
7	<D,F,G>
8	<A,B,D>

定义 3 支持度，把某一遍历路径模式 P 在上述数据库 D 中出现的次数记为 count(P)，称为 P 的支持数。关于上述数据库 D，设|D|为所有遍历路径的总数。所以 P 的支持度可以如下定义：

$$\text{support}(P)=\frac{\text{count}(P)}{|D|} \tag{1}$$

定义 4 可扩展模式[3]，若一个模式 P 扩展为长度更长的模式后，有可能成为频繁模式，则称模式 P 为可扩展模式。如果判断一个模式 P 是否具有可扩展性，则 P 要满足下述情况，给定一个最小支持数 minsup，若 k-模式 P 要扩展到 k+1 模式，必须满足 count(P)>=minsup，则称模式 P 具有可扩展的可行性(Feasible)。例如，设定最小支持数 minsup=2，有 2-模式 P:<A，B>，从上述访问路径数据库 D 中得出 count(P)=2，然后可得 count(P)>=minsup，则 P 具有扩展 3-模式<A，B，C>或<A，B，D>的可行性。

定义 5 权支持度[3](weighted support)，如模式 P 的权支持度(wsupport)可定义如下：

$$\text{wsupport}(P)=\left(\sum_{v_j\in P} w_j\right)\text{support}(P) \tag{2}$$

这里 $V=\{v_1,v_2,\ldots,v_n\}$为图的顶点集合，每一个顶点 v_j 有一个权 $w_j\geqslant 0$，$w_j\in W$，$W=\{w_1,w_2,\ldots,w_n\}$。

定义 6 权频繁模式[3]，如果 P 为权频繁模式，则 P 的权支持度(wsupport)一定要大于或等于给定的一个最小权支持度(min$wsup$)。数学表达式如下：

$$\text{wsupport}(P)\geqslant \text{min}wsup \tag{3}$$

所以根据式(1)、式(2)和式(3)，如 P 是权频繁项，可以得到以下的不等式：

$$\text{count}(P)\geqslant \frac{\text{min}wsup\times|D|}{\sum_{v_j\in P} w_j} \tag{4}$$

把式(4)右边进行向上取整，得到的值定义为 P 支持数的较低限度(lower bound)，简称为 sbound(P)。如式(5)：

$$\text{subout}(P)=\left\lceil\frac{\text{min}wsup\times|D|}{\sum_{v_j\in P} w_j}\right\rceil \tag{5}$$

最终，综合式(4)式和式(5)，可以得到以下定义，当 count(P)$\geqslant$sbound(P)时，模式 P 为权频繁模式。

2 基于加权有向图的权频繁模式挖掘算法

本文提出的基于加权有向图的权频繁模式挖掘算法(简称 GTWF 算法)过程与传统 Apriori 算法基本相似，最主要的过程是剪枝步和候选项的产生，不同的是 GTWF 算法提出模式的可扩展性的概念，并把它应用到剪枝和候选项产生操作中，这样可以找出可能具有权频模式的所有模式。下面，对 GTWF 算法的过程进行详细描述。

2.1 剪枝步

在本文算法中，剪枝操作主要是把没有可能成为权频繁模式的候选项剪掉，保留那些有可能成为权频繁模式的候选项。由定义 4 可知，如果模式 P 满足条件 count(P)$\geqslant$minsup，则它具有可扩展性，否则 P 从候选模式集中剪掉。

结合定义 4 可得出下列算法：

算法 1：Prune_Canditate(C_k,G,minsup)Alorithm
Input：候选模式集C_k，加权有向图 G，最小支持数 minsup
Output：剪枝操作后的模式集C_k'

```
1:for (∀ p ⊂ C_k){
2:if (support(p)<=m)then
3:C_k'=C_k-{p};      //P 如果不可扩展则剪掉
4:}
```

2.2 候选项的产生

本文算法的候选项主要是从可扩展模式集中产生的，如果一个可扩展 k-模式<p_1,p_2,…,p_k>的两个(k-1)-子模式<p_1,p_2,…,p_{k-1}>和<p_2,p_3,…,p_k>都是可扩展的，则它们之间有完全向下闭合性。当存在完全向下闭合性时，则可以从可扩展的 k-模式集 C_k 产生一个候选(k+1)-模式集 C_{k+1}，算法描述如下：

算法 2：Gen_candidate(C_k,G)Alorithm
Input：候选模式集C_k，加权有向图 G
Output：新模式候选集C_{k+1}

```
1:C_{k+1}=ø
2:for(∀ p ⊂ C_k){      //P=<p_1,p_2,…,p_k>
3:    for(∀ <p_k,v> ∈ G)
4:        if(v ∉ P)&(Q=<p_1,p_2,…,p_k,v> ∈ C_k)then
5:            P 可扩展为 P'=<p_1,p_2,…,p_k,v>;
6:        C_{k+1}=C_{k+1}=+P';
7:}
```

2.3 基于有向加权图权的频繁模式挖掘算法(GTWF算法)步骤

GTWF算法的主要过程与Apriori算法基本相似,就是把剪枝操作算法与候选模式产生算法结合在一起。下面是GTWF算法的详细步骤:

算法3:GTWF Algorithm

Input:路径遍历数据库D,加权有向图G,
最小支持数$minsup$,最小加权支持数$minwsup$

Output:权频繁模式表L

//步骤1,找出权频繁模式的最大可能长度u

1: u=max(length(P_{id})),$P_{id}\in D$;

//步骤2,初始化长度为1的候选模式

2: $C_1=V(G)$

3: For ($k=1;k<=u\&\&C_k<>\varnothing;k++$){

//步骤3,计算支持数count(P)

4: For each $P_{id}\in D$

5: For each $P\in C_k$

6: If ($P\subset P_{id}$) then

7: Count(p)++;

//步骤4,确定权频繁模式

8: L={$P|P\in C_k$,count(P)>=sbound(P)}

9: If ($k<u$) then {

//步骤5,剪枝操作,得出候选模式集

10: C_k=Prune_Canditate(C_k,G,m); //采用算法1

11: C_{k+1}=Gen_candidate(C_k,G); //采用算法2

12: }

13: }

2.4 实例分析

通过图1和表1数据做GTWF算法相应的实例分析。假设最小权支持度$minwsup=5$,最小支持数$minsup=2$。下面根据GTWF算法步骤做具体实例分析。

(1)根据算法步骤1,扫描路径遍历数据库D,得出权频繁模式的最大可能长度$u=4$,总的数据库记录数$|D|=8$。

(2)根据算法步骤2,初始化长度为1的候选模式如下:

C_1={<A>, <B>, <C>, <D>, <E>, <F>, <G>}

(3)表2、表3、表4和表5是根据算法步骤3、步骤4、步骤5循环执行所得出的结果。

得出最终的权频繁模式集为:{<C,E>,<B,C,E>}。

3 实验分析

本实验是在Pentium4 3.0 GHz,内存为1GB的PC上进行,算法的实现环境为VC++6.0和SQL Server 2005。实验数据是合成数据,实验数据中加权图的顶点数(v)最多有500个,顶点权值(w)范围为$0<w\leqslant 12$。合成路径遍历数据库$|D|=10\,000$,最小权支持度($minwsup$)可分别为1,2,3,4,5。最小支持数(m)可分别为2,3,4,5。

表2 一项模式集

Pattern(P)	Count(P)	Sbound(P)	Weight frequent	Feasible
<A>	5	20		√
<B>	4	8		√
<C>	4	6		√
<D>	3	7		√
<E>	3	5		√
<F>	2	4		√
<G>	1	40		

表3 二项模式集

Pattern(P)	Count(P)	Sbound(P)	Weight frequent	Feasible
<A,B>	2	6		√
<A,C>	2	5		√
<B,C>	2	4		√
<B,D>	1	4		
<C,E>	3	3	√	√
<D,F>	0	–		
<E,D>	1	4		
<E,F>	1	40		
<F,G>	1	–		

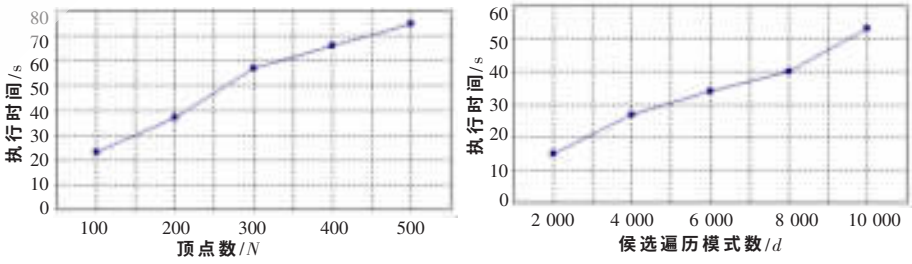
表4 三项模式集

Pattern(P)	Count(P)	Sbound(P)	Weight frequent	Feasible
<A,B,C>	0	–		
<A,C,E>	1	3	√	
<B,C,E>	2	2	√	√

表5 四项模式集

Pattern(P)	Count(P)	Sbound(P)	Weight frequent	Feasible
<A,B,C,E>	0	–	–	

实验研究了不同的最小支持数阈值、最小权支持度阈值、候选遍历模式数、图的顶点数与运行时间的关系。图2、图3分别进行算法性能评估。图2(a)主要说明不同顶点数与执行时间的关系——随着图顶点数的增加,算法执行时间也是递增的。图2(b)主要说明候选遍历模式数与执行时间的关系——随着候选遍历模式数目的



(a) 当$minwsup=5$,最小支持数$m=2$时,不同顶点数,算法的执行时间

(b) 当$minwsup=5$,最小支持数$m=2$时,不同候选遍历模式数,算法的执行时间

图2 算法性能评估

增加，执行时间也是呈递增状态。图3随着最小支持数(m)的增加，执行时间将减少，因为当最小支持数增加时，可扩展模式数就越来越少，相应的剪枝等操作也少了，所以执行时间就变少。另外图中还说明随着最小权支持度(min$wsup$)的增加，执行时间也减少，因为随着最小权支持度的增加，从算法循环一开始，可确定的权频繁模式就减少，直接导致可供下次循环的候选模式也减少了，从而减少了搜索和剪枝等操作，所以执行时间就减少。

本文主要分析讨论了通过加权有向图的遍历来挖掘权频繁模式的问题，提出了解决此问题的GTWF算法。在算法中，利用权支持度、可扩展模式和权频繁模式等概念，并通过剪枝操作和候选模式产生操作来实现算法对图遍历模式的挖掘，最后通过实验对算法的性能进行了验证。实验结果表明算法在可伸缩性与执行性能等方面具有较好的效果。

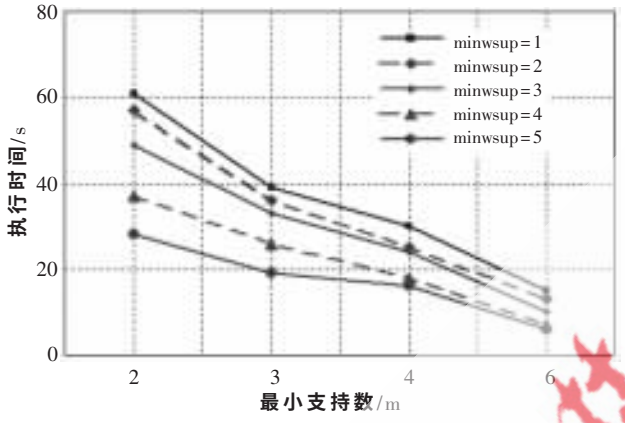


图3 在不同的最小支持数上，关于每个 $winwsup$，算法的执行时间

参考文献

[1] HEYDARI M, HELAL R A, GHAUTH K I. A graph-based Web usage mining method considering client side data [C]. 2009 IEEE.

[2] TAO Yu Hui, HONG Tzung Pei, SU Yu Ming. Web usage mining with intentional browsing Data[C].2007 Elsevier Ltd.

[3] LEE S D, PARK H C. Mining weighted frequent patterns from path traversals on weighted graph[C]. Department of Computer Engineering, Korea Maritime University, Korea

[4] 张云涛，龚玲. 数据挖掘——原理与技术[M].北京:机械工业出版社，2004.

[5] 韩家炜，KAMBER M. 数据挖掘：概念与技术[M].北京：机械工业出版社,2001:351-364.

[6] 郭岩,白硕,杨志峰,等.网络日志规模分析和用户兴趣挖掘[J].计算机学报，2005,28(9):1483-1496.

(收稿日期：2010-07-30)

作者简介：

封军，男，1983年生，硕士研究生，主要研究方向：Web数据挖掘和数据库技术。

郑诚，男，1964年生，副教授，主要研究方向：数据挖掘、数据库技术与语义Web。

郑小波，男，1983年生，硕士研究生，主要研究方向：语义Web和数据挖掘。

肖云，女，1985年生，硕士研究生，主要研究方向：数据库与Web技术。