基于概念格的 Web 日志路径挖掘算法

杨 飞

(重庆青年管理干部学院计算机系 重庆400013)

摘 要 路径挖掘适用于探索用户沿超连接寻找和浏览网页的规律,而 Web 日志的完美结构使挖掘更加容易和有效。由二元关系导出的概念格作为一种非常有用的形式化工具,体现了概念内涵和外延的统一,反映了对象和特征间的联系以及概念的泛化与例化关系,因此非常适于发现数据中潜在的信息。本文通过概念格模型,提出了一种 Web 日志的路径挖掘算法,并进行了相关的分析与展望。

关键词 Web 挖掘,Web 日志挖掘,关联规则,路径挖掘,概念格

An Algorithm for Path Mining of Web Log Based on Concept Lattice

YANG Fei

(Department of Computer, Chongqing Youth Management Cadre College, Chongqing 400013)

Abstract Path Mining is used to discover the regularities when Web users browse and select Web pages along hyperlinks. The perfect structure of Web log make Web mining easier and more efficient. Concept lattice, induced from a binary relation between objects and features, is a very useful formal tool and has been used in many fields. It realizes the unification of concept intension and concept extension, represents the association between objects and features, and reflects the relationship of generalization and the specialization among concepts, so it is fit for discovering the potential information of the data. In this paper, based on the Concept lattice, an algorithm of Web log path mining is presented. And the related analysis and expectation are given.

Keywords Web mining, Web log mining, Association rules, Path mining, Concept lattice

1 引言

近来,国际上很多人利用对 Web 信息的挖掘来提高 Web 的功能。Web 路径挖掘便是其重要任务之一。众所周知,Web 网页都是通过彼此的 Hyperlink(超连接)相互联系,用户一般都是从特定的根节点出发,通过 Hyperlink 方式检索,希望能通过最短的路径找到兴趣的网页。所以作为网站设计者,了解用户的访问路径,挖掘用户的访问模式,对于改善网站的设计有重大作用。

而 Web 是一个松散的分布式信息系统,从理论上讲,对其挖掘是困难的,获取的知识是不可靠的。然而,Web 日志却有完美的结构,每当用户访问 Web 站点时,所访问的页面、时间、用户 ID 等信息,在日志中部有相应的记录。分析 Web 日志,能快速方便地对用户行为进行有效的挖掘。

基于以上两点,本文在关联规则的基础上给出基于概念格的算法,并使其应用于 Web 日志的路径挖掘。本文第2节阐述了关联规则与路径挖掘的基本概念,第3节阐述了概念格基本概念,第4节给出了基于概念格的 Web 日志路径挖掘算法,第5节进行了结论与相关的展望。

2 关联规则与路径挖掘

2.1 关联规则

设 $I=\{i_1,i_2,\cdots,i_m\}$是由 m 个不同的项目组成的集合。给定一个事务数据库 D,其中的每一个事务 T 是 I 中一组项目的集合,即 $T\subseteq I$,T 有一个唯一的标识符 TID。若项集 $X\subseteq I$ 且 $X\subseteq T$,则事务 T 包含项集 X。一条关联规则就是形如 $X\Rightarrow Y$ 的蕴涵式,其中 $X\subseteq I$ 且 $Y\subseteq I$,$X\cap Y=\phi$。关联规则 $X\Rightarrow Y$ 成立的条件是:①它具有支持度 s,即事务数据库 D 中至少有 s%的事务包含 $X\cup Y$。②它具有置信度 c,即在事务数据库 D 中包含 X 的事务至少有 c%同时也包含 Y。

关联规则的采掘问题就是在事务数据库 D 中找出具有用户给定的最小支持度 minsup 和最小置信度 minconf 的关联规则,其具体如 Apriori 算法[1]。采掘关联规则问题可以分解为以下两个子问题[1~2]:

① 找出存在于事务数据库中的所有大项集。项集的支持度 support(X)不小于用户给定的最小支持度 minsup,则称 X 为大项集(large itemset)。

② 利用大项集生成关联规则。对于每个大项集 A,若 B,B,量 Support(A)/Support(B)>=minconf,则有关联规则 B=>(A−B)。

目前大多数研究集中在第一个子问题上。以下我们对算法的介绍也集中在第一个子问题上。

2.2 路径挖掘

路径挖掘[3]能反映用户访问页面的次序,其适合于访问路径挖掘的用户会话模型为:

$s=\langle ip_s,id_s,|(l_1^s.url,l_1^s.time),\cdots,(l_m^s.url,l_m^s.time)|\rangle$,对于$1\leqslant i\leqslant j\leqslant m$,$l_i^s\in L$,$l_i^s.ip=ips$,$l_i^s.id=id_s$,$l_i^s.time<l_j^s.time$。

把页面集 $p=\langle u_1,u_2,\cdots,u_v\rangle$,$u_i\in U$ 称为路径,长度为 v−1,记为 $p^{(v-1)}$,设路径 $p=\langle u_{i1},u_{i2},\cdots,u_{ik}\rangle$,若存在$1\leqslant j\leqslant m-k+1$,使得 $l_j^s.url=u_{i1},\cdots,l_{j+k-1}^s.url=u_{ik}$。则称路径 p 在会话 s 中出现,路径 $p=\langle u_1,u_2,\cdots,u_v\rangle$,$p'=\langle u_{i1},u_{i2},\cdots,u_{ik}\rangle$,

若存在$1\leqslant j\leqslant v-k+1$使得$u_i=u_{i1},\cdots,u_{i+k-1}=u_{ik}$，则称 p' 为 p 的子路径，显然，若 p 在 s 中出现，则 p 的任何子路径都在 s 中出现。

设会话集为$s=(s_1,s_2,\cdots,s_n)$，路径 p 在 s 中的支持度为$sup(p)=|\{s/s\in S$ 且 p 在 s 中出现$\}|/n$。路径挖掘就是要找出支持度大于最小支持度 minsup 的路径，称为频繁路径[4]。

路径$p=\langle u_1,u_2,\cdots,u_v\rangle$，记$\{p\}=\{u_1\cup u_2\cup\cdots\cup u_v\}$若 p 为频繁路径，则 p 的任何子集也为频繁路径，并且$\{p\}$定为频繁路径集，路径集合 A1和 A2的连接$A1\rightarrow A2=\{p/p=\langle u_1,u_2,\cdots,u_v\rangle$并且存在$1<i<v$使得$\langle u_1,u_2,\cdots,u_i\rangle\in A1,\langle u_1,u_2,\cdots,u_v\rangle\in A2\}$长度为 k 的路径集合记为$A^{(k)}$，显然$A^{(0)}\in U$，$A^{(1)}\subseteq(A^{(0)}\times A^{(0)})$当$k>1$时$A^{(k)}\subseteq(A^{(k-1)}\rightarrow A^{(1)})$。

下面是路径挖掘的基本算法：

输入 会话集合最小支持度
输出 频繁路径集合
1. $A^{(0)}=\{u/u\in U$ 且 $\{u\}$为频繁项目集$\}$
2. 对于每一个 $p\in(A^{(0)}\times A^{(0)})$，且$\{p\}$为频繁项目集，如果 sup(P)>minsup，则把 p 加入 $A^{(1)}$
3. FOR k=2 TO s 中最大会话长度
{
对每一个$p\in(A^{(k-1)}\times A^{(1)})$且$\{p\}$为频繁项目集，如果 sup(p)>minisup，则把 p 加入 $A^{(k)}$
}

在下文中我们将根据以上基本算法给出基于概念格的路径挖掘算法。

3 概念格的基本概念

概念格(Concept lattice)，又为 Galois 格，由 R. Wille 于1982年提出[5]，概念格的每个节点由两部分组成：外延，概念所覆盖的实例；内涵，概念所覆盖实例的共同特征。概念格通过 Hasse 图生动简洁地表达了概念之间的泛化和特化关系，关于概念格的比较详尽的形式化描述可以参考文[6～8]。

假设给定形式背景(context)为三元组 T=(O,D,R)，其中 O 是事例集合，D 是描述符(属性)集合，R 是 O 和 D 之间的一个二元关系，则存在唯一的一个偏序集合与之对应，并且这个偏序集合产生一种格结构，这种由背景(O,D,R)所诱导的格 L 就称为一个概念格。格 L 中的每个节点是一个序偶(称为概念)，记为(Y,X)，其中 Y 是幂集 P(O)中的事例集合，称为概念的外延；$X\in P(D)$是 Y 中所有事例的共同描述符的集合，称为概念的内涵。每一个序偶关于关系 R 必须是完备的，即只有最大扩展的序偶才出现在概念格结构中，一个序偶$(Y,X)\in P(O)\times P(D)$关于关系 R 是完备的，当且仅当满足性质：

(1)$Y=\{y\in O|\forall\ x\in X,xRy\}$

(2)$X=\{x\in D|\forall\ y\in Y,xRy\}$

在概念格节点间能够建立起一种偏序关系，给定$H_1=(Y_1,X_1)$和$H_2=(Y_2,X_2)$则$H_1<H_2\Leftrightarrow X_1\subset X_2$，领先次序意味着$H_1$是$H_2$的父节点或称直接泛化。实际上，格中集合$X_1$和$X_2$之间存在着对偶关系。即$X_1\subset X_2\Leftrightarrow Y_2\subset Y_1$从而$H_1<H_2\Leftrightarrow Y_2\subset Y_1$。所以，概念格本质上是相互联系的两个格。格的 Hasse 图是根据偏序关系产生的：如果$H_1<H_2$并且在格上不存在另一个元素H_3使得$H_1<H_3<H_2$。则从H_1到H_2就存在一条边。Hasse 图揭示了概念的内涵与外延之间的泛化关系和特化关系，可作为数据分析与知识获取的一种工具。

4 基于概念格的 Web 日志路径挖掘算法

路径挖掘目的就是挖掘各类用户可能的网页浏览顺序，以便网站能够进行自动重排。这里我们采用最大向前路径(MFP，Maximal Forward Path)的思想[9]来对原始 Web 日志库进行处理，其出发点在于能识别用户事务中的最大向前路径，能在日志库中抽取有效的用户访问路径，而消除反向关联的影响。其挖掘全过程为[10]：

①从 weblog 中寻找所有的 MFP。

②从找出的 MFP 中求出频繁关联路径序列。

③从频繁关联路径中找出最大频繁关联路径。

如现已知日志中一记录为 ABDBEHKLAFCGIJ，找出 MFP 为{ABD，ABEHK，ABEHL，ACF，ACGI，ACGJ}如图1所示。

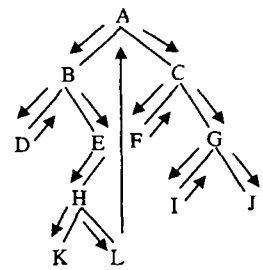

图1 访问路径

在找到了所有的 MFP 后，我们给定一个频繁关联最小支持度的阈值 minsup，如果某一最大 MPF 的出现次数到达该阈值，则发现一个频繁关联路径序列，在基于概念格模型中，我们给出了以下算法[10～11]：

```
输入：概念格 L；等待加入的事物 T；最小支持度的阈值 minsup；指针矩阵 M；每个元素为概念格 L 中的顶层。
输出：更新后的概念格 L；指针矩阵 M；频繁关联序列 S。
//P(X)为节点 X 的所有事例公有描述符的子序列
//sup(X)为节点 X 的支持度
//
//processed 是记录处理过的节点
Processed=0;
//选出所有与 T 相交的有效节点
Set=SelectNodes(L,M,T);
for(X∈Set)
 for(ascending length(P(X)))
  {//如果 H 为 T 的祖先节点，就只改支持度信息，不生成新信息
   if (P(X)⊆T)
   {
     sup(X)=sup(X)+1; //支持度加1
     //如果满足大于最小支持度的阈值
     if(sup(X)>minsup)
      {
       //是否已经放到频繁序列中，如无，放入频繁序列集中
       for(X'∈S)
        if(P(X')!=p(X))S=S∪{X};
      }
       //标记为处理过
       Processed=Processed∪{X};
   }
    else
   {
     //如果 X 不是 T 的祖先节点，生成新的节点插入 L
     Common=P(X)∩{T}
     //Common 为 X 和 T 的交集，现循环处理其中每个公共子序列
     do
     {
       Element=Common. get_element;
       if(X∉Processed && P(X)=Element)
        {
          //构造新节点
          Creat New Node N=(sup(X)+1,element);
          //标记为处理过
          Processed=Processed∪{N};
           //如果满足大于最小支持度的阈值
             if((sup(X)+1)>minsup)S=S∪{N};
          //连接两点，N 成为 X 的父节点
          connect(X,N);
          X. Parent=N;
          //在所有 X 的祖先节点中找到 N 的父节点
          for(tmp∈ancestors of X)
            { if(tmp==N. Parent)
              { Parentset= Parentset∪{tmp};}
```

```
}
while(Parentset!=ϕ)
 {
  A=Parentset.get_element;
  if( P(A)⊆Element)
   {
    //连接两点,A 成为 N 的父节点
    connect( A,N);
    N.Parent=A;
    //删除无效边
     if(A==X.Parent)
      {
       Disconnect(A,X);
       Delete X.Parent;
      }
   }
 }
 for(x∈P(T))
 //更新指针矩阵 M
   Update M according to the updated Lattice L;
 }while(Common!=ϕ)
}
}
```

算法分析:

在概念格的节点子序列 P(X)与等待处理的总事务 T 之间的关系为:

1. P(X)⊆T。这种情况下不生成新的节点,只是增加节点的支持度 sup(X)+1;

2. P(X)∩T≠ϕ。这种情况下要产生新的节点,通过搜索将其插到概念格 L 的正确位置中,并且连接好父节点和子节点,删除无效边;

3. P(X)∩{T}=ϕ。这种情况下为无关节点,我们通过指针矩阵 M,利用过程 SelectNodes 用递归方式求出 L 中所有与 T 内涵交集的有效节点并存入 Set 中,这样就可以不访问格中无关点,提高效率。

如图1,已知 MFP 为{ABD,ABEHK,ABEHL,ACF,ACGI,ACGJ},如果我们设定 minsup=2,算法执行后最终的 Hasse 变换图为图2。

其中频繁访问序列集为 S={ A,AB,AC,ABEH,ACG}

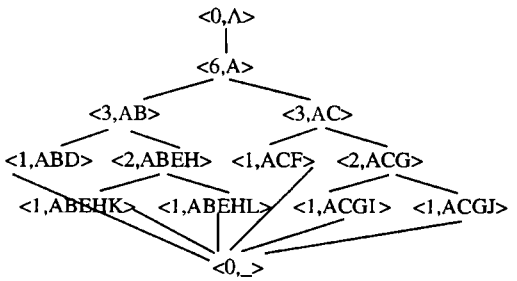

图2 本算法产生的最终概念格 Hasse 图

算法特点:

1. 该算法的可增性

本算法已经把所有事务 T 整理记载入概念格 L 中,使以后的更新操作只需要对该概念格 L 进行局部更新,不需要重新生成概念格。

2. 对 minsup(最小支持度的阈值)的变动适应性

当概念格建好后,所有格中的偏序关系已经建立,对于每个节点的支持度可以确定,所以当最小支持度阈值变动后,只需要在格中扫描该阈值的适配值,而不需要重新生成格。

结论与展望 本文在介绍了关联规则、路径挖掘和概念格的基础上讨论了 Web 日志中关联规则的路径挖掘问题,并提出了一种基于概念格的高效算法。

不同职业的人群,访问同一站点的目的是不一样的,但相同职业的人,往往具有共性。通过路径挖掘,我们可以看出,每类路径代表了某类职业人员访问站点的共同目的,因而可以把每一类路径所访问过的页面集中放在新的 Web 页面中,由站点管理者分析新 Web 页面的特点,赋予相关的标题,不同职业的人群可以只访问与自己有关的主题页面。另一方面,可以自动改进 Web 站点的链接结构。设 $p=\langle u_1,u_2,\cdots,u_x\rangle$ 为任一频繁路径,对于 $1\leqslant i\leqslant j\leqslant x$,若没有 u_i 到 u_j 的超链接,则增加 u_i 到 u_j 的超链接。设 $p=\langle u_1,u_2,\cdots,u_x\rangle$ 为任一频繁路径,若 $\sup(\{p\})/\sup(\{u_x\})>1$,则认为用户要访问的是 u_x,增加 u_1 到 u_x 的链接,这样就建立了自适应 Web 站点[4,12]。

在 Web 的应用和规模快速增长下,把数据挖掘技术应用于 Web 是一个极具挑战性的研究方向。至今在该领域仍有许多值得探讨的问题,主要有:多站点 Web 日志的挖掘;发现模式的可视化分析;把对 Web 页面的标记信息、文本信息和发现的用户访问模式,集成于其他数据仓库中,提供更加丰富完善的功能等,这些也将是下一步要研究解决的问题。

参 考 文 献

1 Agrawal R, Imielinski T, Swami A. Mining associations between sets of items in large databases [A]. In: Buneman Peter, ed. Proc. of the 1993 ACM-SIGMOD Intl. Conf. on Management of Data 1993. 207

2 Agrawal R, Srikant R. Fast algorithms for mining association rules. In: Proc 20th Int Conf Very Large Databases. 1994. 487～499

3 Borges J, Levene M. Mining association rules in hypertext databases. http://citeseer.nj.nec.com/article/borges99 mining.html. 2001-01-21

4 宋爱波,胡孔法,董逸生. Web 日志挖掘. 东南大学学报,2002,32(1):17～18

5 Wille R. Restructuring lattice theory: An approach based on hierarchies of concepts. In: I Rival, ed. Ordered Set. Dordrecht Boston: Reidel, 1982. 445～470

6 Godin R, Missaoui R, Alaui H. Incremetal concept formation algorithms based on Galois (concept) lattices. Computational Intelligence, 1995, 11(2): 246～267

7 Nourine L, Raynaud O. Afast algorithm for building lattices. Information Processing Letters, 1999, 71(5-6): 199～204

8 王志海,胡可云,胡学刚,等. 概念格上规则提取的一般渐进式算法. 计算机学报,1999,22(1):60～70

9 Chen M-S, Park J S, Yu P. Efficient data mining for path traversal patterns. IEEE Trans on Knowledge and Data Engineering, 1998, 10(2): 209～221

10 金阳,左万利. 有序概念格与 WWW 用户访问模式的增量挖掘. 计算机研究与发展,2003,40(5):677～679

11 谢志鹏,刘宗田. 概念格与关联规则发现. 计算机研究与发展,2000,37(12):1415～1421

12 Perkowitz M, Etzioni O. Adaptive Web sites: automatically synthesizing Web pages. http://citeseer.nj.nec.com/perkowitz98adaptive.html. 2001-03-11