关联规则在网站结构优化中的改进算法

Improvement Algorithm of Association Rules for the Structure of Website

井福荣 （江西理工大学 信息工程学院 江西 赣州 341000）
谢辅雯 （江西理工大学信息工程学院2005级研究生 341000）

摘要：关联规则在网站结构优化中的改进，使用 Apriori 算法，通过日志挖掘发现网页间的关联关系，使网站更趋合理，便于用户访问。针对网站超链接结构的一条超链接只能建立在两个网页上的特点，该算法使用逐层搜索的迭代方法，即只需发现频繁集并找出所 2 – 项集，即可降低 Apriori 的时间复杂度。

关键词：网站结构优化 日志挖掘 关联规则 Apriori 算法

1 引言

网站结构的设计直接影响网站的访问效率。并且，用户所关心的内容会随时变化[1]。发现用户使用的模式，可以为改良网页的结构设计提供良好的建议。Web 数据挖掘是在因特网上抽取并发现有价值知识的技术。它可以对网络内容，网络结构，用户行为进行数据挖掘，以期得到更好的信息链接，从而引导用户更有效地使用网络。对网站的拓扑结构挖掘，可以使用关联规则来发现网页之间的关联关系。在它们之间加上超链接使用户更容易地访问他们所需的信息。然而挖掘频繁集的算法时间复杂度仍然很大。特别是发现长频繁集在最坏情况下可达到指数级[2,3,4]，使问题难以解决。由于网站超链接结构的特点，即：一条超链接只能建立在两个网页上。因此，发现频繁集只需找出所 2 – 项集即可，从而降低了 Apriori 的时间复杂度。

本文其它部分的内容安排如下：第 2 章介绍了关联规则的概念及网站拓扑结构。第 3 章介绍了 Apriori 算法。第 4 章依据网站超链接的特性对 Apriori 进行了改进并分析了其时间复杂度。第 5 章依据实验对改进算法进行了验证。第 6 章对本文进行了总结。

2 相关概念

2.1 关联规则

关联规则[2]是发现交易数据库中不同商品之间的联系，这些规则反映顾客购买行为模式，如购买了某一商品的影响。发现这样的规则可以应用于顾客购物分析、目录设计、商品广告邮寄分析、追加销售、商品货架设计、仓储规划、网络故障分析以及根据购买模式对用户进行分类。

（1）定义 1。关联规则挖掘的数据集记为 D，称之为一般事务数据库，$D=\{t_1,t_2,\cdots,t_k,\cdots,t_n\}$，$t_k=\{i_1,i_2,\cdots i_m,\cdots,i_p\}$，$I=\{i_1,i_2,\cdots,i_m\}$ 是 D 中全体项目组成的集合，I 的任何子集 X 称为中的项目集（Itemset），$|X|=k$ 称集合 X 为 k 项目集（k – Itemset），t_k（$k=1,2,\cdots,n$）称为事务（Transactions），i_m（$m=1,2,\cdots,p$）称为项目（Item）。

（2）定义 2。数据集 D 中包含项目集 X 的事务数称之为项目集 X 的支持数，记为 ς_x。项目集 X 的支持度记为 support（X）：

$$\text{support}(X)=\frac{\varsigma_x}{|D|}$$

其中 |D| 是数据集 D 的事务数，若 support（X）不小于用户指定的最小支持度（minsupport），则称 X 为频繁项目集，简称频集（或大项目集），否则称 X 为非频繁项目集，简称非频集（小项目集）。

（3）定义 3。若 X、Y 为项目集，且 $X\cap Y=\Phi$，蕴涵式 $X\Rightarrow Y$ 称为关联规则，X、Y 分别称为关联规则 $X\Rightarrow Y$ 的前提和结论。项目集 $X\cup Y$ 支持度称之为关联规则 $X\Rightarrow Y$ 的支持度，记作：support（$X\Rightarrow Y$），support（$X\Rightarrow Y$）= support（$X\cup Y$）。

（4）定义 4。$\frac{\text{support}(X\cup Y)}{\text{support}(X)}$ 称关联规则 $X\Rightarrow Y$

的置信度，记作：confidence(X⇒Y)。

支持度用于衡量关联规则在整个数据集中的统计重要性，置信度用于衡量关联规则的可信程度。一般来说，只有支持度和置信度均较高的关联规则才可能是用户感兴趣、有用的关联规则。

(5) 定义 5。若 support(X⇒Y) ≥ minsupport，且 confidence(X⇒Y) ≥ minconfidence，称关联规则 X⇒Y 为强规则，否则称关联规则 X⇒Y 为弱规则。

经过关联规则处理后，我们可将数据间的关系抽象为一个有向图，X 和 Y 为图的顶点。X⇒Y 的关系看作为图的边。

2.2 网站结构

网页是信息的载体，网站是链接组织在一起的网页的集合。通过点击链接，用户可以在网页间移动。网站结构是以网状层次结构组织在一起的，如图 1 所示。网站的超链接按照作用可分为基本链接和附加链接两种[4]。基本链接是方向由上层指向下层并且沿着内容分类结构的链接；附加链接是其余为方便用户者需要而设置的链接。

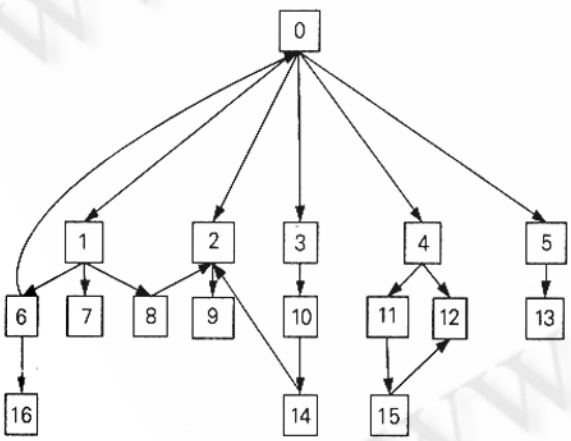

图 1　网页组织结构图

若从用户访问日志中发现网页之间的关联关系，并以此在网页之间添加超链接，就可以使用户更容易地访问自己所需信息，从而提高用户的满意度，增大客户量。例如由上图所示，1 与 14 在网络结构中距离较远的网页，用户访问要经过许多中间页面。如果能够发现 1 与 14 有关联关系，就可以在它们之间添加超链接，这样就方便了用户的访问，进而就改善了用户访问质量。

3 网页结构的关联规则挖掘算法

3.1 Apriori 算法

Apriori 算法[3]是一种最有影响的挖掘布尔关联规则频繁项集的算法。该算法使用频繁项集性质的先验知识，正如我们将看到的。Apriori 使用一种称作逐层搜索的迭代方法，k－项集。首先，找出频繁 1－项集的集合。该集合记作 L_1。L_1 用于找频繁 2－项集的集合 L_2，而 L_2 用于找 L_3，依此类推直到不能找到频繁 k 项集。每找出一个 L_k 需要对数据库进行逐一扫描。

为了提高频繁项集逐层产生的效率，Apriori 用到了一个基本组合性质，这就是：一个大项目集的任何子集一定是大的。因此，组合具有 k－1 个数据项的回答数据项目集，再删除那些含有任何小子集的数据项目集，从而生成具有 k 个数据项的候选项目集。所产生的候选数据项目要小得多，提高了算法的效率。

Apriori 算法如下：

L_1 = {large 1－itemsets};

for ($k = 2$; $L_{k-1} \neq \Phi$; $k++$) do begin

　C_k = apriori－gen(L_{k-1});

　forall transaction $t \in D$ do begin

　　C_t = subset(C_k, t)

　　for candidates $c \in C_t$ do

　　c.count ++;

　end

　$L_k = \{ c \in C_k | c.count \geq minsupp \}$

end

Answer = $\bigcup_{k=1}^{m} L_k$

3.2 Apriori 候选集的产生

Apriori 选候集产生函数 apriori－gen 的参数为 L_k，即所有大型(k－1)项目的集合。它返回所有大型 k 项目集的集合的一个超集(Superset)。首先，在连接(Join)步骤：为找 L_k，L_{k-1}通过与自己连接产生候选 k－项集的集合。该候选项集的集合记作 C_k。设 l_1 和 l_2 是 L_{k-1}中的项集，我们把 L_{k-1}和 L_{k-1}相连以获得候选的最终集合的一个超集 C_k：

(1) Insert into C_k(2) Select $p_{[1]}$, $p_{[2]} \cdots p_{[k-1]}$, $q_{[k-1]}$

(3) from $L_{k-1}p$, $L_{k-1}q$

(4) where $p_{[1]}=q_{[1]},\cdots,p_{[k-2]}=q_{[k-2]},p_{[k-1]}<q_{[k-1]}$

接着,在修剪(Prune)步骤,我们将删除所有的项目集 $c\in C_k$,如果 c 的一些(k-1)子集不在 L_{k-1} 中,为了说明这个产生过程为什么能保持完全性,要注意对于 L_k 中的任何有最小支持度的项目集,任何大小为 k-1 的子集也必须有最小支持度。因此,如果我们用所有可能的项目扩充 L_{k-1} 中的每个项目集,然后删除所有(k-1)子集不在 L_{k-1} 中的项目集,那么我们就能得到 L_k 中项目集的一个超集。

上面的合并运算相当于用数据库中的所有项目来扩展 L_{k-1},如果删除扩展项目集的 k-1 个项目后得到的 k-1 项目集不在 L_{k-1} 中,则删除该扩展项目集。条件 p[k-1]<q[k-1]保证不会出现相同的扩展项。因此,经过合并运算,$C_k\supseteq L_K$。相似的原因,在删除运算中,我们删除 C_k 中其 k-1 子项目集不在 L_{k-1} 中的项目集,同样没有删除包含在 Lk 中的项目集。

(1) for itemsets $c\in Ck$ do
(2) forall (k-1)-subsets s of c do
(3) If ($s\notin L_{k-1}$) then
(4) Delete c from Ck;

虽然 Apriori 能够有效地频繁集,然而当要挖掘的模式数量多且长度很长时,其时间复杂度代价很大[5]。其在最坏的情况下的时间复杂度呈指级增长。

4 Apriori 算法在站点优化中的改进

根据网站结构的特点,基于这样一种假设:网页之间的超链接存在于两个网页之间的。所以对于发现网页之间的频繁集而言,只需要发现频繁 2-项集即可,而不必发现大于 2-项集的频繁集。在最坏情况下,其时间复杂为 $O(n(n-1))=n^2$。这样,关联规则算法不仅可以满足任务的需要,而且可以大大缩短算法的时间复杂度。下面是基于 Apriori 算法的改进算法:

$L_1=\{$large 1-itemsets$\}$;
begin
　$C_2=$ apriori-gen(L1);
　forall transaction $t\in D$ do begin
　　Ct = subset(C_2, t)
　　for candidates $c\in Ct$ do
　　c.count ++;
　end
　$L_2=\{c\in C_k \mid c.count\geq minsupp\}$
end
Answer = ∪L2

5 实验与性能分析

实验分析的环境为 Dell workstation 370 series。以 Windows XP + Mysql + Java 为软件平台,用某网站日志数据为样本。实验的主要分析了 Apriori 改进算法的时间复杂度。设支持度为 0.2%,最大置信度为 5%。本实验的主要工作是对算法的时间复杂度进行分析。

在算法的性能方面,实验主要从页面数与时间来考虑的,如图 2 所示。实验比较了关联规则算法在时间复杂度为 $O(n(n-1))=n^2$ 及 Apriori 改进算法随着页面数的增加时的时间开销。由图可知,随着页面数的增加 Apriori 改进算法比关联规则算法在时间复杂度为 $O(n(n-1))=n^2$ 要缓慢得多,时间变化比较平缓。

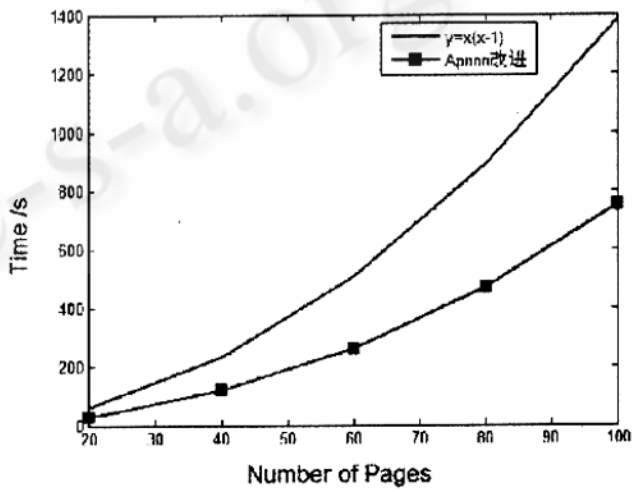

图 2　页面数-时间

6 结束语

网站结构的合理设计影响到网页的访问。本文在分析网站超链接结构的关联规则时,发现超链接是

(下转第 50 页)

(上接第 46 页)

建立在两个网页之间的,提出发现频繁集只需发现 2-频繁集即可。在 Apriori 算法的基础上对其改进,减小其运算时间复杂度。实验结果表明,改进算法的时间复杂度在最坏情况下为 $O(n(n-1))=n^2$,通常情况下远低于 $O(n(n-1))=n^2$。从而可有效发现网页间的关联关系。

参考文献

1 William Ford, Wilam Topp. Data structure with C++[M]. Prentice-hall International, Inc., 1996. 7. 481-485.

2 R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93, 207-216, Washington, D. C.

3 R. Agrawal and R. Srikant. Fast algorithms for mining association rules. VLDB'94 487-499, Santiago, Chile.

4 Nakayama T., H. Kato and Y. Yamane. Discovering the gap between Web site designers' expectations and users' behavior, Artificial Intelligence, 2000, 118. 245-275.

5 J. Han, J. Pei and Y. Yin, (2000). "Mining Frequent Patterns without Candidate Generation", Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 1-12, 2000.