数据挖掘技术及其在旅游线路规划系统的应用

吴春阳[1],何友全[2]
(1.重庆交通大学 管理学院,重庆 400074;
2.重庆交通大学 计算机与信息学院,重庆 400074)

摘 要:研究了旅游线路规划的现状,介绍了在旅游线路规划中使用的方法,引入了关联规则挖掘的基本概念,以及分析了其主要过程。并通过分析关联规则挖掘中的Apriori算法及其改进算法的基础上,提出了一种将Apriori改进算法与旅游线路规划挖掘结合的概念,通过与Apriori算法相比较,提高了系统的效率,并给出了一种典型应用,获得了较理想的应用效果。最后结合当前的旅游网站特点,充分应用网站的信息,设计了一个旅游线路规划的挖掘系统。

关键词:数据挖掘;关联规则;Apriori算法;频繁项集
中图分类号:TP18 **文献标识码**:A **文章编号**:1673-629X(2008)09-0235-04

Application of Association Rule in Data Mining for Tour Planning

WU Chun-yang[1], HE You-quan[2]
(1. Management Department, Chongqing Jiaotong University, Chongqing 400074, China;
2. Computer & Information Engineering Department, Chongqing Jiaotong University, Chongqing 400074, China)

Abstract: Analyses the current situation in route planning, the methods which were used in route planning and the concepts and main process about association rules are introduced, and analyze the process of association rules. And then by analyzing Apriori algorithm and its improvement, a concept was advanced which was based on a new improvement of Apriori algorithm and tour route planning mining, compared with Apriori algorithm, increased the efficiency of system, it gives out an application in tour route planning mining and acquires good effects. Finally according to the actuality of tour website, to make full use of the information on the website, proposes a route planning mining system which based on association rules.

Key words: data mining; association rules; Apriori algorithm; frequent itemset

0 引 言

随着我国经济的发展,旅游业竞争也随之越来越激烈,如何提高企业的竞争力和经营业绩、减少旅游社安排路线的盲目性和随意性,为客户提供合适的旅游线路,从而提高顾客的认可度,已逐渐被相关企业提到日程上来。当前的一些旅游公司在规划旅游线路时,一般采用以下几种方式:一是主题旅游线路设计,比如红色旅游等来安排旅游线路;二是超市型旅游路线设计,顾客可以根据自己需求随意挑选景点,来安排自己的旅游线路;三是应用运筹学方法来寻求最优线路,从而把整个路线关联起来[1];最后就是市场导向,根据市场的要求来安排路线。但这些方法并没有充分利用信息技术的优势。随着数据库技术的发展,尤其是数据挖掘在各个行业中的广泛应用,为挖掘出合适的旅游线路提供了可能。相关公司可以利用数据挖掘技术提出更为合理、受用户欢迎的旅游线路,从而提高企业的经济效益。因此,应用R. Agrawal等人提出的关联规则挖掘技术[2,3]帮助企业分析和处理数据,为在旅游线路规划中提供科学、有效的决策越来越受到相关行业的关注。

1 关联规则描述及相关算法

关联规则挖掘就是从给定的数据集中发现数据项之间所存在的有价值联系。最经典的一个应用技术购物篮分析,通过发现交易数据库中不同商品之间的联系,找出顾客该买的行为模式。根据R、Agrawal等人提出的算法思想[4],关联规则挖掘可以分为两个阶段:

1)找出满足最小支持度的所有频繁项集。

收稿日期:2007-12-29
基金项目:重庆市自然科学基金项目(CSTC, 2007BB2439);重庆市教委基金项目(0634167)
作者简介:吴春阳(1984-),男,河南人,硕士研究生,研究方向为数据挖掘、电子商务;何友全,博士,教授,主要研究方向为智能控制、数据挖掘。

2)从找出的频繁项集中提取满足最小信任度的规则。

定义 1 关联规则:设 $I=\{i_1,i_2,\cdots,i_m\}$ 为数据项集合;D 为相关的数据集合;T 为交易项子集,即 $T<I$,每个交易有一个唯一的交易识别号 TID,$D=\{T_1,T_2,\cdots,T_n\}$ 是交易库,$X\subseteq I$,$|X|=k$ 是 I 上的一个 k 项目集,$X\rightarrow Y$,$X\subseteq I$(满足最小支持度和最小信任度)是一个关联规则[5]。

定义 2 旅游项目:数据库中的一个属性字段。例如用户在访问旅游站点时点击和关注了“北京”和“重庆”,则“北京”和“重庆”就代表了两个不同的项目。

定义 3 旅游交易:用户在一次访问时,发生的所有项目的集合。比如{“北京”,“重庆”,“上海”}。

定义 4 旅游项目集:多个旅游项目的集合。项目集可能是一个交易,也可能不是,但一个旅游交易一定是一个项目集。

定义 5 支持度:项目集 X 在数据集合 D 上的支持度为 $\text{support}(X=>Y)=P(X\cup Y)$。由用户自己定义的一个用来衡量支持度的支持度阈值叫做最小支持度(min_sup)。

定义 6 可信度:$X\rightarrow Y$ 的可信度为 $\text{confidence}(X=>Y)=P(X|Y)$。由用户自己定义的用来衡量可信度的一个可信度阈值(min_conf)。

定义 7 旅游频繁项集(Frequent Itemset):对一个旅游项集 X,如果 X 的支持度不小于最小支持度,即 $\text{support}(X)>\text{min\_sup}$,则 X 即为旅游频繁项集。

1.1 Apriori 算法介绍

Apriori 算法是挖掘产生关联规则所需频繁项集的基本算法。其根据有关频繁项集特性的先验知识,利用层次顺序搜索的循环方法来完成频繁项集的挖掘。算法分为两步,包括先找出所有的频繁项集,然后找出满足最小支持度和最小信任度的关联规则。算法过程[6]如下:

1) $L_1=$ find_frequent_1_itemsets(D);
2) for ($k=2$; $L_{k-1}\neq\emptyset$; k++) {
3) $C_k=$ aproiri_gen(L_{k-1},min_sup);
4) for each transaction t∈D{ //scan D for count
5) $C_t=$ subset(C_k,t); //get subsets of t that are candidates
6) for each candidate c∈C_t
7) c.count++;
8) }
9) $L_k=\{c\in C_k|\ c.\text{count}\geqslant\text{min\_sup}\}$
10) }
11) return $L=\bigcup_k L_k$;

procedure apriori_gen(L_{k-1}: frequent (k-1)-itemset; min_sup: support)
1) for each itemset $l_1\in L_{k-1}$
2) for each itemset $l_2\in L_{k-1}$
3) if ($l_1[1]=l_2[1]$)∧…∧($l_1[k-2]=l_2[k-2]$)∧($l_1[k-1]<l_2[k-2]$) then {
4) c = l_1 l_2; //join step: generate candidates
5) if has_infrequent_subset(c,L_{k-1}) then
6) delete c; // prune step: remove unfrequent cadidate
7) else add c to C_k;
8) }
9) return C_k;

procedure has_infrequent_subset(c: candidate k-itemset; L_{k-1}:frequent (k-1)-itemset)
// use priori knowledge
1) for each (k-1)-subset s of c
2) if c∉L_{k-1} then
3) return TRUE;
4) return FALSE;

但 Apriori 算法存在着两点不足:一是对数据库的扫描次数过多,在内存容量有限,事务数据过多时,系统 I/O 负载较大,会造成每次扫描数据库时间较长,影响系统的效率;二是算法会产生大量的候选项集。k-项集的数量将随着项目的数量的增长将成几何级数增长。

1.2 旅游线路挖掘算法设计

现有的一些 Apriori 改进算法[6,7],基本上都是从优化 Apriori 算法的一项来设计的,并没有考虑到旅游线路挖掘的具体特点。

文中基于文献[8]提出的 Apriori 改进算法的基础上,结合旅游线路挖掘的特点,将之应用到旅游线路规划系统中。Apriori 算法虽然利用了所有非空子集任一子集也应是频繁项集这一性质对候选项集进行了大幅的压缩,但仍需要扫描整个数据库。根据 Apriori 算法的性质,利用 L_{k-1} 来获得 L_k 主要包含两个步骤,即连接和删除。在连接步骤中,为发现 L_k,可以将 L_{k-1} 中的两个项集连接以得到一个 L_k 的候选项集 C_k。C_k 是 L_k 的一个超集,它其中的各个元素并非都是频繁项集,但所有的频繁 k-项集都必定在 C_k 中,即有 $L_k\subseteq C_k$。扫描一遍即可以决定 C_k 中各候选项集的支持频度,并获得 L_k 的频繁 k-项集。由于 C_k 中的候选项集较多,计算量会很大,为减少 C_k 的大小,利用 Apriori 算法性质“一个非频繁项集($k-1$)-项集不可能成为频繁 k-项集的一个子集”,可以将一个候选 k-项集中

任一子集不属于 L_{k-1} 的项集从 C_k 中删去。

当扫描事务数据库,对候选项集的集合 C_k 中的候选 k - 项集计数时,同时产生每个事务的所有($k+1$) - 项集,利用 Hash 表技术可以帮助有效减少候选 k - 项 $C_k(k>1)$ 所占用的空间。在连接阶段所产生的候选($k+1$) - 项集,若其Hash表计数对应的支持度低于最小支持度,则相应的项集为非频繁项集而被移出候选项集。利用 Hash 表技术可以帮助有效减少需要检查的候选 k - 项集数目,尤其当 $k=2$ 时。Hash 函数的构造关系式为:

$$\mathrm{Hash}(x_1,x_2,x_3,\cdots,x_n)=\sum_{i=1}^{k}(\mathrm{order}(x_i)(2^r)^{k-i})\mathrm{mod}(\mathrm{prime}(C_n^k\overline{ED}))=(\mathrm{order}(x_1)(2^r)^{k-1}+\mathrm{order}(x_2)(2^r)^{k-2}+\cdots+\mathrm{order}(x_k)(2^r)^0)\mathrm{mod}(\mathrm{prime}(C_n^k\overline{ED}))$$

在式中,函数 order(x_i) 返回项 x_i 在 C_1 表中的编号,$r(r\in N)$ 是基规模度,n 是候选 1 - 项集的个数 | C_1 |,C_n^k 是没有支持度限制的候选 k - 项集的组合数,$\overline{E}(0<\overline{E}\leqslant 1)$ 是事务项组合存在度,$\overline{D}(0<\overline{D}\leqslant 1)$ 是事务项组合稠密均度,函数 prime(x) 返回不大于 x 的最大素数。

1.3　算法比较分析

在旅游管理系统中,由于数据库 D 中存在大量的数据,故在这里只给出部分数据如表 1、表 2 所示。

表 1　旅游目的地

项目编号	项目
I_1	北京
I_2	上海
I_3	大连
I_4	桂林
I_5	重庆

表 2　旅游网站用户兴趣地调查

TID	用户希望旅游的地区
001	I_1,I_3,I_4
002	I_1,I_2,I_4
003	I_2,I_3,I_5
004	I_1,I_2,I_4,I_5
005	I_1,I_2,I_3
006	I_2,I_5
007	I_1,I_2,I_3,I_5
008	I_2,I_4,I_5
009	I_2,I_4

利用 Hash 函数 $h(x_1,x_2)=(\mathrm{order}(x_1)*10+\mathrm{order}(x_2))\mathrm{mod}\,7$ 构造 Hash 表,如表 3 所示。

表 3　Hash 表

项目	TID							
I_1	001	002	004	005	007			
I_2	002	003	004	005	006	007	008	009
I_3	001	003	005	007				
I_4	001	002	004	008	009			
I_5	003	004	006	007	008			

由表 3 中的数据产生频繁项:

(1) 计算 1 - 项集。表中每行元素个数即为 1 - 项集的支持度。如表 4 所示。

表 4　1 - 项集支持度表

项集	支持度
I_1	5
I_2	8
I_3	4
I_4	5
I_5	5

(2) 发现频繁 2 - 项集。对于 2 - 项集$\{I_i,I_j\}$,只需扫描第 i 项 Hash 表和第 j 项 Hash 表中相同元素的个数,即为二项集$\{I_i,I_j\}$ 的支持度。如表 5 所示。

表 5　2 - 项集支持度表

项集	支持度
$\{I_1,I_2\}$	3
$\{I_1,I_3\}$	3
$\{I_1,I_4\}$	3
$\{I_1,I_5\}$	2
$\{I_2,I_3\}$	3
$\{I_2,I_4\}$	4
$\{I_2,I_5\}$	5
$\{I_3,I_4\}$	1
$\{I_3,I_5\}$	2
$\{I_3,I_5\}$	2

(3) 发现频繁 3 - 项集。若要计算 3 - 项集$\{I_i,I_j,I_k\}$ 的支持度。需要扫描相关项 Hash 表中相同元素的个数。结果如表 6 所示。

表 6　3 - 项集支持度表

项集	支持度
$\{I_1,I_2,I_3\}$	1
$\{I_1,I_2,I_4\}$	1
$\{I_1,I_2,I_5\}$	1
$\{I_2,I_3,I_4\}$	0
$\{I_2,I_3,I_5\}$	2
$\{I_3,I_4,I_5\}$	0

(4) 由上可知,如要计算 k - 项集$\{I_{i1},I_{i2},\cdots,I_{ik}\}$ 的支持计数,只需扫描 $I_{i1},I_{i2},\cdots,I_{ik}$ 项 Hash 表中相同元素个数即可。

假设最小支持度为 2 的情况下，则 3 - 项集{I_2, I_3, I_5}为所需频繁项集，后面的 4 - 项集、5 - 项集也就不用再计算。由此可见，改进后的算法不需产生大量候选项集，同时也节省了存储空间，在支持计算计数时，只需扫描部分项的 Hash 表。

而使用未改进原 Apriori 算法，在支持度同样为 2 的情况下，需要多次扫描数据库，并产生大量的候选频繁项，极大影响了算法的执行效率。

2 基于旅游线路规划的挖掘系统结构设计

根据上述算法，设计了一个基于关联规则的旅游线路挖掘系统(Association Rules based on Tour Route Mining System, ARTRMS)。该系统的设计基于 B/S 结构，通过用户的访问数据来发现和返回结果。如图 1 所示。

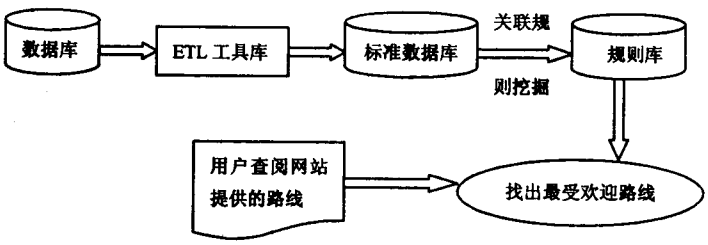

图 1 系统体系结构图

ARTRMS 作为旅游管理系统的一个子系统，其具有两个功能模块，分别为规则挖掘模块和推荐模块。

(1)关联规则挖掘模块是利用 ETL(清理、转换，以及加载)工具从原始数据库中抽取加载数据后生成标准数据，用来进行下一步的挖掘，通过应用改进后的 Apriori 算法，来完成关联规则的挖掘，把挖掘到的关联规则写入到规则库。

(2)推荐模块的作用就是通过 web 直接为访问系统的用户服务。其应记录用户访问的旅游站点，作为模块的输入数据，在网站访问者不知情的情况下，无需用户提供额外的信息，即可为用户提供服务，也使用户不用担心个人信息的泄漏。这一点体现了 ARTRMS 的智能化特点。通过关联规则模块利用其记录的访问记录来进行处理并反馈结果给推荐模块，推荐模块将反馈的规则以一定的形式返回给系统，用户通过访问系统得到推荐的结果。

3 结束语

介绍了关联规则挖掘及其相关算法等，并提出了基于关联规则的旅游线路挖掘算法的应用，重点从旅游线路挖掘出发，通过运用原 Apriori 算法和改进后的算法进行对旅游管理系统中的数据进行挖掘和比较分析后，发现改进后的算法有着比较明显的优势，其可以很好地挖掘出消费者喜欢的景点相关信息，产生了对旅游线路挖掘有指导意义的反馈信息。但由于很多算法都是基于“支持度 - 可信度”框架，这样的结构容易挖掘出错误的规则。为了改进这些算法，人们又引入了兴趣度等概念来修剪无趣的规则。但由于以上都是基于系统方面的讨论，规则可用性的程度由用户自己把握，所以如何将用户的需求和系统结合，应该是以后研究的重点。

参考文献：

[1] 唐力帆. 图论在旅游线路及游览线路设计中的应用[J]. 水运管理，1998(10)：19 - 21.

[2] Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules in Large Database[C]// Proceedings of the 20th International Conference on Very Large Data Bases. Santiago, Chile: [s. n.], 1994: 487 - 499.

[3] Fayyad U, Piatetsky - Shapiro G, Smyth P. From data mining to knowledge discovery: an overview[C]//Fayyad U. In Advances in Knowledge Discovery and Data Mining. Chambridge, Mass: MIT Press, 1996: 1 - 36.

[4] Agrawal R, Imielinski T, Swami A. Mining Association Rules between sets of Items in Large Database[C]//In: Proc of the 1993 ACM SIGMOD Conference. Washington D. C, USA: [s. n.], 1993: 207 - 216.

[5] 佟 强，周园春，阎保平. 关联规则挖掘算法[J]. 微电子学与计算机，2005(6)：68 - 73.

[6] 韩家炜. 数据挖掘概念与技术[M]. 英文第 2 版 . 北京：机械工业出版社，2006：137 - 138.

[7] 马盈仓. 挖掘关联规则中 Apriori 算法的改进[J]. 计算机应用与软件，2004，21(11)：82 - 84.

[8] 陈文庆，许 棠. 关联规则挖掘 Apriori 算法的改进与实现[J]. 微机发展(现名：计算机技术与发展)，2005，15(8)：155 - 177.

(上接第 234 页)

京航空航天大学出版社，2005.

[3] 尚 宇，郅 琦. μC/OS - II 在 LPC2210 上的移植研究[J]. 计算机技术与发展，2007，17(2)：103 - 105.

[4] 樊庆林，吴建国. 提高软件测试效率的方法研究[J]. 计算机技术与发展，2006，16(10)：52 - 54.

[5] 黄燕平. μC/OS ARM 移植要点详解[M]. 北京：北京航空航天大学出版社，2005.

[6] 包枫叶. 嵌入式系统在多端口电缆气压采集器中的应用[J]. 计算机技术与发展，2007，17(3)：222 - 224.