基于改进权重增量 Apriori 算法的产品推荐方法①

王昕妍, 王晓峰

(上海海事大学 信息工程学院, 上海 201306)

摘 要: 采用权重增量挖掘思想优化算法, 为用户推荐个性化产品配置提供了有效的解决方案. 方法主要主要分为 3 个部分, 首先利用平台搭建起来的用户跟踪模块对用户行为进行跟踪和数据的收集; 然后结合用户最近的行为习惯, 使用基于权重增量的Apriori算法进行关联规则挖掘; 最后根据挖掘出的结果完成产品推荐的过程. 通过对挖掘算法的优化, 大大提高了系统的运行效率和准确性, 产品推荐随着用户行为的改变而改变, 更加符合实际情况. 实验结果表明, 该算法可以有效解决产品推荐问题, 相比于传统关联规则挖掘算法, 准确率提高了 4%.

关键词: 权重增量; 产品推荐; 关联规则挖掘

Products Recommendation Based on Improved Weight Increment Apriori Analysis

WANG Xin-Yan, WANG Xiao-Feng

(College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China)

Abstract: In this paper, the weight of the incremental mining thinking optimization algorithm, for users to recommend personalized product configuration provides an effective solution. The method is mainly divided into three parts, first using the platform to build up user tracking module for tracking user behavior and collecting data; then combined with the user's behavior recently, the use of association rules mining based on the weight increment Apriori algorithm; final complete the product according to the recommended procedure to dig out results. By mining algorithm optimization, greatly improving the efficiency and accuracy of the product is recommended with the change in user behavior and changes in the system, more in line with the actual situation. Experimental results show that the algorithm can effectively solve the problem of product recommendation, compared to the traditional association rule mining algorithm, the accuracy is improved by 4%.

Key words: weight of incremental; products recommendation; Apriori algorithm

在互联网技术高速普及, 网络传输和数据压缩、存储等技术快速发展的今天, 人们对互联网的依赖性惊人的增长. 随着电子商务的发展, 互联网上的商品类目和数量不断增加, 加大了用户对自己想买产品的选择, 同时, 电子商务企业也很难了解用户真正感兴趣的内容, 产品推荐功能可以根据用户的兴趣爱好给用户推荐其可能感兴趣的信息, 提高信息的处理效率. 在电子商务中, 很多用户在购买产品后都会发表相对应的评论, 同时, 即使用户最终并没有对产品下单, 执行购买, 其在电子商店中对部分产品的查看和浏览也可以成为其"感兴趣"的依据[1]. 在产品推荐功能中, 用户的浏览、评论、收藏等信息往往隐含了用户对商品的喜好程度以及对商品特定方面的关注程度等潜在信息. 将这些信息提取出来, 为用户的兴趣爱好建立模型, 可以更加精准地为用户推荐合适的商品.

Apriori 算法是 R.Agrawal 和 R.Srikant 与 1994 年提出的为布尔关联规则挖掘频繁项集的原创性算法[2], 是挖掘海量数据关联规则频繁项集的有效算法, 其有效性中体现在 Apriori 性质及其连接、剪枝的操作步骤上[3]. 目前, 许多国内外研究学者都提出了用此方法或

① 收稿时间:2015-03-13;收到修改稿时间:2015-04-29

其改进策略解决类似问题. 例如, 文献[4]提出基于事务压缩对经典的 Apriori 算法进行优化[4]; 文献[5]提出利用有向图对候选集支持度进行优化[5], 从而优化 Apriori 经典算法的剪枝与连接操作; 文献[6]提出利用 Apriori 算法和协同过滤技术相结合, 进行用户评论的文本挖掘, 从而得到基于用户综合偏好和历史评分的产品推荐算法[6].

在上述文献与方法中, 需要大量的数据整合和处理的过程, 在文献[6]的方法中, 作者要进行手动剪枝和对数据简化的过程, 不适合海量数据的处理. 相比之下, 本文旨在通过 Apriori 算法及其改进算法对平台产品的关联性进行进一步挖掘, 使平台更高效精准地处理海量大数据, 并向用户推荐其感兴趣的产品. 关键词库的结合大大提高了信息抽取算法的准确性和通用性, 基于 Web 信息抽取的混合交通出行方案生成与表示系统的成功实验也证明了本文提出的 Web 信息抽取算法的实用性.

1　基于用户群组行为分析平台介绍

本文产品推荐功能平台的流程, 如图 1 所示. 主要是为了给用户提供个性化的产品推荐服务. 用户通过登陆平台, 进行产品的浏览(用户浏览的产品基本是用户感兴趣的类型), 得知产品的参数和价格, 同时了解产品所在类型, 无论最终用户是否购买成功, 用户事务数据库都会记录下产品编号, 产品类型以及搭配等信息. 平台分为以下三个模块来呈现用户行为数据分析流程:

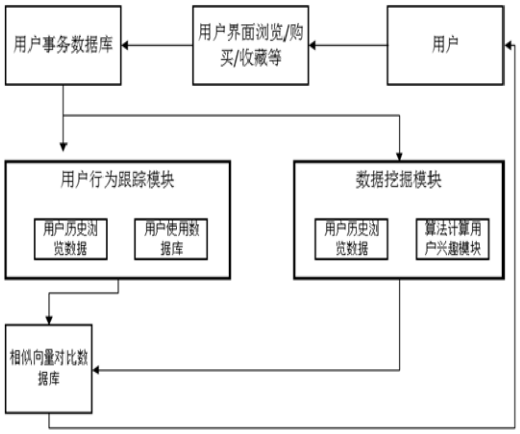

图 1　产品推荐功能平台的流程

1) 用户行为跟踪模块: 通过对用户在网站上的点击, 浏览, 收藏, 存储等行为的跟踪, 将产品数据和浏览产品的客户数据转化为用户购买产品的行为操作数据, 并通过日志数据对用户进行第一次分组;

2) 数据挖掘模块: 将用户日志数据进行传统 Apriori 算法分析和基于权重改进的 Apriori 算法分析并分别取得用户频繁项的关联规则, 这样可以挖掘出用户在最近行为中的规则习惯;

3) 结果展示模块: 通过用户行为追踪模块对用户的分组数据, 以及数据挖掘模块分析出来的产品的关联规则, 基于相似向量比对用户的相似度后, 聚集相似规则用户, 将相似比对结果做 Top-N 推荐.

2　Apriori算法及其改进

2.1 Apriori 算法

Apriori 算法是一种挖掘关联规则的频繁项集算法, 其核心思想是: 连接步和剪枝步. 连接步是自连接, 确保前 K-2 项相同, 并按照字典顺序连接, 剪枝步, 是使任一项频繁项集的所有非空子集也必须是频繁的. 反之, 如果某个候选的非空子集不是频繁的, 那么该候选肯定不是频繁的, 从而可以将其从 CK 中删除.

Apriori 挖掘算法主要步骤如下(简要描述):

依据支持度找出所有频繁项集(频度)

依据置信度产生关联规则(强度)

关于 Apriori 算法的实现步骤, 其使用的是一种被称作“逐层搜索”的迭代方法, “K-1 项集”用于搜索“K 项集”.

首先, 找出频繁“1 项集”的集合, 该集合记做“L1”. L1 用于找频繁“2 项集”L2, 而 L2 用于找 L3, 如此下去, 直到不能找出“K 项集”, 找每个 LK 都需要一次数据库的扫描.

2.2 基于改进权重增量的 Apriori 算法

传统数据库由于要计算全部的用户行为跟踪数据, 因此要获得用户的高频繁文件, 这就一定会造成系统的执行时间以及运行成本的大量增加, 影响整个系统推荐功能的及时性. 同时, 用户最近浏览/购买/收藏/查询的数据不一定会一直围绕相同的类别和风格, 因此本文采用了基于权重的增量式数据挖掘(Incremental Mining Based on Weight, IMW)的思想, 从而更高效的找出用户在最近时间内操作数据的兴趣类别, 增量式挖掘不但可以缩短数据的挖掘时间, 适合大数据量的挖掘, 同时还可以动态挖掘出用户最近的习惯.

Apriori 算法作为挖掘布尔关联规则频繁项集的重要算法, 它是至今为止最有影响力的关联规则算法之一, 其核心思想是基于两阶段频繁项集思想的递推算法. 在文献[7]中, 作者提到了 IMW 的思想, 并使用这种思想优化视频推荐过程[7]. 但在此过程中, 需要不停地比较权重支持度是否超过设定的支持度阈值, 或根据经典传统 Apriori 算法比较最小支持度阈值, 这在整个计算过程中也是非常繁琐的[8]. 本文通过对权重增量思想中的一个参数的设定(即迭代次数阈值 β), 省去了设定权重增量思想的支持度阈值以及 Apriori 算法中的最小支持度阈值, 以达到简化计算提高效率的目的.

本文定义了一个权重支持度(Weight Support, WS)来计算每一类别的交易项目是否是频繁集. 同时, 设定W_j的取值β^{j-1}达到一个阈值时, 或日志交易数据获取完毕后即可停止计算, 并且删除WS_i^j为零的类别文件. (其中, WS_i^j为第 i 个类别文件在第 j 次增量操作数据中的权重支持度; j 为增量挖掘的次数; W_j是权重值大小的计算; β^{j-1}为 β 的 j-1 次方, 为一常数, 其中 β<1).

$$WS_i^j = WS_i^{j-1} + (C_i^j * W_j) ,$$

$$WS_i^0 = 0, W_j = \beta^{j-1}, j = 1,2,\dots\dots,n \quad (1)$$

C_i^j是第 i 个类别文件出现在第 j 次增量交易的出现次数总和.

本文的方法是将权重增量思想加入到 Apriori 算法中并进行优化和改进, 从而求得研究中理想的规则. 挖掘规则的步骤描述如下:

步骤 1: 假设先取操作数据内的最后 n 笔数据, 并计算每一项集类别 i 的次数值.

步骤 2: 以式(1)计算的每一个类别的 WS 值, 判断W_j的取值β^{j-1}的值是否达到一个阈值或者日志交换数据取完.

步骤 3: 如果不符合阈值要求, 或日志交换数据未取完, 就再取下一个 n 笔操作数据, 并再重新计算 WS 值, 直到W_j的取值β^{j-1}达到一个阈值或日志数据取完后, 停止计算.

步骤 4: 删除WS_i^j为零或不足最小支持度 minsup 值的项集项目, 并通过一项集类别的高频繁项目, 组合成二项候选集合, 重复上述方式来计算每一个二项集类别 i.

步骤 5: 二项候选集合依据之前的一项类别集合所做的增量次数(j = 1,2,……,n). 接着删除WS_i^j为零或者不足最小支持度 minsup 值的集合.

步骤 6: 最后剩下的二项集类别将视为用户的最近习惯规则.

最近习惯规则中, WS 的值将大于 minsup 的值.

3 实验结果及分析

3.1 实验平台构建

为验证本文方法的稳定性和有效性, 本文搭建了一个实验平台. 由于实验项目的特殊性和规模, 实验数据来源主要是大致分为三种类型共有大约八十万条数据, 其中浏览数据约为五十万条, 收藏数据约为三十万条, 购买数据约为二十万条(由于每一条数据可能同时进行过浏览、收藏以及购买的操作, 所以数据量会有重复). 大量的测试团队在不同时区对功能进行测试, 生成了极其逼近真实场景的大量测试数据. 部分数据样例如图 2 所示.

id	machine_id	session_id	click_content
1500002	5F8A5506-C893-Z1A7-80B9-B30C2C194557	B784611B-669B-ZCB3-A257-E4946A2A8D38	ProLiant Graphics Server Blades
1500003	5F8A5506-C893-Z1A7-80B9-B30C2C194557	B784611B-669B-ZCB3-A257-E4946A2A8D38	网络
1500017	5F8A5506-C893-Z1A7-80B9-B30C2C194557	B784611B-669B-ZCB3-A257-E4946A2A8D38	交换机
1500020	5F8A5506-C893-Z1A7-80B9-B30C2C194557	B784611B-669B-ZCB3-A257-E4946A2A8D38	路由器
1500021	5F8A5506-C893-Z1A7-80B9-B30C2C194557	B784611B-669B-ZCB3-A257-E4946A2A8D38	HP MSR20 Series
1500025	5F8A5506-C893-Z1A7-80B9-B30C2C194557	B784611B-669B-ZCB3-A257-E4946A2A8D38	HP ProLiant SL6500 可扩展系统
1500026	5F8A5506-C893-Z1A7-80B9-B30C2C194557	B784611B-669B-ZCB3-A257-E4946A2A8D38	HP ProLiant SL230s Gen8 服务器
1500027	5F8A5506-C893-Z1A7-80B9-B30C2C194557	B784611B-669B-ZCB3-A257-E4946A2A8D38	ProLiant Server Blades
1500032	5F8A5506-C893-Z1A7-80B9-B30C2C194557	B784611B-669B-ZCB3-A257-E4946A2A8D38	网络
1500034	5F8A5506-C893-Z1A7-80B9-B30C2C194557	B784611B-669B-ZCB3-A257-E4946A2A8D38	HP 830 统一有线-WLAN 交换机系列
1500004	012EB627-074F-Z744-85C3-091A7F3E789B	621E8B49-7942-ZD56-9995-C532514B8525	交换机
1500007	012EB627-074F-Z744-85C3-091A7F3E789B	621E8B49-7942-ZD56-9995-C532514B8525	路由器
1500009	31E2415B-B8D4-Z75F-A951-1C2DA85FA369	621E8B49-7942-ZD56-9995-C532514B8525	Integrity Server Blades
1500011	31E2415B-B8D4-Z75F-A951-1C2DA85FA369	825C492F-56AE-ZA20-A679-73184A592FFA	HP ProLiant 可扩展系统
1500012	31E2415B-B8D4-Z75F-A951-1C2DA85FA369	825C492F-56AE-ZA20-A679-73184A592FFA	HP ProLiant SL6500 可扩展系统
1500018	31E2415B-B8D4-Z75F-A951-1C2DA85FA369	825C492F-56AE-ZA20-A679-73184A592FFA	HP 830 统一有线-WLAN 交换机系列
1500019	31E2415B-B8D4-Z75F-A951-1C2DA85FA369	825C492F-56AE-ZA20-A679-73184A592FFA	网络
1500024	31E2415B-B8D4-Z75F-A951-1C2DA85FA369	825C492F-56AE-ZA20-A679-73184A592FFA	HP ProLiant 可扩展系统
1500037	31E2415B-B8D4-Z75F-A951-1C2DA85FA369	825C492F-56AE-ZA20-A679-73184A592FFA	路由器
1500039	31E2415B-B8D4-Z75F-A951-1C2DA85FA369	825C492F-56AE-ZA20-A679-73184A592FFA	网络
1500041	5F8A5506-C893-Z1A7-80B9-B30C2C194557	B784611B-669B-ZCB3-A257-E4946A2A8D38	HP 移动安全 IDS/IPS 系统系列

图 2 部分数据样例示意图

上图中可以看到, 整个数据样例表有四类数据(实际数据库中当然不止这四类数据, 只是将跟本文相关的数据显示在这里), 其中 id 是每条数据的唯一标识, 第二列的machine_id则是代表不同的计算机, 即不同的机器对系统进行操作, 会有不同的 machine_id, 第三列的session_id是代表不同的浏览器, 而最后一列则是保存了用户点击过的产品名称. 以上是数据库中跟踪保存的原始数据, 实验通过程序将数据进行整合, 归类,

将 machine_id 和 session_id 分别都相同的数据归并起来，即可以看出一个用户在同一时段都点击了哪些产品(在实际系统数据中，点击并不能代表用户已经进行了最后的购买，只能广泛的表示用户对此产品感兴趣). 整理后的数据样例如图 3 所示.

id	machine_id	session_id	click_content
1500002	5F8A5506-C893-Z1A7-80B9-B30C2C194557	B784611B-669B-ZCB3-A257-E4946A2A8D38	ProLiant Graphics Server Blades
1500003			网络
1500017			交换机
1500020			路由器
1500021			HP MSR20 Series
1500025			HP ProLiant SL6500 可扩展系统
1500026			HP ProLiant SL230s Gen8 服务器
1500027			ProLiant Server Blades
1500032			网络
1500034			HP 830 统一有线-WLAN 交换机系列
1500004	012EB627-074F-Z744-85C3-091A7F3E789B	621E8B49-7942-ZD56-9995-C532514B8525	交换机
1500007			路由器
1500009	31E2415B-B8D4-Z75F-A951-1C2DA85FA369	621E8B49-7942-ZD56-9995-C532514B8525	Integrity Server Blades
1500011	31E2415B-B8D4-Z75F-A951-1C2DA85FA369	825C492F-56AE-ZA20-A679-73184A592FFA	HP ProLiant 可扩展系统
1500012			HP ProLiant SL6500 可扩展系统
1500018			HP 830 统一有线-WLAN 交换机系列
1500019			网络
1500024			HP ProLiant 可扩展系统
1500037			路由器
1500039			网络

图 3　整理后的样例数据表

由图 3 可以看出，同一框图中的产品即为同一个用户在某一时间段所点击和搭配过的产品，根据整理后的数据，我们就可以对这些产品进行关联规则挖掘，从而找出产品之间的关联性，给出产品推荐的结果.

本文构造了三种方法，分别对未加入产品推荐功能，加入 Apriori 算法优化的产品推荐功能以及加入基于改进权重增量的 Apriori 算法的产品推荐功能的系统进行高效性和稳定性的检测，其目的在于比较和检测经典 Apriori 算法和优化后的算法解决这一问题的效率差距，同时验证使用算法优化后，产品推荐功能的精准性和高效性.

本实验采用准确性和高效性这两个指标来衡量实验方法的有效性，为了准确计算实验的准确率和召回率，评价指标公式如公式(2)所示[9]. 其中 TP 表示应用产品推荐方法推荐的且用户实际需要的产品数量，FP 表示应用产品推荐方法推荐的但用户实际并不需要或感觉并不合适的产品数量，FN 表示产品推荐方法没有推荐但用户实际需要的产品数量，TN 表示产品推荐方法没有推荐用户也确实不需要的产品数量.

$$\text{准确率(Pr)}=\frac{TP}{TP+FP}$$

$$\text{召回率(Re)}=\frac{TP}{TP+FN} \quad (2)$$

3.2　实验结果分析

通过对大量测试数据进行采集和分析，采用构建的三种方法分别进行产品推荐，得到如图 4 和图 5 所示的实验数据结果图.

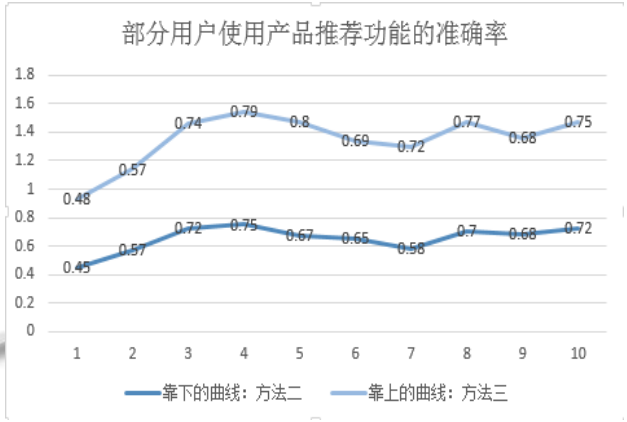

图 4　部分用户使用功能准确率图表

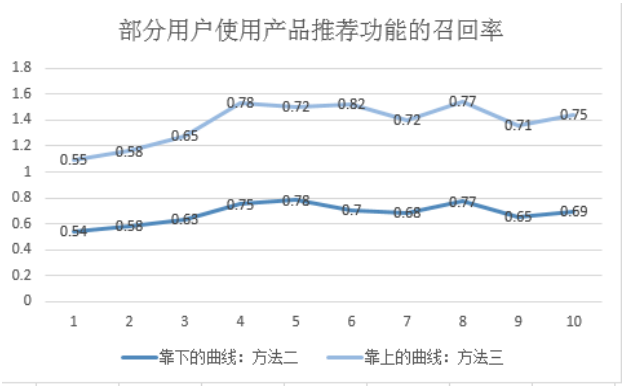

图 5　部分用户使用功能召回率图表

图 4 显示了部分用户的准确率分布，图表中靠下的曲线是在使用方法二的系统中，用户使用产品推荐功能对产品进行配置的准确性，可以看出，在用户数量不多的情况下，方法二和方法三的准确率相差不大，而对于用户渐渐增多，方法三显示出其高效精准的优点.

图 5 显示了部分用户的召回率分布，同样由图中可以看出，用户数量比较低的情况下，无论准确率还是召回率都并没有显示出其算法的优势，而用户数量增长的过程中，才能逐渐看出算法优化后的系统性能的提升. 这也充分说明大数据分析后结果更加精准和有效.

最后，本文将大量的准确率和召回率数据分别进行了简单的平均值计算，以及通过计算三种方法在运行过程中的时间复杂度，得到如表 1 所示的实验结果图(方法一：未加入产品推荐功能时预测用户配置产品

的性能; 方法二: 加入经典 Apriori 算法优化的产品推荐功能后预测用户配置产品的性能; 方法三: 加入基于改进权重增量 Apriori 算法的产品推荐功能后预测用户配置产品的性能).

表 1 三种实验方法的评价指标对比

	方法一	方法二	方法三
准确率	0.14	0.68	0.72
召回率	0.19	0.70	0.76
时间复杂度	无	O(n)	O(lgn)

由于方法一中, 整个系统并没有展示出产品推荐的功能, 而仅仅是将产品的使用和购买情况做了数据分析, 并将分析结果以柱状图或饼状图的形式展示给用户, 因此在方法一中, 需要用户自行通过这些分析数据来判断适合自己配置的产品, 从而无法给出时间复杂度.

可以看到, 对于方法二, 本文使用传统 Apriori 算法优化整个系统, 推出的产品推荐功能, 其准确率和召回率已经大大增长, 只是对于系统所用的数据平台 Storm 来说, 传统的 Apriori 算法需要兼容其分段式的数据处理方式, 以及海量的扫描次数, 所需的时间和空间有很大程度的增加, 对于系统的运行成本也产生了巨大的压力, 因此匹配算法的最小时间复杂度是 O(n). 对于方法三, 采用了基于改进权重增量的 Apriori 算法进行优化, 其本质是一种递归算法, 算法主要对于用户的近期操作行为规则进行增量式更新, 同时采用折半式查找, 因此匹配算法的时间复杂度最大为 0(lgn). 从代价上考虑, 两种时间复杂度都是可以接受的, 但系统追求更高的效率, 并且从表 2 可以看出, 方法三的准确率和召回率相比方法二也有了一定程度的提高. 于是从所有指标上综合考虑, 该算法对于提升产品推荐效果确实起到了重要的作用.

通过以上数据分析可以看到, 方法二利用传统增量挖掘算法得到的结果已经能够非常好的挖掘用户的喜好, 相比之前能够更准确和高效的向用户推荐产品. 而方法三利用优化后的增量挖掘算法, 在稳定性和准确性方面都较方法二有所提高, 更重要的是, 在时间成本上大大压缩了系统的运行时间, 能更快提供用户所需要的服务, 是一种有效的手段.

4 结语

本文首先通过对用户行为的跟踪, 对用户的操作数据进行记录, 结合用户的行为习惯和兴趣爱好, 用传统 Apriori 算法和基于改进权重增量的 Apriori 算法来挖掘关联式规则, 最后通过数据分析结果给用户展现出来进行产品推荐, 完成了整个个性化推荐的过程. 本文通过实验结果证实了此推荐方法的有效性和稳定性, 利用权重增量挖掘的效果比一般挖掘效果要好, 同时也更高效快速. 本实验的准确率高达 72%, 召回率高达 76%, 比其他推荐方法更为精准. 综合上述实验方法, 可以证明本文使用的算法优化的产品推荐方法是一种行之有效的产品推荐策略, 基本达到了预期的效果.

本文的主要贡献在于, 在极其逼真的情况下以及大量数据的状态下对想法进行试验, 利用相关规则挖掘的方法分析用户的喜好和行为习惯, 找到相同喜好的用户从而进行产品推荐. 在产品推荐的实际应用中, 推荐的及时性往往比准确性更为重要[10], 因此系统运行的时间成本也是本文考虑的重要因素之一. 在今后的研究中, 作者将继续深入探索基于行为分析的产品推荐方法, 以及数据分析的优化方案, 使得系统能更好达到用户的需求.

参考文献

1 SKrishnapp, DK, Zink M, Griwodz C. Cache-centric vide recommendation: 0020an approach to improve the efficiency. if YouTube caches. Proc. of the 4th ACM Multimedia System Conference. Oslo. 2013. 261–270.

2 Tan PN, Steinbach M, Kumar V. 范明,范宏建,译.数据挖掘导论.北京:人民邮电出版社, 2006.

3 陈安,陈宁,周龙骧,等.数据挖掘技术及其应用.北京:科学出版社,2006.

4 颜雪松,蔡之华.一种基于 Apriori 的高效关联规则挖掘算法的研究.计算机工程与应用,2002:209–211.

5 白似雪,朱涛,梅君.基于图的 Apriori 算法改进.南昌大学学报(工科版),2009,31(1).

6 扈中凯,郑小林,吴亚峰,陈德人.基于用户评论挖掘的产品推荐算法.浙江大学学报(工学版), 2013,8.

7 Awadalla MH, Elfar SG. Aggregate function based enhanced apriori algorithm for mining association rules. International Journal of Computer Science Issues, 2012, 9(3): 277–287.

8 胡吉明,鲜学丰.挖掘关联规则中的 Apriori 算法的研究与改进.计算机技术与发展,2006,16(4): 99–101.

9 牛丽敏.Apriori 算法分析与改进综述.桂林电子科技大学学报,2007,27(1):27–30.

10 徐章艳,刘美玲,张世超.Apriori 算法的三种优化方法.计算机工程与应用,2004,40(36):190–193.