

doi:10.3772/j.issn.1000-0135.2010.04.018

一种基于加权关联规则的协同推荐算法<sup>1)</sup>

哈进兵 郑锐 甘利人

(南京理工大学经济管理学院, 南京 210094)

**摘要** 协同过滤技术不需要分析待推荐资源的内容信息,在电影、音乐、图书等非结构化数据占主流的电子商務推荐领域得到了广泛的应用,成为电子商务推荐领域的主流技术。针对基于项目的协同过滤算法不能实现“跨类型”推荐的缺点,本文提出了一种新的基于关联性评分预测的协同过滤算法 IAPCF。区别于传统的算法, IAPCF 算法根据项目之间的关联规则,而不是根据多用户对项目评分形成的向量间的相似度来寻找项目的最近邻居集合。该算法能较好地实现“跨类型”项目的推荐。实验结果表明, IAPCF 算法具有更好的推荐精度。

**关键词** 推荐系统 协同过滤 关联规则 相似度

## Research on a Novel Collaborative Filtering Algorithm

Ha Jinbing, Zheng Rui and Gan Liren

(Nanjing University of Science &amp; Technology, Nanjing 210094)

**Abstract** Collaborative filtering technology does not need to analyse the content information of the recommended resources, and has been widely used and become the mainstream technology in e-commerce recommendation field. An item-association-prediction-based collaborative filtering algorithm (IAPCF) is proposed to overcome the shortcomings of the traditional item-based collaborative filtering algorithms. Different from the traditional algorithms, IAPCF algorithm does not need to calculate the similarities between items. It uses the association rules between items to predict the user preference of user for the items. The IAPCF algorithm performs well in the cross-type recommendation. The experiment results suggested that IAPCF could provide better recommendation results.

**Keywords** recommendation system, collaborative filtering, association rule, similarity

## 1 引言

随着互联网的普及和电子商务的广泛应用,人们在享受网上购物便捷性的同时也陷入信息过载的困境,用户在大量的产品信息中难以找到自己需要

的商品。因此,电子商务推荐系统应运而生。基于用户模型的协同推荐系统从数据中抽取关系描述模型,推荐速度通常较慢;实时推荐用离线方法完成复杂的预计算,运算速度较快。不足之处在于与基于全体存储纪录的方法相比,算法会牺牲一定的精确度。

收稿日期: 2009年5月20日

作者简介: 哈进兵,女,1975年生,2003年南京航空航天大学控制科学与工程专业博士后出站,博士,副教授,主要研究领域:情报学、电子商务、网络信息资源组织与检索。E-mail: hajinbingshuai@yahoo.com.cn。郑锐,男,1985年生,南京理工大学情报学硕士,主要研究方向:电子商务。甘利人,女,1957年生,教授,博士生导师,主要研究领域:网络信息资源管理。

1) 本文系总装备部基础科研项目“基于门户网站的个性化信息服务技术方案及应用研究”(项目编号:2004QB1505)的研究成果之一。

关联规则是数据挖掘技术能发现的非常重要的一类规则,它首先由 Agrawal 等于 1993 年提出<sup>[1]</sup>,能够揭示项集之间的联系。Rakesh Agrawal 和 Ramakrishnan Skrikant 提出的 Apriori 算法<sup>[2]</sup>是最经典的关联规则挖掘算法。这是一个基于两阶段频集思想的方法,能发现所有的有充分支持度和置信度的关联规则。算法原理是寻找那些事务的支持度超过最小支持度的项目的所有组合,即频繁项目集,然后用频繁项目集产生需要的规则。

## 2 基于加权关联规则的协同推荐处理

### 2.1 加权关联规则的概念

在数据库知识发现(KKD)研究中,一般的关联规则挖掘都假设数据库中的各项目具有相同的重要性,计算频繁项目集时,只考虑项目出现的频率。但在某些应用领域,用户对不同的项目关注程度不同,即项目的重要性是不同的。为体现项目的重要性,引入了项目加权。对项目加权后,挖掘这种加权的关联规则时,要综合考虑项目集在事务数据库出现的频率和项目的权值。

同时,加权的过不仅可以区分项目的重要程度,使挖掘出的结果变得更加合理,而且还能够大大提高算法的运行效率。因为在关联规则算法中,产生频繁项目集阶段将耗费主要的机器运算时间,如果在频繁集产生的初期将那些权重小的无关项目剪掉,便能有效地降低算法的时间复杂度。

对于上述事务数据库,为了表征项目的重要性,我们为每一个项目  $i_j$  赋以权值  $w_j$ , 其  $0 \leq w_j \leq 1, j = \{1, 2, \dots, n\}$ 。  $I_1 = \{i_1, i_2, \dots, i_k\} \subseteq I$ , 设  $w\text{support}(I_1)$  为项集  $I_1$  的加权支持度,大于最小加权支持度  $m\text{wsupport}$  的项集为加权频繁项集。

### 2.2 协同推荐中加权关联规则的挖掘

本文研究的协同推荐算法分为离线的加权关联规则集生成阶段和在线的推荐应用阶段。加权关联规则集生成阶段比较费时,离线周期进行保证推荐算法的实时性要求;在线阶段根据生成的加权关联规则集和用户的访问行为向用户提供实时的推荐服务。其具体的算法步骤如下:

从算法 1 可以看出,步骤①②③④为离线的加权关联规则集生成阶段,步骤⑤⑥为在线的推荐应用阶段。离线阶段比较费时,是整个基于加权关联规则的协同推荐中需要解决的重点问题。该阶段涉

及的关键问题有五个方面:

#### 算法 1 基于加权关联规则的协同推荐的算法步骤

- ① 根据近期用户下载记录数据库中注册用户下载过的所有资源的数据创建用户下载资源事务记录,构造事务数据库。
- ② 根据资源近期下载量表和文章著录表生成资源权重表。
- ③ 接受用户设置的频繁一项集覆盖率,生成最小支持度  $\text{minSupport}$ 。
- ④ 根据资源权重表 and 用户资源下载事务表采用基于 Apriori 算法的频繁项集发现方式生成加权关联规则,记加权关联规则集合为  $wR$ 。
- ⑤ 对每个当前用户  $u$ ,根据  $wR$  和用户资源推荐集生成算法来生成该用户的推荐集  $R_u$ 。
- ⑥ 将  $R_u$  返回给用户  $u$ 。

#### (1) 事务的划分问题

本文采用了以一个用户一次登陆作为一次事务的划分。因为用户一次登陆过程中,常带着一个主要问题或多个问题向系统提问,这些问题可能不属于一个主题,但反映的是该用户此时的兴趣,该划分原则能发现不同主题之间有趣的联系。

#### (2) 项目权重的设置

本文中所指的项目是文档资源,资源的新颖程度是吸引用户最重要的参数。即选择资源的著录时间作为生成资源项目权重的参数。具体项目权重设置如下:

设资源近期下载量表中资源序列为  $(d_1, d_2, \dots, d_j, \dots, d_n)$ , 其对应的著录时间序列为  $(t_1, t_2, \dots, t_j, \dots, t_n)$ 。设  $T_{\max}$  为著录时间序列中的最新时间,  $T_{\min}$  为最旧时间,则资源  $d_j$  的权重值计算公式为:

$$w_j = \alpha + (1 - \alpha) * (t_j - T_{\min}) / (T_{\max} - T_{\min}) \quad (1)$$

其中,  $\alpha$  为常数,具体的值可以根据经验或者试验来确定。

#### (3) 加权支持度的定义

文献[3]中将项集的加权支持度定义为  $\sum_{j=1}^k w_j(\text{support}(I_1))$ ; 文献[4]中将项集的加权支持度定义为  $\max(w_1, w_2, \dots, w_j)(\text{support}(I_1))$ ; 文献[5]中将项集的加权支持度定义为  $\frac{1}{k}(\sum_{j=1}^k w_j)(\text{support}(I_1))$ 。各文献关于加权支持度的定义各不相同。本文给出如下的定义方法:

设  $I = \{i_1, i_2, \dots, i_m\}$  为资源项目集,其对应的权重集为  $W = \{w_1, w_2, \dots, w_m\}$ ,  $W$  是经过归一化后

的权重集。归一化的过程如下:设  $a = w_1 + w_2 + \dots + w_m$ , 则  $\{w_1/a, w_2/a, \dots, w_m/a\}$  即是归一化后的项目权重集, 归一化后的权重集不妨仍记为  $W = \{w_1, w_2, \dots, w_m\}$ 。

**定义 1.** 定义项集  $I_1 = \{i_1, i_2, \dots, i_k\}$  的加权支持度为:

$$wSupport(I_1) = \sum_{j=1}^k w_j (Support(I_1))$$
, 其中  $Support(I_1)$  为项集  $I_1$  的普通支持度。

**定义 2.** 定义加权关联规则  $I_1 \Rightarrow I_2$  的加权支持度为:

$$wSupport(I_1 \Rightarrow I_2) = \sum_{j=1}^{k+p} w_j (Support(I_1 \cup I_2))$$

**定义 3.** 定义加权关联规则  $I_1 \Rightarrow I_2$  的加权置信度为:

$$wConfidence(I_1 \Rightarrow I_2) = Support(I_1 \cup I_2) / Support(I_1)$$

**定义 4.** 定义加权关联规则  $I_1 \Rightarrow I_2$  的推荐度为:

$$recom(I_1 \Rightarrow I_2) = wSupport(I_1 \Rightarrow I_2) * wConfidence(I_1 \Rightarrow I_2)$$

#### (4) 最小支持度阈值的设定

文献[4]提出一种允许用户设定多个最小支持度、给定数据各项的权重表的解决方案。该方法对用户的专业性要求高, 操作的难度大。本文提出通过控制频繁项集的数量来控制频繁项集的规模, 从而控制规则的数量。以此得出符合用户要求的最佳的最小支持度阈值。上述方法降低了用户设定最小支持度阈值的操作难度, 具有很好的用户友善性。

#### (5) 加权关联规则的挖掘算法

文献[3]提出了基于概率分布的加权关联规则挖掘算法; 文献[4]提出了一种基于 Apriori 算法的挖掘加权频繁项集的算法 MWFS; 文献[6]提出了一种基于多支持度的加权关联规则挖掘算法; 文献[7]、文献[8]给出挖掘布尔型属性关联规则的两个典型算法: 快速算法与划分算法; 文献[9]给出两个加权关联规则的挖掘算法: MINWAL(O) 算法和 MINWAL(W) 算法; 文献[10]提出利用普通关联规则算法发现加权关联规则, 通过完善普通关联规则的挖掘算法来提高挖掘的效率。这种改进算法具有广泛的适应性和可扩展性。PIRS 系统中采用基于 Apriori 算法的频繁项集发现方式和文献[10]的相关思路来生成加权关联规则, 其算法流程为:

#### 算法 2 加权关联规则的挖掘

① 以用户一次登录作为事务的划分, 根据近期下载记录表生成用户资源下载事务表。近期下载记录表的存储结构为 {用户 ID, 登录时间, 用户提问, 提问时间, 下载的文档, 下载时间}, 划分每个事务的标准为登陆时间。

② 根据资源近期下载量表和文章著录表生成资源权重表。

③ 接受用户设置的频繁项集覆盖率, 生成最小支持度 minSupport。

④ 根据资源权重表 and 用户资源下载事务表采用基于 Apriori 算法的频繁项集发现方式生成加权关联规则。

第一步, 扫描数据库, 搜索数据库中事务的最大长度 Size 返回该数值。

第二步, 访问资源权重表, 生成按降序排列的资源权重序列  $W(w_1, w_2, \dots, w_j, \dots, w_n)$ 。

第三步, 根据第一步中的返回结果 Size 与权重集 W 计算  $\Delta = 1 / \sum_{j=1}^{Size} w_j$ 。根据 minSupport 和  $\Delta$  生成最小加权支持度 wMinSupport,  $wMinSupport = minSupport / \Delta$ 。

第四步, 使用 Apriori 算法以 minSupport 为最小支持度生成频繁集, 同时填充项集的加权支持度。

第五步, 采用 wMinSupport 为最小加权支持度筛选加权频繁项集。

第六步, 在加权频繁集的基础上生成加权关联规则。

### 3 基于项目关联性的预测计算

本文从项目之间的关联性进行改进, 以期能更准确地预测用户对项目的评分。经过修正后的计算用户对项目预测评分的公式为:

$$P_{a,p} = \bar{R}_p + \frac{\sum_{n \in MAI} AC(p, n) (R_{a,n} - \bar{R}_n)}{\sum_{n \in MAI} AC(p, n)} \quad (2)$$

相关定义:

①  $AC(p, n)$  为项目  $p$  和项目  $n$  之间关联规则  $p \Rightarrow n$  的置信度。

②  $MAI$  为项目  $p$  的最近关联项目集合。

具体构成规则为: 首先, 提取以项目  $p$  为前项构成的一阶关联规则  $p \Rightarrow n$ ; 其次, 对提取的一阶关联规则按置信度降序排列; 然后, 按照最小置信度阈值设定, 提取大于最小置信度阈值的关联规则的后项组成  $MAI$ ; 或者按照规则的个数, 提取前  $N$  个规则的后项组成  $MAI$ 。

因为式(2)是按照项目之间的关联性来寻找用户可能感兴趣的项目, 所以可能给用户推荐出全新的项目, 实现过去不能实现的“跨类型”推荐。

## 4 IAPCF 算法的实验测试

### 4.1 实验设计

#### (1) 实验数据集处理

用于实验的数据集包含 1 000 000 条评分数据, 由 6040 个用户对 3900 部电影的评分构成。平均每个用户评分的电影数为 166 部, 其中每个用户至少对 20 部电影进行了评分。

#### (2) 实验内容和编程环境

使用随机隐藏用户对一个项目的评分, 得出数据集上的平均绝对偏差 MAE 值, 用来对比 IAPCF 算法和 Item-Based 算法的推荐精度。实验采用 Java/eclipse3.2 编程环境实现这两种算法。实验的数据库系统采用了 MySQL5.0, 服务器为 TomcatApache。本实验在 Windows XP 操作系统、CPU 1.7G、内存 DDR 512M 的硬件环境下实现。

### 4.2 实验结果及分析

按 5 折交叉法预处理数据集, 也即在训练集与测试集比率  $x = 0.8$  的前提下进行五次实验。项目最近邻居个数和项目最近关联项目个数按同样规律变化, 从 10 ~ 100 变化, 测试 IAPCF 算法和 Item-Based 算法的平均绝对偏差 MAE 的变化。实验结果见表 1、图 1。

表 1 算法测试 MAE

关联项目个数 \ 算法	IAPCF 算法	Item-Based 算法
10	0.913	0.917
20	0.865	0.873
30	0.856	0.859
40	0.856	0.859
50	0.859	0.863
60	0.862	0.866
70	0.864	0.868
80	0.865	0.869
90	0.865	0.870
100	0.865	0.871

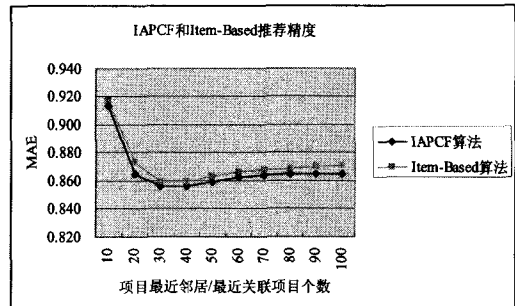


图 1 IAPCF 和 Item-Based 推荐精度比较

均绝对偏差 MAE 的值逐渐减小, 邻居个数大于 30 时, MAE 减少得非常有限, 并且曲线趋于平滑。但是 IAPCF 算法的预测质量比 Item-Based 算法在整个区间内都稍好一些。

(1) 本文提出的 IAPCF 算法虽然在实验数据上进行测试后表现出良好的推荐精度, 但是实验数据和实际应用系统环境数据存在一定的距离。IAPCF 算法解决了“跨类型”推荐的问题, 尤其在大型综合的电子商务推荐系统中, 有比传统的基于项目的算法更大的应用空间。下一步的工作可以是基于 IAPCF 算法开发成型的推荐系统, 通过和现实的电子商务网站结合来验证推荐算法的有效性。

(2) 根据实验所用的数据集, 无法推出邻居个数大于 100 以后的 MAE 的变化情况。但当邻居个数大于 100 以后, Item-Based 算法需要计算项目之间的相似性, 会变得更加复杂; 由于 IAPCF 算法在整个运算过程中都不需要计算项目之间的相似性, 只需要根据项目之间的关联规则来计算电子商务用户对项目的评分, 而其关联规则的生成是离线进行的, 这样就大大减少推荐的时间, 满足实时性的要求。

(3) 在个数为 30 的时候, MAE 变化较大说明本次实验的数据在一个特定用户已经积累了相当数量数据的这种稳定状态下, 当邻居个数为 30 的时候, 达到最好的推荐精度。本文在对算法进行测试时, 设计了测试方案是用于测试算法在数据集相对较稠密的情况的表现, 不同的测试数据集和 X 的选定会对算法的表现有着较大的影响, 如何消除测试方案对算法表现的影响是本文需要进一步研究的课题。

## 5 结 论

本文针对传统的基于项目的协同过滤算法存在的问题, 提出了一个基于项目关联性评分预测的改

从图 1 的折线可以看出: 当项目最近邻居/最近关联项目的个数从 10 增加到 30 时, 两种算法的平

进算法 IAPCF。与传统的基于项目的协同过滤算法相比,IAPCF 算法不需要进行项目之间相似性的计算,只需要寻找项目之间的一阶关联规则集合。文章证明了 IAPCF 算法能有效地解决传统的基于项目的协同过滤算法不能实现“跨类型”推荐的问题。实验表明,IAPCF 算法比传统的基于项目的协同过滤算法表现出更好的推荐精度。

### 参 考 文 献

- [ 1 ] Agrawal R, Imielinski T, Swami A. Mining Associations Between Sets of Items in Large Databases[C]//Proc. of the ACM SIGMOD Int'l Conference on Management of Data, Washington D.C., May, 1993: 207-216.
- [ 2 ] Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules[C]//Proc. 20th Int. Conf. Very Large Data Bases, VLDB, Santiago, Chile, Sept., 1994: 487-499.
- [ 3 ] 赵亮,胡乃静,张守志.个性化推荐算法设计[J].计算机研究与发展,2002(8):986-991.
- [ 4 ] Sarwar B M, Karypis G, Konstan J A, et al. Item-based collaborative filtering recommendation algorithm [C]//Proceeding of the Tenth International World Wide Web Conference, ACM Press, 2001.
- [ 5 ] Herlocker J L, Konstan J A, et al. An Algorithmic Framework for Performing Collaborative Filtering [C]//Proceedings of ACM SIGIR'99, ACM press, 1999.
- [ 6 ] Oyanagi S, Kubota K, Nakase A. Application of Matrix Clustering to Web Log Analysis and Access Prediction[C]//Proceedings of the WebKDD Workshop (WebKDD 2001), pp.13-21, San Francisco, CA, 2005.
- [ 7 ] Shardanand U, Maes P. Social information filtering: algorithms for automating word of mouth [C]//Roberts T, Robertson S. Proceedings of the ACM CHI'95 Conference on Human Factors in Computing Systems. New York: ACM Press, 1995: 210-217.
- [ 8 ] 周军锋,汤显,郭景峰.一种优化的协同过滤推荐算法[J].计算机研究与发展,2004(10):1842-1847.
- [ 9 ] 邓爱林,朱扬勇,等.基于项目评分预测的协同过滤算法[J].软件学报,2003(9):1621-1628.
- [ 10 ] Sarwar B M, Karypis G, Konstan J A, et al. Application of Dimensionality Reduction in Recommender System-A Case Study[C]//ACM WebKDD 2000 Workshop, 2000.

(责任编辑 芮国章)