小 型 微 型 计 算 机 系 统
Journal of Chinese Computer Systems

2017 年 4 月 第 4 期
Vol. 38 No. 4 2017

多层聚簇中基于协同过滤的跨类推荐算法

李瑞远[1],洪 亮[2],曾 承[1]

[1](武汉大学 计算机学院,武汉 430072)
[2](武汉大学 信息管理学院,武汉 430072)
E-mail:liruiyuan@whu.edu.cn

摘 要:大多数电子商务系统采用协同过滤的方法向用户推荐不同类别的商品.分析发现,相似用户对相关类别商品的喜好程度类似(称之跨类关联).因此,推荐时应同时考虑用户之间和商品之间的关联度.实际应用中,商品类别通常组织成多级目录,能够体现不同商品之间的层次关系.由于低层类别商品少,而高层类别商品多,因此不同层类别的数据稀疏度不同.为缓解现有推荐算法数据稀疏问题,本文提出一个高效多层挖掘算法,挖掘不同类别层次上用户-商品/类别的关联度.为提高推荐性能,还提出一个基于层次聚簇的跨类推荐通用框架,此模型扩展现有协同过滤算法.在真实数据集上的实验表明,本文提出的算法具有较高的准确率和效率.

关 键 词:多层聚簇;跨类挖掘;推荐系统;协同过滤
中图分类号:TP311 **文献标识码**:A **文 章 编 号**:1000-1220(2017)04-0657-07

Cross-category Recommendation Based on Collaborative Filtering in Multi-level Biclusters

LI Rui-yuan[1],HONG Liang[2],ZENG Cheng[1]
[1](Computer School,Wuhan University,Wuhan 430072,China)
[2](Information School,Wuhan University,Wuhan 430072,China)

Abstract:Most electronic commerce(E-commerce) systems use *Collaborative Filtering* (CF)-based methods to recommend items belonging to different categories. Market basket analysis has found that a group of like-minded users have similar tastes on items belonging to a subset of correlated categories(called *cross-category dependence*) rather than all the categories. Therefore,we should consider both user-to-user similarity and item-to-item similarity in recommendations. In real applications,items are usually organized into a multi-level taxonomy which provides hierarchical relationships between items and categories. Note that,the degree of data sparsity varies in different level of categories as there are more items in a category than those in any of its sub-categories. To alleviate the data sparsity problem of existing recommendation methods,we propose an efficient *multi-level biclustering* algorithm to mine user-item/category biclusters(i.e. cross-category dependence) at each level of the taxonomy. Then we propose a general framework for cross-category recommendation which extends existing CF methods by utilizing multi-level biclusters to improve their recommendation performance. Experiments on a real datasets show that our recommendation framework based on multi-level biclusters can recommend correlated items to a group of users precisely and efficiently.

Key words:multi-level biclustering;cross-category;recommendation;collaborative filtering

1 引 言

大数据时代的信息爆炸,使得推荐系统成为人们信息筛选必不可少的工具.大部分电子商务网站,例如亚马逊、eBay、淘宝和京东,利用推荐系统向用户推荐可能感兴趣的商品.这些商品通常属于多种类别,这些类别可以组织成层次目录.

本文研究跨类推荐问题,即向用户推荐属于不同类别的商品.协同过滤[1-3](Collaborative Filtering,CF)是推荐系统中采用最广泛的方法之一.当前协同推荐系统根据用户对所有商品的喜好程度找出用户之间的关系,并通过相似用户的评分向用户推荐商品.CF 方法通过用户相似度[4](user-to-user)或者商品相似度(item-to-item)做出预测.然而,分析发现,相似用户对相关类别的部分商品喜好程度类似[5,6],称为跨类关联.因此,在多层类别的情形下,考虑跨类关联可以提高基于 CF 方法的推荐性能.

图 1(见下页)是亚马逊(www.amazon.com)商品的类别,不同类别下的商品与类别/子类别形成层次结构,称为类别树.因此,跨类关联基于两方面.1)相似用户通常对属于同一父类别的商品具有类似喜好.2)相似用户对同层相关类别下的商品具有类似喜好.例如,喜欢歌曲"Hey Jude"的用户很可能也喜欢歌曲"Abbey Road",因为这两首歌曲属于同一类别"摇滚";喜欢摇滚歌曲的用户也可能喜欢动作电影,因为这两个类别在同一层,且相互关联.

为发现跨类关联关系,我们基于类别树从购买/评分记录

收稿日期:2016-02-02 收修改稿日期:2016-03-03 基金项目:国家自然科学基金青年项目(61303025)资助. 作者简介:**李瑞远**,男,1990 年生,硕士研究生,CCF 会员,研究方向为数据挖掘、机器学习、云计算;**洪 亮**,男,1982 年生,博士,副教授,CCF 会员,研究方向为数据管理、社会网络;**曾 承**,男,1978 年生,博士,副教授,CCF 会员,研究方向为服务计算,信息推荐.

中挖掘出用户-商品(user-item)和用户-类别(user-category)聚簇(cluster). 聚簇是指购买/评分记录中具有强关联关系的一组记录. 由于类别 c 下的用户购买/评分的记录数一定大于其下任何商品/子类的记录数,因此,不同层级之间的用户-类别交易记录的稀疏程度不同. 如图 1 所示,第 1 层的用户-类别交易记录比第 0 层的浓得多,在第 0 层,只有小部分的用户-类别能够覆盖. 因此,我们应该在不同的层中挖掘聚簇,也就是说,通过挖掘高层更浓密的用户-类别记录,来解决低层数据稀疏问题. 然而,从大规模用户-商品/类别的交易记录中挖掘多层聚簇并不容易,因为聚类问题是一个 NP 完全问题[10]. 此外,多层聚类过程中,在增加覆盖度的同时应保证聚类质量,即只挖掘那些能够提高推荐准确率的聚簇.

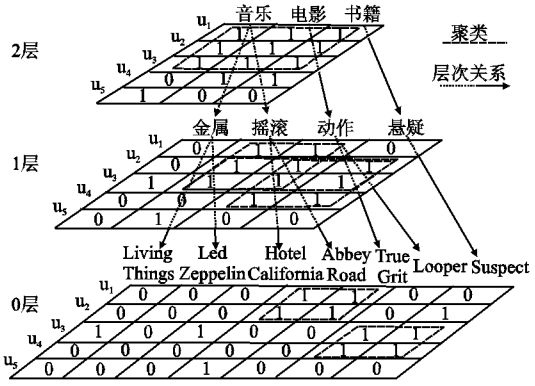


图 1 亚马逊多层聚簇例子

Fig. 1 Example of multi-level biclusters at amazon

本文提出的多层聚类算法,能够自底向上挖掘类别树中每层用户-商品/类别的聚簇. 如图 1 所示,本文提出的方法中,用户和商品/类别在同层和不同层上都能形成聚簇. 例如,u_2 与 u_1、u_3 可能对摇滚音乐与动作电影具有相近的兴趣,同时,u_2 与 u_3、u_4 对动作电影和悬疑书籍有相近的兴趣. 注意到,高层聚簇比低层聚簇具有更高的覆盖度,其质量却小于更细粒度的低层聚簇,因为可以用低层更详细的商品/类别信息推测用户的兴趣. 例如,在推荐时知道用户喜欢摇滚音乐比知道其喜欢音乐更有用.

此外,本文基于多层聚簇,扩展传统的 CF 方法,提出一个推荐框架,旨在提升基于 CF 方法的推荐系统性能. 基于多层聚簇的推荐框架(Recommending with Multi-level Biclusters,RMB)考虑了不同聚簇的权重. 聚簇的权重取决于它在类别树的层数以及其内用户和商品的数目.

本文主要贡献如下:

1)提出一个基于频繁项挖掘的多层聚簇算法,能够在类别树上高效挖掘所有聚簇.

2)基于多层聚簇,提出一个推荐框架,提升 CF 方法的推荐性能.

3)在大规模真实数据集的实验结果表明,本文提出的推荐框架能够提升传统 CF 方法的 top-K 推荐准确率. 此外,多层聚簇算法比现有方法具有更高效率.

论文其余部分组织结构如下. 在第 2 部分介绍相关的研究工作. 第 3 部分中,简要介绍本文提出的解决方案框架,同时定义本文研究的问题. 第 4 部分描述多层聚簇算法. 第 5 部分详细介绍推荐框架. 第 6 部分是实验. 最后在第 7 部分做出总结.

2 相关工作

2.1 协同过滤

协同过滤可以分成两类:基于内存的 CF 方法和基于模型的 CF 方法[3]. 基于内存的 CF 方法在早期推荐系统中广泛使用,如 Tapestry[1] 和 GroupLens[2],以及电子商务系统,如亚马逊[7],因为其易于实现,且具有较高效率. 然而,由于使用原始数据(通常非常稀疏)计算相似度,基于内存的 CF 方法会遇到数据稀疏的问题. 此外,随着数据集的增大,用户数目和商品数目的增多,基于内存的 CF 方法的扩展性欠佳. 为克服其不足,提出了基于模型的 CF 方法. 基于模型的 CF 方法首先利用训练数据训练模型,然后基于模型进行预测. 基于模型的 CF 方法包括贝叶斯 CF 模型[8]、潜在语义 CF 模型[9]、聚类 CF 模型[10-14]、基于时序行为的 CF 模型[15-16] 和多模型融合的 CF 方法[17] 等. 最近提出的许多方法[18-19] 开始利用类别来解决数据稀疏问题. 然而与本文提出的方法不同的是,这些方法在推荐过程中并没有使用聚类技术. 聚簇 CF 模型更强调数据稀疏性、扩展性及其他问题[3],因此本文集中精力设计一个聚簇 CF 模型.

2.2 聚簇 CF 模型

聚簇 CF 模型从用户评分/购买记录中挖掘出聚簇,以少量数据而非整个数据集来进行协同过滤. 有两种不同的聚簇 CF 模型[14]. 一种是单边聚簇模型,只考虑用户之间或者商品之间的相似度. Sarwar 等[10] 首先将数据划分成聚簇,然后对每个聚簇利用基于内存的 CF 算法预测. *RecTree* 算法[20] 将评分用户不断划分成两个子聚簇,然后使用活跃用户对应的叶节点进行预测. O'Connor 等[11] 从评分数据中找出商品聚簇,利用商品聚簇进行推荐. Ungar 和 Foster[21] 利用吉布斯采样(Gibbs sampling)对用户和商品分别聚类.

单边聚簇模型忽略了用户和商品之间的关系. 因此,提出了双边聚簇 CF 模型. George[12] 等提出一个基于 CF 的联合聚簇(co-clustering)模型,能够同时挖掘用户-商品的聚簇. 联合聚簇和单边聚簇都不允许两个聚簇之间重叠,每个用户或者商品只能属于单个聚簇. 论文[14]指出,一个用户可能对不同的商品集感兴趣,一个商品也可能引起多个用户的兴趣. 鉴于此,有必要允许聚簇之间的重合,也就是说,每个用户和商品可以属于多个聚簇.

与本文最相关的是双向聚簇(biclustering)CF 模型,它允许聚簇之间重合. 双向聚簇算法在生物领域取得了广泛研究[22,23]. Symeonidis 等[13] 将双向聚簇应用到了协同过滤. 双向聚簇揭示了用户与商品的关系,捕获了部分匹配用户的偏好. 近年来提出的 MCoC 算法[14] 能够找到用户-商品的子群(也就是双边聚簇). MCoC 算法利用传统 CF 方法建立一个统一框架,任何用户或者商品能够属于多个子群. 这与双边聚簇 CF 模型类似. 传统聚簇 CF 模型不足在于没有考虑聚簇之间的层次关系. 当数据稀疏时,只覆盖小部分用户和商品. 此外,现有聚簇算法在处理大规模数据时效率不足.

2.3 跨领域推荐

近年来跨领域推荐引起了广泛关注. Li 等[24]提出了评分矩阵模型来解决跨领域协同过滤. 他们最近的研究中[25],通过利用其他领域的评分矩阵来缓解数据稀疏问题. Pan 等[26]提出了主矩阵迁移模型框架来解决 CF 推荐中的数据稀疏问题. 这些方法在一定程度上可以解决单个领域上的跨类推荐问题,但是没有考虑到不同类之间的相互联系. 此外,现有的跨领域推荐算法要求数据是稠密的,然而,现实世界中数据通常非常稀疏.

3　预备知识

3.1　问题定义

给定 m 个商品 n 个用户,利用"用户-商品-交互"矩阵(User-Item-Interaction,UII)记录用户的购买记录,包括显式交互(评分/购买等)和隐式交互(点击/浏览等),用 0 或 1 表示. 为简便起见,我们集中于二进制评分矩阵. 但本文提出的双向聚簇算法也适用于实值数据.

本文考虑 $h+1$ 层树状结构的商品类别,0 层表示商品,1 层到 h 层表示商品类别. 第 l 层($1 < l \leqslant h$)类别对应第 $l-1$ 层若干个子类别. 商品类别树由领域专家定义.

定义 1. UCI 矩阵. 用户-类别-交互(User-Category- Interaction)矩阵记为 UCI,第 l 层($1 \leqslant l \leqslant h$)记为 UCI_l,其中每行表示一个用户,每列表示第 l 层中的一个商品类别,每个单元格 $\mathrm{UCI}_l[i][j]$ 记录了第 i 个用户与 l 层第 j 个类别的评分情况. 令 UCI_0 为 UII.

定义 2. 模式. 推荐系统中,所有类别/商品的集合记为 $U=\{v_1,v_2,\cdots,v_m\}$. 模式 P 是 U 的子集,即 $P \subseteq U$. 若用户 u 对 P 中所有商品/类别都评分过,则称 u 支持 P.

定义 3. 频繁模式. 模式 P 的支持度是指支持 P 的用户数,记为 $sup(P)$. 如果 $sup(P)$ 的支持度大于用户给定阈值 $minsup$,那么 P 称为一个频繁模式.

定义 4. 双向聚簇. 第 l 层的双向聚簇 B_{li} 是 UCI_l 的一个频繁模式,其中包含 m 个用户,n 个类别/商品,$m \geqslant minsup$,$n \geqslant 2$,$minsup$ 是频繁项的最小支持度. B_{li} 中用户的集合记为 $U(B_{li})$,类别/商品的集合记为 $C(B_{li})$. 为方便起见,双向聚簇简称为聚簇.

本文研究的问题为:给定 u 的 UCI_0 矩阵和商品类别树,基于多层聚簇,预测 u 对商品的评分,然后推荐 top-K 个最高评分的商品给用户 u.

3.2　算法框架

算法分成离线处理和在线处理两部分. 离线处理过程中,多层聚簇算法从大规模的数据中挖掘出所有频繁模式. 与传统聚簇算法不同的是,频繁模式挖掘能够找出所有的聚簇,并能够判断聚簇之间是否重合. 为叙述简便,下文中的频繁模式与聚簇含义相同. 多层聚簇算法首先从 UCI_0 矩阵中挖掘聚簇,对于那些不能被第 0 层聚簇覆盖的用户和商品,从 UCI_1 上进行挖掘. 此过程不断向上重复,直到所有用户和商品都被覆盖到,或者达到了顶层.

多层聚簇结束后,每个用户和商品至少被一个聚簇覆盖. 在线过程中,基于传统 CF 方法的推荐框架结合多层聚簇进行 top-K 推荐. 具体来说,对于每个用户 u,利用每个覆盖到 u 的聚簇 B_i 使用 CF 方法,预测 u 对 B_i 内所有商品的评分;然后使用一个相似度计算模型整合所有的结果;最后选择 top-K 个商品推荐给 u.

4　多层聚簇

这一小节,我们介绍多层聚簇算法,在不同的类别层上高效挖掘频繁模式. 多层聚簇算法自底向上从 UCI 矩阵中挖掘同时包含用户和商品的聚簇. 由于本文提出的推荐框架利用多层聚簇扩展传统的 CF 方法,因此,推荐质量与聚簇质量相关. 多层聚簇算法能够大大提高聚簇的覆盖度和质量,同时具有较高效率.

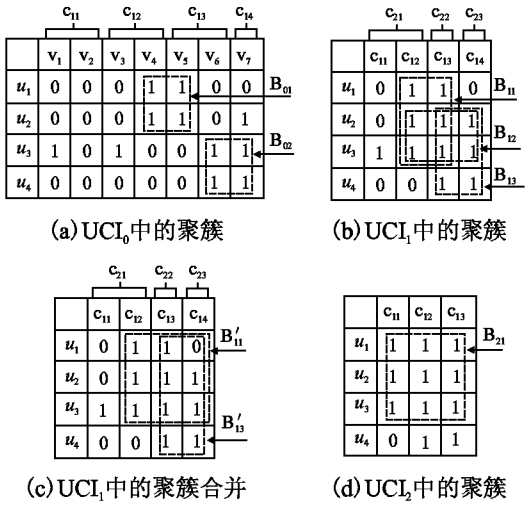


图 2　多层聚簇例子

Fig. 2　Example of multi-level biclustering

类别树中,叶子节点是商品,商品类别是内部节点. UCI_0 矩阵可以从叶子层的用户-商品-交互数据中获得,而 UCI_l 矩阵($1 \leqslant l \leqslant h$)可以由 UCI_{l-1} 中求出. 算法 1 描述了这一过程. 图 2(a)是 UCI_0 矩阵,对应图 1 中第 0 层的商品. 图 2(b)、图 2(c)分别显示 UCI_1 矩阵和 UCI_2 矩阵. 对于第 l 层的用户 u 和商品类别 c,UCI_l 矩阵对应的值为 1 当且仅当 u 至少评论了属于 c 中的一个商品.

为了从多层 UCI 矩阵中找出聚簇,一种简单的办法是针对每层 UCI 矩阵,直接使用现有的频繁模式挖掘算法[27-29]. 然而,这种方法存在许多问题. 首先,现有方法挖掘出的聚簇质量不高. 传统模式压缩算法如[28,29]在推荐时未充分考虑模式间的关系. 例如图 2(b)中,我们可以从 UCI_1 挖掘出 B_{11},B_{12} 和 B_{13},但是 B_{11} 和 B_{12} 之间的距离很小. 传统模式挖掘算法通常丢弃 B_{11}. 这种情况下,u_1 不再属于任何聚簇了,这会减弱推荐质量. 此外,由于每层的数据稀疏度不同,不能定义一个全局的支持度. 因此,有必要设计一个动态支持度机制.

算法 1. 构建多层 UCI 矩阵

输入:UCI_0 矩阵;h 层类别树

输出:UCI_l 矩阵($1 \leqslant l \leqslant h$)

1) **for** $l=1 \cdots h$ **do**

2)　**for** each user u_i and category c_{ij} **do**

3)　　**if** u_i rates/buys any item belonging to c_{ij} **then**

4) $UCI_l[i][j] = 1$;
5) **else**
6) $UCI_l[i][j] = 0$;

其次,传统方法引入了额外的开销. 对于第 l 层的聚簇 B_{li},在第 $l+1$ 层可能存在聚簇 $B_{(l+1)j}$,使得 $U(B_{(l+1)j}) = U(B_{li})$,并且 $C(B_{(l+1)j})$对应的子类别集合与 $C(B_{li})$相同. 正如第 1 节所述,我们从底向上挖掘聚簇,因此,如果在第 l 层挖掘出了 B_{li},那么在 $l+1$ 层就没有必要挖掘 $B_{(l+1)j}$.

本文采用[28]中如下定义.

定义 5. 闭模式. 模式 P_1 是封闭的,当且仅当不存在模式 P_2,满足 $P_1 \subseteq P_2$,并且 $sup(P_1) = sup(P_2)$.

定义 6. 模式距离. 两个封闭模式 P_1、P_2 的距离定义如下:$D(P_1,P_2) = 1-|U(P_1) \cap U(P_2)|/|U(P_1) \cap U(P_2)|$. 其中,$U(P)$表示评论/购买过 P 中对应商品的用户集合.

定义 7. δ-覆盖. 模式 P_1 被模式 P_2 δ-覆盖当且仅当 $P_1 \subseteq P_2$,并且 $D(P_1,P_2) \leqslant \delta$. 其中 δ 是用户自定义参数.

为便于算法描述,本文提出如下定义.

定义 8. δ-容忍最大聚簇(δ-Tolerance Maximum Bicluster, δ-TMB). 给定 l 层的闭模式 P_1,所有被 P_1 δ-覆盖的模式集为 $S(P_1)$. 那么 l 层的 δ-TMB 是指模式 B_l,满足 $C(B_l) = \cup C(P'_l)$,并且 $U(B_l) \supseteq \cup U(P'_l)$,其中 $P'_l \in S(P_1)$.

定义 9. 完全覆盖类别. l 层的类别 c_l 是一个完全覆盖类别,当且仅当对于每个用户 u 和 c_l 的子类别 c_{l-1},(u,c_{l-1})要么被至少一个 l'层($0 \leqslant l' \leqslant l-1$)的聚簇覆盖,要么无法被任何层的聚簇覆盖.

为克服传统挖掘算法聚簇质量的不足,本文提出以传递最小支持度挖掘 δ-TMB 算法. 给定 0 层的最小支持度阈值 $minsup$,第 l 层的传递最小支持度定义为 $minsup/\rho^l$,其中 $0 \leqslant \rho \leqslant 1$ 是传递因子. 例如图 2(b)中,u_1 很可能评分/购买类别 c_{14}下的商品,因此可将 B_{11} 和 B_{12} 合并成一个 δ-TMB B'_{11}. 传递因子 ρ 取决于类别的层数和粒度.

为提高聚簇算法的效率,提出完全覆盖类别的概念. 因此,我们不必挖掘那些只包含完全覆盖类别 c_l 的聚簇. 因为对于用户 u,可以利用低层的聚簇推荐 c_l 中的商品给 u,通常这样推荐更有效. 忽略那些仅包含完全覆盖类别的聚簇能够节省很多时间. 以图 2(c)为例,类别 c_{22} 和 c_{23} 是完全覆盖类别,因为对于用户 u_1、u_2、u_3、u_4 和子类别 c_{13}、c_{14},所有的用户-类别对都被 B'_{11} 和 B'_{13} 覆盖. 注意(u_5,c_{14})不能被任何聚簇覆盖. 因此,我们不必挖掘第 2 层仅仅只覆盖 c_{22} 和 c_{23} 的聚簇.

推论 1. 挖掘最小数目的多层聚簇是一个 NP 难问题.

证明:根据[28],代表模式的最优化问题对应了原始的集合覆盖问题. 每个 δ-TMB 与一个代表模式对应,因此多层聚簇问题也与集合覆盖问题对应.

鉴于推论 1,我们使用贪心算法来挖掘多层聚簇. 本文使用结合 FP-growth[30] 和集合覆盖的 RPlocal[31] 算法,但本文也可以使用其他的 FP-growth 算法. 算法 2 描述了多层聚簇算法.

算法 2. 挖掘多层 δ-TMB
输入:UCI_l 矩阵($0 \leqslant l \leqslant h$);第 0 层的最小支持度 minsup,传递因子 $\rho(0 \leqslant \rho \leqslant 1)$
输出:每层 δ-TMB 集

1) Scan UCI_0 matrix, initial FP-tree $F(0)$;
2) Call RPlocal Algorithm to mine δ-TMB(0);
3) **for** $l = 1 \cdots h$ **do**
4) Scan UCI_l matrix, initial FP-tree $F(l)$;
5) **for** each category c_l that all sub-categories exist in δ-TMB($l-1$) **do**
6) Pruning $F(l)$ by cutting c_l notes;
7) Call RPlocal Algorithm to mine δ-TMB(l);
8) **return** all δ-TMB;

首先利用 UCI_0 求第 0 层的 δ-TMB 集(1-2 行). 对于每个 UCI 矩阵,先建立一颗 FP 树,然后将在下一层 δ-TMB 中已经覆盖的所有类别的 FP 节点剪枝,再调用 RPlocal 算法挖掘当期层的 δ-TMB(3-7 行). 最后返回所有层 δ-TMB 集合.

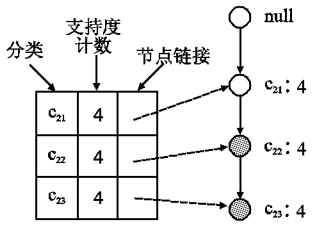


图 3 部分 FP 树
Fig. 3 Conditional FP-tree

例 1. 根据定义 9,c_{22} 和 c_{23} 是完全覆盖类别. 在算法 2 中,为 UCI_2 建立了一颗 FP 树,如图 3 所示. 虽然$\{c_{22},c_{23}\}$是一个频繁模式(也就是聚簇),但是在剪枝过程中会把节点 c_{22} 和 c_{23} 过滤,因此不会挖掘出$\{c_{22},c_{23}\}$.

5 利用多层聚簇推荐

5.1 推荐框架

这一节阐述利用现有协同过滤技术结合多层聚簇的推荐框架. 本文提出的推荐框架首先对每个聚簇应用 CF 方法,预测每个用户在每个聚簇中的缺失值. 然后整合每个预测结果,并进行推荐.

对于原始 CF 方法,输入 UCI_0 矩阵,输出矩阵中缺失的预测值. 然而,本文提出的框架中,输入是多个 UCI 矩阵(除第一层 UCI_0). 因此,对于每个聚簇 B_l,应将之转化成 UCI_0 中的聚簇$\overline{B_l}$. 此过程是算法 1 的逆过程,即把第 l 层的类别 c_l 替换成 0 层与之对应的商品. 这个过程在线下执行,因此不会影响在线性能. 至此,不同层的聚簇能够转化成下层的 UCI 矩阵,因此我们可以利用原始 CF 方法进行预测. 我们应该集中精力找到那些高质量的聚簇.

最后需要解决的问题是合并不同的聚簇产生的预测结果. 对于用户-商品对(u,v),有 3 种不同的情形.

情形 1. (u,v)不被任何聚簇覆盖. 利用传统 CF 方法预测用户 u 对 v 的评分.

情形 2. (u,v)被一层或多层的聚簇覆盖,且最低层只有一个聚簇覆盖. 取最低层的聚簇 B_l 对应的$\overline{B_l}$,利用 CF 方法预测 u 对 v 的评分.

例 2. 在图 2(a)中,(u_3,v_4)被 B'_{11} 和 B_{21} 覆盖,这种情况下,在 B'_{11} 中使用 CF 方法进行预测.

情形 3. (u,v)被一层或多层的聚簇覆盖,且最低层有多个聚簇覆盖.令 $Pre(u,v)$为 u 对 v 的最终预测评分,$Pre(u,v,\overline{B}_{lk})$是利用聚簇 B_{lk}预测 u 对 v 的评分.假设在最低层有 p 个聚簇覆盖了(u,v),则 $Pre(u,v)$计算如公式(1).

$$Pre(u,v) = \sum_{k=1}^{p} W'(u,\overline{B}_{lk}) Pre(u,v,\overline{B}_{lk}) \quad (1)$$

$$W(u,\overline{B}_{lk}) = \frac{|U(\overline{B}_{lk})|}{|\bigcup_{k=1}^{p} U(\overline{B}_{lk})|} + \frac{|N(\overline{B}_{lk})|}{|\bigcup_{k=1}^{p} N(\overline{B}_{lk})|} \quad (2)$$

公式(1)中,$\overline{B}_{lk}$是 B_{lk}在 0 层对应的子矩阵,$W'(u,\overline{B}_{lk})$是 $W(u,\overline{B}_{lk})$规则化权重.公式(2)中,$U(\overline{B}_{lk})$是$\overline{B}_{lk}$的用户集,$N(\overline{B}_{lk})$是评价$\overline{B}_{lk}$中商品的用户数.公式(2)表示了用户 u 与$\overline{B}_{lk}$的关系.

本文的推荐框架与特定的 CF 方法无关,通过与不同的 CF 方法结合,可以预测单个用户的多个评分,然后推荐那些高评分的商品给用户.

5.2 Top-*K* 推荐

本节介绍如何使用本文的推荐框架进行 top-K 推荐.

1. 利用算法 1 建立多层 UCI 矩阵.

2. 利用算法 2 挖掘多层聚簇.然后将 l 层($1 \leqslant l \leqslant h$) 的聚簇 B_l 映射到 0 层的聚簇.

3. 在每层的聚簇中运行 CF 方法预测那些缺失值,并利用本文的推荐框架整合所有预测结果.

4. 推荐 top-K 个评分最高的商品给用户.

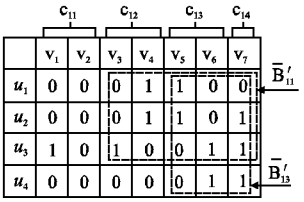


图 4　UCI_0 矩阵聚簇转换

Fig. 4　Transformed Biclusters in UCI_0 Matrix

例 3. 离线处理过程中,基于图 1 亚马逊的类别树建立多层 UCI 矩阵,如图 2 所示.然后挖掘多层聚簇如图 2(a)、2(b)和 2(c)所示.假设预测(u_2,v_5)的缺失值.(u3,v5)被图 2(c)中的B'_{11}和B'_{13}覆盖,而B'_{11}和B'_{13}分别对应图 4 中的$\overline{B}'_{11}$和$\overline{B}'_{13}$.根据本文的推荐框架,$W(u_3, B'_{11}) = 3/4 + 3/3 = 1.75$,$W(u_3,\overline{B}'_{11}) = 3/4 + 2/3 \approx 1.42$,而 $W(u_3, B'_{13}) \approx 0.55$,$W(u_3,\overline{B}'_{13}) \approx 0.45$.在$\overline{B}'_{11}$和$\overline{B}'_{13}$分别利用基于用户的 CF 方法,可以预测 $Pre(u_3,v_5,\overline{B}'_{11})$和 $Pre(u_3,v_5,\overline{B}'_{13})$,最后利用公式(1)计算 $Pre(u_3,v_5)$.

6 实　验

6.1 数据集

实验用到了大规模真实数据集:京东.京东数据集(JD)通过爬取 www.jd.com 收集得到.京东是中国最大的电子商务网站之一,用户可以购买属于不同类别的商品,然后提交商品评价和评分(评分在 1-5 之间).JD 包括 64,425 个用户在 532,786 个不同商品上 2,655,797 条评分记录.我们随机选择 10,000 个用户以及他们的评分作为实验数据集.这些商品所属类别组织成 5 层的类别树,非叶子层中,每层的类别数目分别为 17,138,818,8197.用户评分分布如图 5 所示,可知,数据集的评分非常稀疏,且符合幂律分布[26].

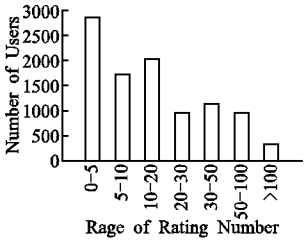


图 5　用户评分分布

Fig. 5　User rating distribution

6.2 对比方法

我们将多层聚簇(Multi-Level Biclusters,MLB)算法与两种 CF 方法进行比较:

1)基于用户的 CF(User-based CF,UB-CF):User-based CF 利用用户之间的相似度来预测目标用户的评分.用户相似度利用用户评分商品的 Jaccard 相似度度量.

2)基于商品的 CF(Item-based CF,IB-CF):Item-based CF 使用商品之间的相似度来预测目标用户的评分.商品相似度使用评价该商品的用户集的 Jaccard 相似度度量.

本文的方法 MLB 与论文[14]的单层 CF 方法(记为 SLB)以及 User-based CF(记为 CF-u)、Item-based CF(记为 CF-i)进行比较.由于 MLB 可以使用两种 CF 方法,因此 MLB 和 SLB 可以分成 MLB-u、MLB-i、SLB-u 和 SLB-i.

6.3 评价标准

MAE 和 RMSE:我们使用平均绝对误差(Mean Absolute Error,MAE)和均方根误差(Root Mean Square Error,RMSE)衡量评分预测的准确率.MAE 和 RMSE 的值越小,表明预测越准确.MAE 和 RMSE 定义如下.

$$MAE = \frac{\sum_{i,j} |RT_{ij} - \widehat{RT}_{ij}|}{N} \quad (3)$$

$$RMSE = \sqrt{\frac{\sum_{i,j} (RT_{ij} - \widehat{RT}_{ij})^2}{N}} \quad (4)$$

其中 RT_{ij}和$\widehat{RT}_{ij}$分别表示用户 u_i 对商品 v_j 的真实评分和预测评分,N 表示预测数量.

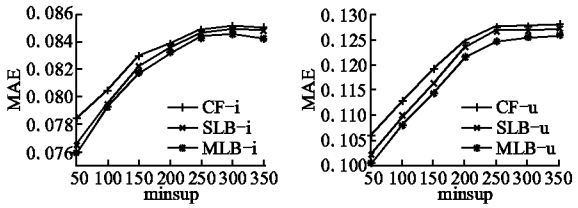
6.4 参数设置

参数选择非常重要.实验中默认时,支持度阈值 *minsup* 为 50,传递因子 ρ 为 0.8,模式误差 δ 为 0.1,K 为 5.采用 10-交叉验证方法随机选择 90% 的数据作为训练集来预测剩下 10% 的评分.

6.5 *minsup* 的影响

下页图 6 和图 7 分别显示 MAE 和 RMSE 随着 *minsup* 变化情况.为显示清楚,已将 IB-CF 和 UB-CF 分开.由图可以看出,整体而言,在 JD 数据集上,IB-CF 要优于 UB-CF,因为同等 *minsup* 情况下,IB-CF 的 MAE 和 RMSE 比 UB-CF 的小.*minsup* 越小,推荐的效果越佳,因为挖掘的聚簇质量更好.在

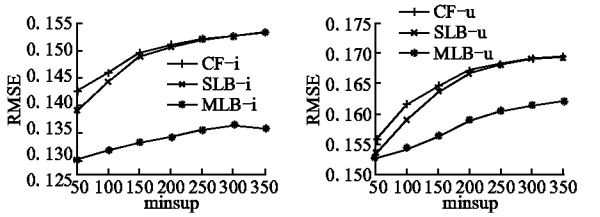
minsup 从 50 到 350 范围内,不管是 IB-CF,还是 UB-CF,MLB 算法优于 SLB 算法,也优于传统的 CF 方法. 注意到,当 *minsup* 越大时,SLB 算法与传统 CF 方法的 RMSE 越接近.



(a)IB-CF's MAE vs *minsup* (b)UB-CF's MAE vs *minsup*

图 6 MAE 随 *minsup* 变化

Fig. 6 MAE vs *minsup*



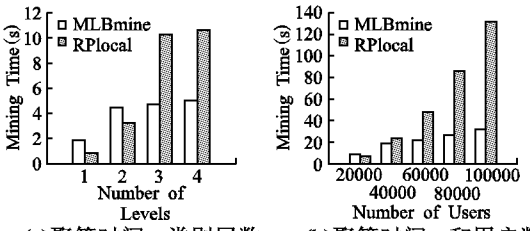
(a)IB-CF's RMSE vs *minsup* (b)UB-CF's RMSE vs *minsup*

图 7 RMSE 随 *minsup* 变化

Fig. 7 RMSE vs *minsup*

6.6 效率和扩展性

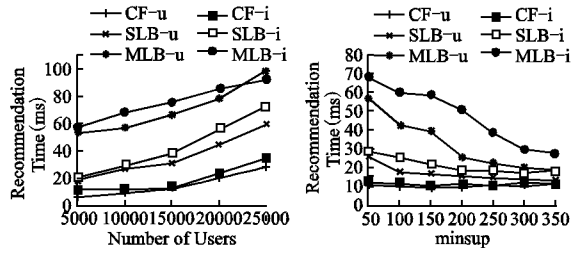
为衡量多层聚簇算法(记为 MLBmine)的性能,我们将 MLBmine 和 RPlocal 算法的挖掘时间进行比较. 从图 8(a)中可以看出,随着类别树层数增加,MLBmine 比 RPlocal 的时间开销越来越少,因为 MLBmine 会将冗余的聚簇过滤,MLBmine 在较高层挖出聚簇比 RPlocal 少. 注意到,当层数为 1 时,MLBmine 的时间开销比 RPlocal 多,因为 MLBmine 需要额外的时间检查类别是否被完全覆盖.



(a)聚簇时间vs类别层数 (b)聚簇时间vs和用户数

图 8 聚簇时间比较

Fig. 8 Comparison of clustering time



(a)推荐时间vs用户数 (b)推荐时间vs最小支持度

图 9 推荐时间比较

Fig. 9 Comparison of recommendation time

如图 8(b)所示,当数据量越来越多时,MLBmine 比 RPlocal 的优势越来越明显,因为过滤的类别会更多. 这体现了我们提出的方法具有更高的效率和更好的扩展性.

同时,我们还比较了不同方法的在线推荐性能. 如图 9 所示,MLB 算法比 SLB 和传统的 CF 方法需要更多的时间. 因为 MLB 在推荐时考虑了更多的聚簇,因此需要更多的计算时间通过不同的聚簇推荐,并整合多个预测结果. 然而,图 9(a)显示推荐时间仍然在一个可以接受的范围内,随着用户的增加,仍具有较好的扩展性. 例如,当用户达到 25000 时,MLB-u 的推荐效率仍小于 100ms. 图 9(b)显示随着 *minsup* 的增加,推荐时间越来越少,因为若 *minsup* 越大,找出的聚簇就越少.

7 总 结

用户在相关类别上的喜好程度类似,称为跨类关联. 跨类关联可以通过挖掘用户-类别-交互数据获得. 现有 CF 聚簇模型没有考虑聚簇间的层次关系,也没有利用类别信息,因此难以找到高质量的聚簇,同时还会遇到数据稀疏问题. 本文提出了一个推荐框架,利用多层聚簇扩展现有 CF 模型. 为快速找出高质量的多层聚簇,本文充分利用类别的层次关系,提出一个多层聚簇算法. 在真实数据上的实验表明,本文提出的推荐框架能够提高传统 CF 方法的性能,同时具有较高的扩展性.

References:

[1] Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry [J]. Communications of the ACM, 1992, 35(12): 61-70.

[2] Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of netnews[C]. Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, ACM, 1994: 175-186.

[3] Su X, Khoshgoftaar T M. A survey of collaborative filtering techniques[J]. Advances in Artificial Intelligence, 2009, 2009(12): 1-20.

[4] Rong Hui-gui, Huo Sheng-xu, Hu Chun-hua, et al. User similarity-based collaborative filtering recommendation algorithm[J]. Journal on Communications, 2014, 35(2): 16-24.

[5] Gary Russell, Ann Petersen. Analysis of cross category dependence in market basket selection[J]. Journal of Retailing, 2000, 76(3): 367-392.

[6] Manchanda P, Ansari A, Gupta S. The "shopping basket": a model for multi-category purchase incidence decisions[J]. Marketing Science, 1999, 18(2): 95-114.

[7] Linden G, Smith B, York J. Amazon. com recommendations: Item-to-item collaborative filtering[J]. Internet Computing, IEEE, 2003, 7(1): 76-80.

[8] Miyahara K, Pazzani M J. Collaborative filtering with the simple Bayesian classifier[M]. PRICAI 2000 Topics in Artificial Intelligence. Springer Berlin Heidelberg, 2000: 679-689.

[9] Hofmann T. Latent semantic models for collaborative filtering[J]. ACM Transactions on Information Systems, 2004, 22(1): 89-115.

[10] Sarwar B M, Karypis G, Konstan J, et al. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering[C]. Proceedings of the Fifth International Conference on Computer and Information Technology, 2002, 1.

[11] O'Connor M, Herlocker J. Clustering items for collaborative filtering[C]. Proceedings of the ACM SIGIR Workshop on Recommender Systems, UCBerkeley, 1999, 128.

[12] George T, Merugu S. A scalable collaborative filtering framework based on co-clustering[C]. Proceedings of the 5th IEEE Conference on Data Mining, 2005: 625-628.

[13] Symeonidis P, Nanopoulos A, Papadopoulos A, et al. Nearest-biclusters collaborative diltering[J]. Lecture Notes in Computer Science, 2006, 11(1): 36-55.

[14] Xu B, Bu J, Chen C, et al. An exploration of improving collaborative recommender systems via user-item subgroups[C]. Proceedings of the 21st International Conference on World Wide Web, ACM, 2012: 21-30.

[15] Sun Guang-fu, Wu Le, Liu Qi, et al. Recommendations based on collaborative filtering by exploiting sequential behaviors [J]. Journal of Software, 2013, 24(11): 2721-2733.

[16] Dou Ling-yuan, Wang Xin-hua, Sun Ke. Collaborative filtering fusing label feature and time context[J]. Journal of Chinese Computer Systems, 2016, 37(1): 48-52.

[17] Huang Bin, Peng Zhi-ping. Recommendation method of multi-model combination based on the cascaded filtering[J]. Journal of Chinese Computer Systems, 2016, 37(1): 33-37.

[18] Ziegler , Lausen G, Schmidt-Thieme L. Taxonomy-driven computation of product recommendations [C]. Proceedings of the 13th ACM international conference on Information and Knowledge Management, ACM, 2004: 406-415.

[19] Koenigstein N, Dror G, Koren Y. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy [C]. Proceedings of the 5th ACM Conference on Recommender Systems, ACM, 2011: 165-172.

[20] Chee S H S, Han J, Wang K. Rectree: An efficient collaborative filtering method[M]. Data Warehousing and Knowledge Discovery, Springer Berlin Heidelberg, 2001: 141-151.

[21] Ungar L H, Foster D P. Clustering methods for collaborative filtering[C]. AAAI Workshop on Recommendation Systems, 1998, 1: 114-129.

[22] Cheng, Church. Biclustering of expression data[C]. Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, 2000, 8: 93-103.

[23] Madeira, Oliveira. Biclustering algorithms for biological data analysis: a survey[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2004, 1(1): 24-45.

[24] Li B, Yang Q, Xue X. Transfer learning for collaborative filtering via a rating-matrix generative model[C]. Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009: 617-624.

[25] Li B, Yang Q, Xue X. Can movies and books collaborate cross-domain collaborative filtering for sparsity reduction[C]. Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009, 9: 2052-2057.

[26] Pan W, Xiang E W, Liu N N, et al. Transfer learning in collaborative filtering for sparsity reduction[C]. Proceedings of the 26th Annual International Conference on Machine Learning, 2010, 10: 230-235.

[27] Pandey G, Atluri G, Steinbach M, et al. An association analysis approach to biclustering[C]. Proceedings of the 15th ACM SIGKDD international conference on Knowledge Discovery and Data Mining, ACM, 2009: 677-686.

[28] Xin D, Han J, Yan X, et al. Mining compressed frequent-pattern sets[C]. Proceedings of the 31st international conference on Very Large Data Bases, VLDB Endowment, 2005: 709-720.

[29] Liu G, Zhang H, Wong L. Finding minimum representative pattern sets[C]. Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012: 51-59.

[30] Han J, Pei J, Yin Y, et al. Mining frequent patterns without candidate generation: a frequent-pattern tree approach[J]. Data Mining and Knowledge Discovery, 2004, 8(1): 53-87.

附中文参考文献:

[4] 荣辉桂, 火生旭, 胡春华, 等. 基于用户相似度的协同过滤推荐算法[J]. 通信学报, 2014, 35(2): 16-24.

[15] 孙光福, 吴 乐, 刘 淇, 等. 基于时序行为的协同过滤推荐算法[J]. 软件学报, 2013, 24(11): 2721-2733.

[16] 窦羚源, 王新华, 孙 克. 融合标签特征和时间上下文的协同过滤推荐算法[J]. 小型微型计算机系统, 2016, 37(1): 48-52.

[17] 黄 斌, 彭志平. 基于级联过滤的多模型融合的推荐方法[J]. 小型微型计算机系统, 2016, 37(1): 33-37.