

融合用户社会地位和矩阵分解的推荐算法

余永红^{1,2,3} 高 阳^{2,3} 王 皓^{2,3} 孙栓柱⁴

¹(南京邮电大学通达学院 南京 210003)
²(计算机软件新技术国家重点实验室(南京大学) 南京 210023)
³(江苏省软件新技术与产业化协同创新中心(南京大学) 南京 210023)
⁴(江苏方天电力技术有限公司 南京 211100)
(yuyh.nju@gmail.com)

Integrating User Social Status and Matrix Factorization for Item Recommendation

Yu Yonghong^{1,2,3}, Gao Yang^{2,3}, Wang Hao^{2,3}, and Sun Shuanzhu⁴

¹(Tongda College, Nanjing University of Posts and Telecommunications, Nanjing 210003)
²(State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023)
³(Collaborative Innovation Center of Novel Software Technology and Industrialization (Nanjing University), Nanjing 210023)
⁴(Jiangsu Frontier Electric Technology Co. LTD., Nanjing 211100)

Abstract With the increasing popularity of online social network services, social networks platforms provide rich information for recommender systems. Based on the assumption that friends share more common interests than non-friends and users tend to accept the item recommendations from friends, more and more recommender systems utilize trust relationships of users to improve the performance of recommendation algorithms. However, most of the existing social-network-based recommendation algorithms ignore the following problems: 1) in different domains, users tend to trust different friends; 2) the degree of influence that a user is affected by their trusted friends is different in different domains since the user has different social status in different domains. In this paper, we first infer domain-specific social trust relation networks based on original users' rating information and social network information, and then compute each user's social status by leveraging PageRank algorithm for each specific domain. Finally, we propose a novel recommendation algorithm by integrating users' social status with matrix factorization model. Experimental results on real-world dataset show that our proposed approach outperforms traditional social-network-based recommendation algorithms.

Key words user social status; matrix factorization; recommendation algorithm; PageRank algorithm; social network

摘 要 随着社交网络服务的日益流行,社交网络平台为推荐算法提供了丰富的额外信息.假设朋友之间共享更多的共同偏好并且用户往往易于接受来自朋友的推荐,越来越多的推荐系统利用社交网络中

收稿日期:2016-09-18;修回日期:2017-03-07
基金项目:国家自然科学基金项目(61432008,61503178,61403208);江苏省自然科学基金项目(BK20150587);江苏省高等学校自然科学基金项目(17KJB520028)
This work was supported by the National Natural Science Foundation of China (61432008, 61503178, 61403208), the Natural Science Foundation of Jiangsu Province of China (BK20150587), and the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province (17KJB520028).

用户之间的信任关系来改进传统推荐算法的性能.然而,现有基于社交网络推荐算法忽略了2个问题:1)在不同的领域中,用户信任不同的朋友;2)由于用户在不同的领域内具有不同的社会地位,因此,用户在不同的领域内受朋友的影响程度是不同的.首先利用整体的社交网络结构信息和用户的评分信息推导特定领域社交网络结构,然后利用 PageRank 算法计算用户在特定领域的社会地位,最后提出了一种融合用户社会地位信息的矩阵分解推荐算法.在真实数据集上的实验结果表明:融合用户地位信息的矩阵分解推荐算法的性能优于传统的基于社交网络推荐算法.

关键词 用户社会地位;矩阵分解;推荐算法;PageRank 算法;社交网络

中图法分类号 TP181

随着互联网技术的不断发展,从海量数据中找到有价值的相关信息变得越来越困难.推荐系统^[1]通过分析用户的历史活动数据,挖掘用户的潜在偏好,为用户提供个性化的推荐服务,成为解决信息过载问题的有效手段,近年来受到学术界和工业界的广泛关注.推荐系统的典型应用包括 Amazon 和淘宝的商品推荐、Netflix 的电影推荐、Last.fm 的音乐推荐、LinkedIn 的朋友推荐、Google News 的新闻推荐等.这些互联网应用或者电商平台通过部署推荐系统,一方面满足了用户的个性化需求,减轻了信息过载的问题;另一方面提高了用户的忠诚度,增加了企业的营业收入.

在推荐系统的研究中,协同过滤算法是目前应用最广泛的推荐技术.协同过滤算法通过分析用户的历史反馈信息,预测用户未来的偏好.由于协同过滤算法仅需要用户的反馈信息,不需要用户的概要信息和项目的描述信息,因此协同过滤的方法独立于具体的应用领域,可以泛化到不同的应用领域中.然而,协同过滤算法存在数据稀疏、冷启动等问题.数据稀疏导致协同过滤算法不能准确地根据用户的评分数据计算用户或者项目之间的相似性,从而严重影响协同过滤推荐算法的准确性.冷启动指由于与新注册用户和新加入系统项目相关的评分数据较少,协同过滤推荐算法不能准确地找到相似用户或者项目,因而不能为新注册用户提供个性化推荐,或者将新加入系统的项目推荐给感兴趣的用户.

随着社交网络的出现,越来越多的推荐算法利用社交网络提供的丰富信息来改进传统推荐算法的性能,特别是解决传统协同过滤方法中的冷启动问题.基于社交网络的推荐算法一般假设社交网络中用户的偏好受到朋友的影响,并且朋友之间具有相似的偏好.典型的基于社交网络的推荐算法如 SoRec^[2], RSTE^[3], SocialMF^[4], TrustMF^[5], TrustSVD^[6]等.然而,已有的基于社交网络的推荐

算法忽略了2个事实:1)在不同的领域中,用户信任不同的朋友;2)用户不仅在不同领域中受到不同朋友的影响,而且不同用户受朋友影响的程度不同.例如在某个社交网络中,用户 u 信任用户 v ,用户 v 在餐饮方面是权威,但是在电子产品方面是入门者.用户 u 往往会接受用户 v 在餐饮方面的建议;相反,在电子产品方面则不会接受用户 v 的推荐,而会转向在电子产品领域的权威朋友征询意见.另外,当用户 v 向用户 u 推荐餐饮方面的产品时,如果用户 u 自身不精通餐饮方面的知识,或者说在餐饮方面具有较低的社会地位,那么用户 u 受到用户 v 的影响程度较大;相反,如果用户 u 对餐饮方面的知识也非常精通,换句话说,用户 u 在餐饮方面具有较高的社会地位,那么用户 u 受到用户 v 的影响程度则较小.因此,在推荐系统中,针对不同的领域,考虑用户在不同领域的社会地位可以更加准确地反映实际的推荐过程,从而有效地改进推荐算法的性能.

为了解决以上的问题,本文提出了融合用户社会地位的矩阵分解推荐算法.首先结合整体社交网络和用户的评分反馈信息,利用用户评分和用户信任关系共现的原则来推导特定领域的用户社交网络结构;然后利用 PageRank^[7]算法计算每个特定领域内用户的社会地位;最后以用户社会地位来约束矩阵分解过程.在真实数据集上的实验结果表明,本文提出融合用户社会地位的矩阵分解推荐算法的性能优于传统的基于社交网络推荐算法.

1 相关研究工作

本节主要介绍2类典型的推荐算法,包括协同过滤推荐算法和基于社交网络的推荐算法.

1.1 协同过滤推荐算法

协同过滤推荐算法仅需分析用户的历史活动记录(一般为用户-项目评分矩阵)来进行推荐,不需要

分析用户概要信息和项目描述信息,避免了复杂的自然语言处理过程,是目前应用最广泛的推荐技术.协同过滤推荐算法主要分为基于内存的协同过滤推荐算法、基于模型的协同过滤算法和混合推荐算法三大类^[1].

基于内存的协同过滤推荐算法使用整个用户-项目评分矩阵进行推荐.它的基本假设是:与当前活动用户(active user)爱好相似用户喜欢的项目,当前活动用户可能也喜欢;或者,与当前活动用户喜欢的项目相似的项目,用户也可能喜欢.协同过滤推荐算法首先通过某种相似度量指标找到与当前活动用户或者目标项目相似的用户或者相似的项目;然后利用相似用户对目标项目评分的权重平均或当前活动用户对相似项目评分的权重平均来预测当前活动用户对目标项目的评分.典型的基于内存协同过滤算法包括基于用户的协同过滤算法^[8]和基于项目的协同过滤算法^[9-10].

与基于内存协同过滤算法不同,基于模型的协同过滤算法使用统计和机器学习技术从用户-项目评分矩阵中学习一个预测模型,预测模型刻画用户的行为模式,然后利用预测模型进行推荐.典型的基于模型的协同过滤算法包括:基于贝叶斯网络推荐算法^[11]、基于聚类模型推荐算法^[12-14]、基于潜在语义分析推荐算法^[15-16]、基于关联规则推荐算法^[17]、基于深度学习推荐算法^[18-20].

基于内存协同过滤算法虽然易于实现而且预测质量较高,但是存在严重的可扩展性问题.随着用户数量和项目数量的不断增长,较差的在线预测性能使得基于内存协同过滤推荐算法越来越不适应大规模的电子商务平台.基于模型的协同过滤算法的响应速度往往比基于内存的协同过滤算法快,但是使用的理论模型复杂,有时并不能准确地拟合实际的数据,另外模型的线下学习和更新过程时间较长.

自 Netflix 竞赛以来,由于在处理大规模数据方面良好的可扩展性和准确的预测能力,基于矩阵分解的推荐算法^[21]受到学术界和工业界的广泛关注.基于矩阵分解的推荐算法认为仅有少量的隐藏因子影响用户的偏好和刻画项目的特征.因此,基于矩阵分解的推荐算法将用户和项目的特征向量同时映射到低维的隐藏因子空间中.在低维的隐藏因子空间中,由于用户偏好和项目特征之间的相关性可以直接计算,矩阵分解的推荐算法利用用户和项目的低维特征向量的内积来预测用户对项目的评分.典型

的基于矩阵分解的推荐算法包括 SVD++^[22], NMF^[23], MMMF^[24], PMF^[25], NPCA^[26]等.

由于用户-项目评分矩阵极度稀疏,导致协同过滤算法存在严重的冷启动问题.如,当活动用户的评分信息较少时,基于用户协同过滤推荐算法^[8]很难根据评分信息找到偏好相似的用户.类似地,基于项目的协同过滤推荐算法^[9-10]也不能准确地找到与目标项目相似的其他项目;基于矩阵分解的推荐算法很难准确地学习新注册用户或者新加入系统的项目的隐藏特征向量.

1.2 基于社交网络的推荐算法

传统协同过滤算法认为用户之间是独立的,忽略了用户的社会属性,即用户总是存在于一个社交圈子中,与社交圈子内的其他用户相互影响.社交网络的出现为解决协同过滤推荐算法中的数据稀疏和冷启动问题带了契机.基于社会学中的同质理论^[27]和社交影响理论^[28],研究人员提出了几种典型的基于社交网络的推荐算法,如 SoRec^[2], RSTE^[3], SocialMF^[4], TrustMF^[5], TrustSVD^[6]等.

Ma 等人^[2]提出了基于概率矩阵分解模型的推荐算法 SoRec. SoRec 集成了用户的评分信息和用户的社交网络信息,并通过用户评分信息和用户社交网络信息之间共享用户隐藏特征矩阵的方式来融合 2 种不同类型的信息源.为了更加准确直观地对推荐过程进行建模, Ma 等人^[3]进一步提出了基于社交网络的推荐算法 RSTE. RSTE 推荐算法假设用户的最终决策是用户自身偏好和朋友偏好的折中权衡,并通过集成参数来融合用户自身偏好和朋友的偏好.在文献[4]中, Jamali 等人提出了 SocialMF 推荐算法. SocialMF 推荐算法在矩阵分解模型中集成了信任的传递机制来改进推荐算法的准确性.在 SocialMF 中,信任传递机制对于处理推荐系统中的冷启动问题特别有效,因为新注册用户的隐藏特征向量依赖于此用户最相似邻居们的特征向量,而邻居们的特征向量可以从他们的评分信息中学习,所以,一定程度上确保了新注册用户的隐藏特征向量可以被学习到.在文献[5]中, Yang 等人集成用户评分信息和用户信任关系信息,提出了 TrustMF 推荐算法. TrustMF 认为用户一方面受到信任朋友的评分信息和评论的影响;另一方面,用户自身的评分信息和评论也会影响其他用户的决定. TrustMF 通过在信任关系矩阵上执行矩阵分解,将用户映射到 2 个不同的隐藏特征空间:信任特征空间和被信任

特征空间. 因此, 在 TrustMF 中, 每个用户同时由信任特征向量和被信任特征向量描述. 在文献[6]中, Guo 等人提出了基于社交网络的推荐算法 TrustSVD. TrustSVD 扩展了 SVD++ 算法, 在 SVD++ 算法的基础上, 综合考虑了评分和信任值的显式和隐式影响, 使用已评分项目的隐藏特征向量和信任朋友的隐藏特征向量来表示用户的隐藏特征向量. 在文献[29]中, 郭弘毅等人提出了融合社区结构和兴趣聚类的协同过滤算法. 上述提及推荐算法的实验结果表明, 社交网络信息有助于改进传统协同过滤推荐算法的性能, 特别是减轻冷启动问题.

另外, 为了建模信任关系的多面性, Yang 等人^[30]提出基于社交网络圈的推荐算法. 基于社交网络圈的推荐算法首先从评分数据和社交网络数据中推导特定领域的社交信任关系; 然后在每个领域内应用 SocialMF 算法学习用户隐藏特征向量和项目隐藏特征向量. Tang 等人^[31]认为用户可能向社交网络中朋友或者全局具有较高声誉的用户寻求建议, 他们同时利用局部和全局的社交关系, 提出了 LOCABAL 推荐算法. 不同于 Yang 等人^[30]提出基于社交网络圈的推荐算法, 本文提出的方法不仅考虑了不同领域的社交网络圈, 还考虑了不同社交网络圈内用户的社会地位存在差异, 不同社会地位的用户受到同一个领域内朋友的影响程度不同. LOCABAL 利用 PageRank 算法在整个社交网络中计算用户的信誉值, 但是未考虑不同领域内, 用户的信誉值是不同的. 在一个包含多个领域项目的推荐系统中, 来自于多种领域的社交关系混合在一起, 这种方法不符合实际情况. 与 LOCABAL 不同点在于, 本文提出的方法考虑了用户的信誉值或者社会地位在不同领域是不同的.

2 推荐问题描述及矩阵分解

2.1 推荐问题的形式化描述

基于社交网络的推荐系统往往包含 2 种不同类型的数据源: 用户-项目评分矩阵和社交网络信息. 用户-项目评分矩阵 $\mathbf{R}_{N \times M}$ 由 2 种实体集合组成: N 个用户集合 $U = \{u_1, u_2, \dots, u_N\}$ 和 M 个项目集合 $I = \{i_1, i_2, \dots, i_M\}$. $\mathbf{R}_{N \times M}$ 中的每项 r_{ui} 是用户 u 对项目 i 的评分. 原则上, 评分数据 r_{ui} 可以为任意的实数, 但通常情况下, 评分数据为整数, 而且 $r_{ui} \in \{0, 1, 2, 3, 4, 5\}$, 其中 0 值表示用户未对此项目进行评

分. 评分值越高意味着用户对当前项目越满意. 用户 u 评过的项目集合表示为 $I_u (I_u \subseteq I)$. 由于用户通常只对少量的项目进行评分, 用户-项目评分矩阵 $\mathbf{R}_{N \times M}$ 通常非常的稀疏. 例如 MovieLens100K 和 MovieLens1M 数据集中分别有 93% 和 95% 的评分项缺失, Netflix 数据集评分缺失项更高达 99%. 用户-项目评分矩阵的稀疏性是影响推荐准确性的重要因素.

社交网络信息通常表示为有向社会关系图 $G = (U, E)$, 其中 U 是用户集合, 边集合 E 表示用户之间的社交信任关系. $T_{u,v} \in [0, 1]$ 表示用户 u 和用户 v 之间信任权重, $T_{u,v} = 0$ 意味着用户 u 和用户 v 之间没有建立信任关系. 推荐系统中所有用户之间的信任关系组成信任关系矩阵 $\mathbf{T}$. 需要注意的是: 由于用户之间信任关系往往不是相互的, 所以信任关系矩阵 $\mathbf{T}$ 通常是非对称的.

基于社交网络推荐系统的任务是利用用户-项目评分信息和社交网络信息预测当前活动用户 u 对目标项目 i 的评分值 $\hat{r}_{ui}$.

2.2 矩阵分解

由于在处理大规模用户-项目评分矩阵 $\mathbf{R}$ 方面的有效性, 矩阵分解技术在推荐系统中被广泛采用. 矩阵分解的目标是通过分析用户-项目评分矩阵 $\mathbf{R}$, 学习用户的隐藏特征矩阵 $\mathbf{P}$ 和项目隐藏特征矩阵 $\mathbf{Q}$, 然后利用 $\mathbf{P}$ 和 $\mathbf{Q}$ 预测 $\mathbf{R}$ 中的缺失项. 形式化地, 设用户的隐藏特征矩阵和项目隐藏特征矩阵分别为 $\mathbf{P} \in \mathbb{R}^{K \times N}$ 和 $\mathbf{Q} \in \mathbb{R}^{K \times M} (K \ll \min(N, M))$, 其中 K 为隐藏特征向量的维数, 矩阵分解的方法将用户-项目评分矩阵 $\mathbf{R}$ 分解为 2 个低维的隐藏特征矩阵 $\mathbf{P}$ 和 $\mathbf{Q}$, 然后用 $\mathbf{P}$ 和 $\mathbf{Q}$ 的乘积来近似评分矩阵 $\mathbf{R}$, 即:

$$\mathbf{R} \approx \mathbf{P}^T \mathbf{Q} = \begin{pmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_N^T \end{pmatrix} (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M), \quad (1)$$

其中, 列向量 $\mathbf{p}_u$ 和 $\mathbf{q}_i$ 分别表示用户 u 和项目 i 的隐藏特征向量. 对于评分矩阵 $\mathbf{R}$ 中的缺失项 $\hat{r}_{ui}$, 使用 $\mathbf{p}_u$ 和 $\mathbf{q}_i$ 的内积来预测:

$$\hat{r}_{ui} = \mathbf{p}_u^T \mathbf{q}_i. \quad (2)$$

矩阵分解通过最小化误差平方和损失函数来学习隐藏特征矩阵 $\mathbf{P}$ 和 $\mathbf{Q}$, 即:

$$\min_{\mathbf{P}, \mathbf{Q}} \frac{1}{2} \sum_{(u,i) \in \Omega} (r_{ui} - \mathbf{p}_u^T \mathbf{q}_i)^2 + \frac{\lambda_1}{2} \|\mathbf{P}\|_F^2 + \frac{\lambda_2}{2} \|\mathbf{Q}\|_F^2, \quad (3)$$

其中, Ω 为 $\mathbf{R}$ 中可观测(用户,项目)对集合, $\|\cdot\|_F^2$ 为 Frobenius 范式;正则化项 $\|\mathbf{P}\|_F^2$ 和 $\|\mathbf{Q}\|_F^2$ 用来避免过拟合; λ_1 和 λ_2 为正则化项参数,用来控制正则化项对隐藏特征向量的影响。

一般使用梯度下降或者随机梯度下降的算法求解式(3)定义的目标函数的局部最优解。另外,从贝叶斯的角度出发,上述描述的矩阵分解算法等价于概率矩阵分解算法(PMF)^[25]。

本文扩展了上述矩阵分解方法,特别是在其中融合了用户社会地位信息。

3 融合用户社会地位和矩阵分解的推荐算法

本节详细描述本文提出的融合用户社会地位和矩阵分解的推荐算法。首先描述融合用户社会地位和矩阵分解推荐算法的框架,然后详细介绍此推荐算法的模型和参数学习过程。

3.1 融合用户社会地位和矩阵分解推荐算法框架

融合用户社会地位和矩阵分解的推荐算法框架如图 1 所示:

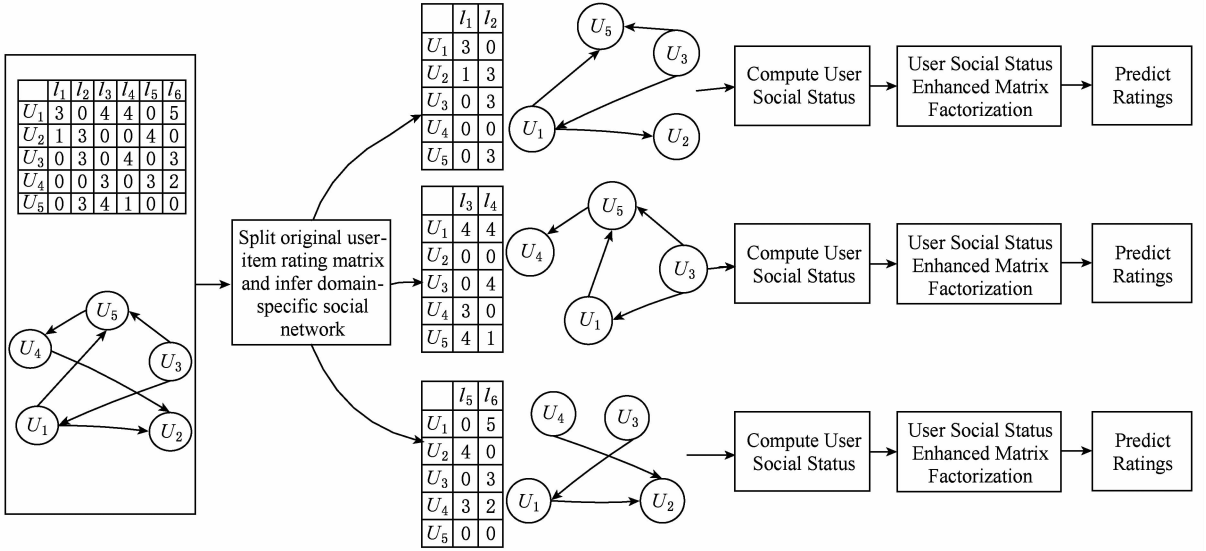


Fig. 1 The framework of our proposed recommendation algorithm

图 1 融合用户社会地位和矩阵分解的推荐算法框架

融合用户社会地位和矩阵分解的推荐算法主要由用户-项目评分数据划分和特定类别用户社交网络的推导、用户社会地位的计算、用户社会地位增强的矩阵分解和评分预测等部分组成。

1) 评分数据划分和特定类别用户社交网络的推导。在此过程中首先按照项目的类别对原始用户-项目评分矩阵 $\mathbf{R}$ 进行划分。如 Epinions 数据集中, 用户对项目的评分表示为三元组的形式: $(uid, pid, categoryid)$, 其中 $categoryid$ 表示被评分项目的类型。原始用户-项目评分矩阵 $\mathbf{R}$ 可以根据 $categoryid$ 分为 $\mathbf{R}^1, \mathbf{R}^2, \dots, \mathbf{R}^l$, l 为用户评分数据集中项目类别的数量。按照项目类别分解完原始用户-项目评分矩阵后, 根据评分和社交关系共现的原则推导特定类别的用户社交网络。用户 u 和用户 v 在类别 c 中存在社交关系的定义如下:

$$T_{uv}^c = \begin{cases} T_{uv}, & T_{uv} \neq 0 \wedge N_u^c > 0 \wedge N_v^c > 0, \\ 0, & \text{others}, \end{cases} \quad (4)$$

其中, N_u^c 和 N_v^c 分别表示用户 u 和用户 v 在 $\mathbf{R}^c$ 中的评分数量, 即用户 u 和用户 v 同时对类别 c 中的项目评过, 并且用户 u 和用户 v 在原始社交网络中存在社交关系, 那么用户 u 和用户 v 在类别 c 中存在社交关系, 并且社交关系值为 T_{uv} , 其他情况为 0。类别 c 中所有用户的社交关系构成类别 c 下的社交关系矩阵 $\mathbf{T}^c$ 。

2) 用户社会地位的计算。在社交网络中, 社交网络的结构一定程度上反映了用户的社会地位。如社会地位高的用户往往可以为其他用户提供有价值的参考信息, 因而拥有大量的 in-link; 社会地位低的用户一般喜欢参考社会地位高的用户的建议, 因而具有较多的 out-link、较少的 in-link。目前有大量的算法根据社交网络的结构计算用户在社交网络中的社会地位, 如 PageRank^[7], HIST^[32]。本文在 $\mathbf{T}^c$ 基础上, 首先采用 PageRank 算法计算用户在每个类别下的 PageRank 值。设 $\mathbf{PR}^c \in \mathbb{R}^{N^c}$ 表示所有用户

在社交关系矩阵 $\mathbf{T}^c$ 中的 PageRank 值, N^c 表示 $\mathbf{T}^c$ 中用户的数量. 用户 u 在类别 c 内的 PageRank 值 $\mathbf{PR}^c(u)$ 可以表示为

$$\mathbf{PR}^c(u) = \alpha \sum_{v \in F^c(u)} \frac{\mathbf{PR}^c(v)}{|\mathbf{F}^c(u)|} + (1 - \alpha) \mathbf{E}^c(u), \quad (5)$$

其中, $F^c(u)$ 为用户 u 在类别 c 中信任朋友的集合; $\mathbf{E}^c$ 为逃逸向量; α 是规范化常数. 然后对 N^c 个用户的 PageRank 值进行排序, 设用户 u 的排名为 $r_u^c \in [1, N^c]$, 则用户的社会地位值 s_u^c 定义为

$$s_u^c = \frac{1}{1 + \lg(r_u^c)}, \quad (6)$$

从式(6)可以看出, 当用户的 PageRank 值在类别 c 中排名第一时, 则 $s_u^c = 1$, 并且 s_u^c 随着用户排名的降低而递减, 也就是用户 PageRank 值排名越靠后, 与之相对应的社会地位值越低.

3) 用户社会地位增强的矩阵分解. 通过 PageRank 算法计算出每个用户在每个类别 c 中的社会地位值后, 针对每个类别, 以用户社会地位值 s_u^c 衡量用户的评分权重, 结合评分数据 $\mathbf{R}^c$ 和社交信任矩阵 $\mathbf{T}^c$, 执行矩阵分解算法, 学习用户和项目隐藏特征矩阵 $\mathbf{P}^c$ 和 $\mathbf{Q}^c$. $\mathbf{P}^c$ 和 $\mathbf{Q}^c$ 详细的学习过程在 3.2 节中描述.

4) 评分预测. 预测缺失项的评分值:

$$\hat{r}_{ui}^c = (\mathbf{p}_u^c)^T \mathbf{q}_i^c. \quad (7)$$

3.2 算法模型及参数学习

在社交网络中, 不同的用户往往具有不同的社会地位, 社会地位低的用户通常会向社会地位高的用户征求意见; 反之, 社会地位高的用户则较少考虑社会低的用户的建议. 换句话说, 社会地位高的用户往往受到其他用户的影响小; 社会地位低的用户则由于缺乏相关领域的知识, 受到其他用户的影响大. 因此, 本文使用用户社会地位衡量用户评分的重要性, 即:

$$\min_{\mathbf{p}^c, \mathbf{q}^c} \frac{1}{2} \sum_{(u,i) \in \Omega^c} s_u^c (r_{ui}^c - (\mathbf{p}_u^c)^T \mathbf{q}_i^c)^2, \quad (8)$$

其中, Ω^c 为 $\mathbf{R}^c$ 中可观测到的(用户, 项目)对. 在式(8)中, 如果用户 u 在领域 c 中的社会地位高, 则要求用户 u 评分更加可靠, 即预测评分与实际评分之间的误差 $r_{ui}^c - (\mathbf{p}_u^c)^T \mathbf{q}_i^c$ 小; 相反, 如果用户 u 在领域 c 中的社会地位低, s_u^c 值小, 则允许预测评分与实际评分之间的误差大.

加上防止过拟合的正则化项后, 目标函数定义为

$$\min_{\mathbf{p}^c, \mathbf{q}^c} \frac{1}{2} \sum_{(u,i) \in \Omega^c} s_u^c (r_{ui}^c - (\mathbf{p}_u^c)^T \mathbf{q}_i^c)^2 + \frac{\lambda_1}{2} \|\mathbf{P}^c\|_F^2 + \frac{\lambda_2}{2} \|\mathbf{Q}^c\|_F^2, \quad (9)$$

式(9)仅从社交网络全局的角度考虑用户社会地位对用户评分的影响, 但是未考虑用户的行为同时受到朋友的影响, 即式(9)未考虑局部的社会影响力.

为了在矩阵分解的过程考虑局部社会影响力对用户行为的影响, 本文在由式(9)定义的目标函数基础上, 加上正则化项:

$$\frac{\lambda_3}{2} \sum_{u=1}^{N^c} \left\| \mathbf{p}_u^c - \sum_{v \in F^c(u)} T_{uv} \mathbf{p}_v^c \right\|_F^2, \quad (10)$$

使得用户的隐藏特征向量依赖于其信任用户的隐藏特征向量. 这种方法对于处理推荐系统的冷启动问题特别有效. 对于新加入系统的用户, 由于其评分数据较少, 矩阵分解的方法不能准确地根据评分数据学习用户的隐藏特征向量. 新用户虽然评分数据较少, 但是可能与其他用户建立了信任关系, 通过式(10)可以使新加入系统的用户的隐藏特征向量与其信任朋友的隐藏特征向量尽可能地靠近, 从而间接地学习新用户的隐藏特征向量.

另外, 不失一般性, 本文通过函数 $f(x) = (x - \min \text{Rating}) / (\max \text{Rating} - \min \text{Rating})$ 将评分值 r_{ui}^c 映射到 $[0, 1]$ 区间, $\min \text{Rating}$ 和 $\max \text{Rating}$ 分别表示最小评分值和最大评分值. 同时, 使用 logistic 函数 $g(x) = 1 / (1 + e^{-x})$ 将预测评分 $(\mathbf{p}_u^c)^T \mathbf{q}_i^c$ 限定在 $[0, 1]$ 范围内. 因此, 本文提出的融合用户社会地位和矩阵分解的推荐算法的目标函数为

$$L^* = \frac{1}{2} \sum_{(u,i) \in \Omega^c} s_u^c (r_{ui}^c - g((\mathbf{p}_u^c)^T \mathbf{q}_i^c))^2 + \frac{\lambda_1}{2} \|\mathbf{P}^c\|_F^2 + \frac{\lambda_2}{2} \|\mathbf{Q}^c\|_F^2 + \frac{\lambda_3}{2} \sum_{u=1}^{N^c} \left\| \mathbf{p}_u^c - \sum_{v \in F^c(u)} T_{uv} \mathbf{p}_v^c \right\|_F^2. \quad (11)$$

采用随机梯度下降的方法求解 L^* 的局部最小解. 目标函数 L^* 关于 $\mathbf{p}_u^c$ 和 $\mathbf{q}_i^c$ 的偏导数为

$$\begin{aligned} \frac{\partial L^*}{\partial \mathbf{p}_u^c} = & -s_u^c (r_{ui}^c - g((\mathbf{p}_u^c)^T \mathbf{q}_i^c)) g'((\mathbf{p}_u^c)^T \mathbf{q}_i^c) \mathbf{q}_i^c + \\ & \lambda_1 \mathbf{p}_u^c + \lambda_3 (\mathbf{p}_u^c - \sum_{v \in F^c(u)} T_{uv} \mathbf{p}_v^c) - \\ & \lambda_3 \sum_{v \in F^c(u)} T_{uv} (\mathbf{p}_v^c - \sum_{w \in F^c(v)} T_{vw} \mathbf{p}_w^c), \end{aligned} \quad (12)$$

$$\frac{\partial L^*}{\partial \mathbf{q}_i^c} = -s_u^c (r_{ui}^c - g((\mathbf{p}_u^c)^T \mathbf{q}_i^c)) g'((\mathbf{p}_u^c)^T \mathbf{q}_i^c) \mathbf{p}_u^c + \lambda_1 \mathbf{q}_i^c, \quad (13)$$

其中, $g'(x) = e^{-x} / (1 + e^{-x})^2$ 为 logistic 函数 $g(x)$ 的导数, $F^c(u)$ 为 T^c 中信任 u 的用户集合.

4 实验与分析

为了验证融合用户社会地位和矩阵分解推荐算法的性能, 本文在真实的数据集上与其他流行的推荐算法进行了对比分析.

4.1 数据集介绍

本文选择 Epinions 数据集验证本文提出推荐算法的性能. Epinions 数据集中含有用户评分、社交关系和项目类别等信息. 在 Epinions 中, 用户可以浏览其他用户对产品的评分和评论信息, 同时用户以评分和评论的方式发表自己的意见. Epinions 中的产品按类别分类, 如电影类、数字产品类和书籍类等. 另外, Epinions 中的每个用户维护一个信任用户列表. Epinions 中信任关系是有向的, 信任值是二值的.

本文使用的 Epinions 数据集是由文献[27]的作者提供的. 此数据集包含 922 267 条评分记录、22 166 个用户和 296 277 个产品、355 813 条信任关系. 用户-项目评分矩阵的稀疏度为 99.986%, 用户评分记录基本符合幂律分布, 如图 2 所示. 从图 2 可以观察, Epinions 数据集中大量用户仅提供较少的评分记录.

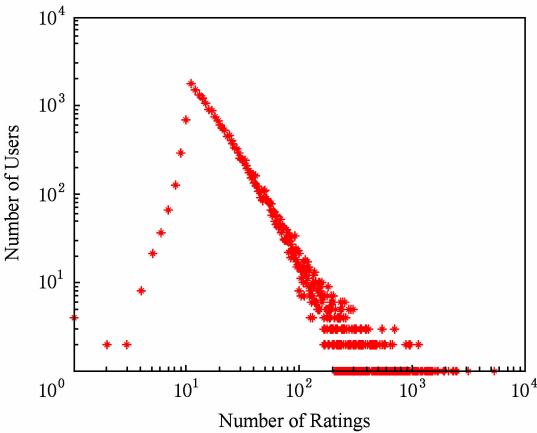


Fig. 2 The power-law distribution of users for Epinions dataset

图 2 Epinions 数据集的用户评分幂律分布

Epinions 中的产品分为 27 个类别, 用户涉及的类别数量分布如图 3 所示.

从图 3 可以观察, 用户平均涉及到的项目类别数较少, 仅 21.7% 的用户涉及到的类别数超过 10,

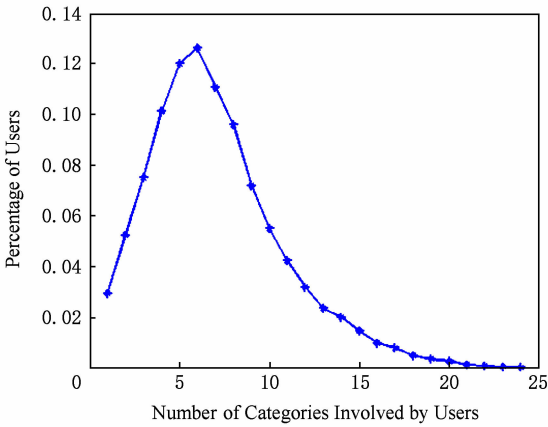


Fig. 3 Distribution of users for different number of categories for Epinions dataset

图 3 Epinions 数据集的用户涉及项目类别分布

也就是说, 用户并不是对所有的类别都感兴趣, 因而不能在所有类别上影响朋友的评分行为. 这也表明了本文提出的推荐算法对用户评分数据和社交关系按项目类别分类处理的必要性. 本文选取“Movie”, “Home Garden”, “Education”这 3 个类别的子数据集来验证本文提出的推荐算法. “Movie”, “Home Garden”, “Education”分别代表大、中、小规模的数据集. 经过评分数据划分和特定类别的用户社交网络的推导后, 3 个类别数据集统计信息如表 1 所示:

Table 1 Statistics of Three Data Sets

表 1 数据集统计信息

Dataset	N	M	$ R $	Density	$\bar{r}$	$ T^c $	$\bar{t}$
Movie	14 180	28 616	167 261	4.1E-4	11.8	153 773	10.84
Home Garden	8 074	20 159	45 113	2.7E-4	5.58	62 805	7.77
Education	3 114	1 108	5 475	15.9E-4	1.76	7 711	2.48

其中, $\bar{r}$ 表示用户平均评分数量, $\bar{t}$ 表示用户平均信任关系数.

4.2 度量指标

目前, 在推荐系统研究领域有很多不同的度量指标被用来度量推荐算法的性能. 例如平均绝对值误差 (mean absolute error, MAE)、均方根误差 (root mean squared error, RMSE) 和归一化的平均绝对值误差 (normalized mean absolute error, NMAE) 等. 本文采用 RMSE 来评价推荐算法的性能. RMSE 的定义为

$$RMSE = \sqrt{\frac{\sum_{(u,i) \in R_{test}} |r_{ui} - \hat{r}_{ui}|^2}{|R_{test}|}}, \tag{14}$$

其中, r_{ui} 和 $\hat{r}_{ui}$ 分别表示实际的评分值和推荐系统预测的评分值; $|R_{\text{test}}|$ 表示测试数据集中的记录条数. RMSE 值越小, 推荐算法的推荐性能越好.

4.3 实验设置

为了验证本文提出推荐算法的有效性, 我们选取如下的推荐算法作为对比方法.

1) PMF. PMF 是由 Mnih 和 Salakhutdinov^[25] 提出, 可以看作是 SVD 模型的概率扩展, 它仅利用用户-项目评分矩阵来生成推荐项.

2) SoRec. SoRec^[2] 同时分解用户评分矩阵和用户信任关系矩阵, 并以用户评分矩阵和用户信任关系矩阵共享用户隐藏特征矩阵的方式来融合评分信息和社交网络信息. 不同于 PMF, SoRec 是一个基于社交网络的推荐算法, 它同时利用用户评分信息和社交网络信息来为用户提供推荐服务.

3) RSTE. RSTE^[3] 在产生推荐项过程中考虑了用户自身偏好和朋友偏好, 并通过集成参数来融合用户自身偏好和朋友偏好. RSTE 是一种基于社交网络的推荐算法.

4) SocialMF. SocialMF 是由 Jamali 等人^[4] 提出的一种基于社交网络的推荐算法. SocialMF 在 PMF 中加入信任的传递机制来提高推荐算法的准确性.

5) TrustMF. TrustMF^[5] 认为用户的行为既受到其他用户评分和评论的影响, 同时他的评分和评论也会影响其他用户. TrustMF 在用户信任关系矩阵上执行矩阵分解, 将用户映射到 2 个不同隐藏特征空间: 信任特征空间和被信任特征空间.

我们随机从用户-项目评分矩阵中抽取 80% 的数据作为训练集, 剩余 20% 的数据作为测试数据. 这种随机抽取独立地执行 5 次, 以 5 次运行结果的平均值作为最后的运行结果. 为了公平对比, 我们参对比算法的相应文献或者实验结果设置不同算法的参数, 在这些参数设置下, 各对比算法取得最优性能. 在 PMF 中, $\lambda_U = \lambda_V = 0.03$; 在 SoRec 中, $\lambda_U = \lambda_V = \lambda_Z = 0.001$, $\lambda_C = 1$; 在 RSTE 中, $\lambda_U = \lambda_V = 0.001$, $\alpha = 0.4$; 在 SocialMF 中, $\lambda_U = \lambda_V = 0.001$, $\lambda_T = 1$; 在 TrustMF 中, $\lambda = 0.001$, $\lambda_T = 1$; 对于本文提出的方法, $\lambda_1 = \lambda_2 = 0.001$, λ_3 在 3 个数据集上分别设置为 0.5, 0.1, 0.1. 另外, 我们设置 PMF 的学习率 $\eta = 0.05$, 本文提出的推荐算法的学习率 $\eta = 0.03$, 其他基于社交网络推荐算法的学习率 $\eta = 0.001$.

另外, 对于 SoRec, RSTE, SocialMF, TrustMF, 我们使用 Epinions 中所有的社交关系来训练预测模型, 对于本文提出的方法, 使用 3.1 节中推导的特定领域的社交关系来训练预测模型.

实验运行环境为: 4 核 Intel® Xeon® E3-1225 CPU, 3.2 GHz 主频, 8 GB 内存, Window7 操作系统和 J2SE1.7.

4.4 性能对比

在隐藏特征维度值 $K=10$ 和 $K=20$ 情况下, 所有对比算法在 3 个数据集上的实验结果如表 2~4 所示.

Table 2 Performance Comparison on Education
表 2 在 Education 数据集上的性能对比

Recommendation Algorithm	RMSE	
	K=10	K=20
PMF	3.0621	3.0838
SoRec	1.2790	1.5268
RSTE	1.2343	1.4515
SocialMF	1.2674	1.5438
TrustMF	1.3213	1.5456
Our Method	1.2126	1.3176

Table 3 Performance Comparison on Home Garden
表 3 在 Home Garden 数据集上的性能对比

Recommendation Algorithm	RMSE	
	K=10	K=20
PMF	2.5542	2.5737
SoRec	1.1764	1.3328
RSTE	1.1600	1.2507
SocialMF	1.1448	1.3292
TrustMF	1.1845	1.3550
Our Method	1.1270	1.2337

Table 4 Performance Comparison on Movie
表 4 在 Movie 数据集上的性能对比

Recommendation Algorithm	RMSE	
	K=10	K=20
PMF	1.5041	1.5117
SoRec	1.1540	1.3341
RSTE	1.2258	1.3232
SocialMF	1.1359	1.3034
TrustMF	1.1806	1.3787
Our Method	1.1354	1.2865

从表 2~4 可以发现:

1) 基于社交网络的推荐算法的性能优于仅利用用户评分信息的推荐算法,再次验证了社交网络信息有助于改进推荐算法的性能。

2) 在 3 个数据集上,本文提出的融合用户社会地位的矩阵分解推荐算法的推荐准确度高于其他对比算法,说明了本文提出算法的有效性。在 Education, Home Garden, Movie 数据集上, $K=20$ 时,与 PMF, SoRec, RSTE, SocialMF, TrustMF 中的最优结果对比,本文提出算法的改进幅度分别为 9.2%, 1.4%, 1.3%。

3) 在 3 个数据集上,随着 K 的增加,所有对比算法的性能下降,说明增加隐藏特征向量的维数不能有效提高矩阵分解算法的准确性,这也验证了矩阵分解算法的假设:仅有少量的隐藏因子影响用户的偏好和刻画项目的特征。

4) 在 $K=10$ 时,对比 PMF, SoRec, SocialMF, RSTE, TrustMF, RSTE 在 Education 数据集上取得最优性能, SocialMF 在 Home Garden 和 Movie 数据集上取得最优性能。这可能是由于 Movie, Home Garden 中的用户平均信任关系系数大于 Education 中的用户平均信任关系系数,即在用户平均信任关系系数较大时 SocialMF 的推荐准确性较高,在用户平均信任关系系数较小时 RSTE 的推荐准确性高。

4.5 参数 λ_3 的影响

在本文提出的推荐算法中,参数 λ_3 是一个影响推荐性能的重要参数。较大的 λ_3 值意味着我们在矩阵分解模型中更加依赖社交网络关系。在极端的情况下,如 $\lambda_3 = \infty$, 本文提出的推荐算法将完全依赖社交网络关系来学习用户和项目隐藏特征向量,而忽视用户的评分信息;相反,较小的 λ_3 值意味着我们赋予用户评分信息更多的权重,如 $\lambda_3 = 0$ 表示推荐算法完全依赖用户评分信息来学习用户和项目隐藏特征向量。在本节中,我们设置 λ_3 为 0, 0.01, 0.1, 0.5, 1, 5 来检验 λ_3 对本文提出推荐算法的影响。在本组实验中,设置隐藏特征向量的维数 $K=10$ 。实验结果如图 4 所示。

从图 4 可以看出,参数 λ_3 确实影响本文提出推荐算法的性能。其次,在 3 个数据集上, $RMSE$ 的值呈现类似的变化趋势:随着 λ_3 的增加, $RMSE$ 先降低,推荐准确性提高,到达最优值后, $RMSE$ 随 λ_3 的增加而增加,推荐准确性降低。最后,在 3 个数据集上,本文提出的推荐算法并没有在同一个 λ_3 值下获得最低 $RMSE$,说明数据集不同则本文提出推荐算

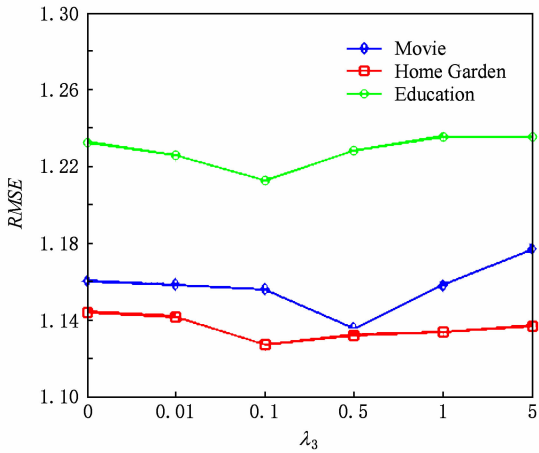


Fig. 4 The impact of λ_3

图 4 λ_3 的影响

法对社交网络信息的依赖程度不同。在 Movie, Home Garden 和 Education 上, $RMSE$ 分别在 $\lambda_3 = 0.5, 0.1, 0.1$ 时取得最优性能,表示本文提出的推荐算法在 Movie 数据集上更加依赖社交网络信息。这是由于在 Movie 类别中,用户平均信任关系系数比 Home Garden 和 Education 中的用户平均信任关系系数相对大。

4.6 参数 K 的影响

隐藏特征向量的维数 K 是影响本文提出融合用户社会地位和矩阵分解推荐算法性能的另一个重要参数。在本节中,我们变化 K 的值,从 5~50,递增步长为 5,观察 $RMSE$ 在 3 个数据集上的变化情况。参数 λ_3 分别设置为 0.5, 0.1, 0.1。实验结果如图 5 所示:

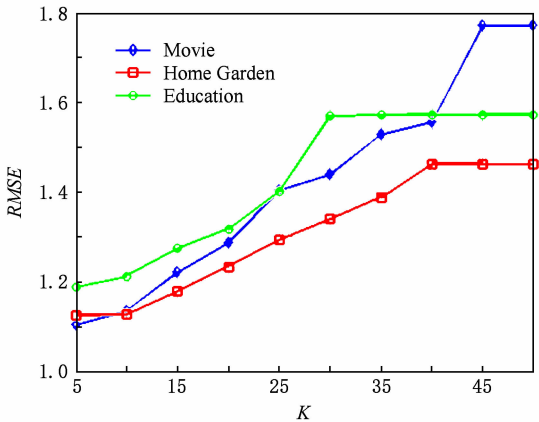


Fig. 5 The impact of K

图 5 K 的影响

从图 5 可以观察到,随着 K 的递增, $RMSE$ 的值先增加,后趋于稳定,整体推荐性能下降。这个观察结果表明:虽然 K 值增大使得矩阵分解模型可以

表示较多的隐藏特征,但是同时会引入一些噪音,降低矩阵分解推荐算法的准确性.这个观察结果再次验证了矩阵分解模型的基本假设:仅有少量的隐藏因子影响用户的偏好和刻画项目的特征.

4.7 模型训练时间对比

在本节中,我们对比不同推荐算法的模型训练时间,以检验本文提出推荐算法的效率.在 Movie, Home Garden, Education 数据集上,参数 λ_3 分别设置为 0.5, 0.1, 0.1, $K=10$. 其他对比推荐算法的参数设置同 4.3 节. 不同推荐算法在 3 个数据集上的模型训练时间如表 5 所示:

Table 5 The Runtime of Model Training (min:sec)			
表 5 模型训练时间(分:秒)			
Recommendation Algorithm	Movie	Home Garden	Education
PMF	0:0.74	0:0.64	0:0.13
SoRec	1:24	0:19	0:12
RSTE	55:44	23:22	2:06
SocialMF	27:36	25:55	25:39
TrustMF	01:01	00:39	00:25
Our Method	05:20	01:56	00:04

从表 5 可以观察到,PMF 在 3 个数据集上的模型训练时间最少.这是由于 PMF 在学习用户和项目隐藏特征向量时,未考虑社交网络关系对用户隐藏特征向量的影响.在基于社交网络的推荐算法中, SoRec 和 TrustMF 在效率方面胜过 SocialMF, RSTE 和本文提出的推荐算法.就模型训练时间而言,这说明在用户评分矩阵和用户社交关系矩阵中共享用户隐藏特征向量的推荐模型(即 SoRec 和 TrustMF)优于通过社交关系约束用户隐藏特征向量的模型(即 SocialMF 和 RSTE).本文提出的推荐算法在 Movie 和 Home Garden 数据集上,效率高于 SocialMF 和 RSTE 算法,在 Education 数据集上效率优于 SoRec, RSTE, SocialMF, TrustMF.

综上所述,从实验结果看出:本文提出的融合用户社会地位和矩阵分解的推荐算法在 3 个数据集上的推荐准确性优于对比算法,在模型学习的效率方面,与目前主流的基于社交网络推荐算法相当.

5 结 论

基于社交网络的推荐算法一般假设社交网络中用户的偏好受到朋友的影响,并且朋友之间具有类

似的偏好.然而,在现实生活中,用户在不同的领域往往信任不同的朋友;另外,用户在不同领域具有不同的社会地位,因此,用户在不同的领域内,受朋友的影响程度各不相同.为了解决传统基于社交网络推荐算法中存在的上述问题,本文提出了融合用户社会地位和矩阵分解的推荐算法.首先利用用户评分和社交信任关系共现的原则来推导特定领域内的用户社交网络结构;然后利用 PageRank 算法计算每个特定领域内用户的社会地位;最后以用户社会地位来约束矩阵分解过程.在真实数据集上的实验结果表明,本文提出融合用户地位信息的矩阵分解推荐算法的性能优于传统的基于社交网络推荐算法.

本文仅考虑以社交网络结构信息来衡量用户的社会地位,而用户的评分信息以及用户的评论信息也在一定程度上反映了用户的社会地位.结合社交网络结构信息、评分信息和评论信息来衡量用户的社会地位是本文未来的研究方向;另外,现实的生活场景中,用户往往只提供隐式的反馈信息,如购买记录、点击记录等,即此时我们面对的是 One-Class Collaborative Filtering^[33-35]问题.在 One-Class Collaborative Filtering 推荐场景中,如何学习用户的社会地位值,并将其用于改进推荐算法的性能也是本文未来的研究方向.最后,深度学习技术已经在自然语言处理、计算机视觉等领域表现出强大的潜力,结合深度学习技术来改进现有的推荐算法将是推荐系统研究领域一个非常有趣的研究方向.

参 考 文 献

[1] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions [J]. IEEE Trans on Knowledge and Data Engineering, 2005, 17(6): 734-749

[2] Ma Hao, Yang Haixuan, Lyu M R, et al. SoRec: Social recommendation using probabilistic matrix factorization [C] //Proc of the 17th ACM Conf on Information and Knowledge Management (CIKM'08). New York: ACM, 2008: 931-940

[3] Ma Hao, King I, Lyu M R. Learning to recommend with social trust ensemble [C] //Proc of the 32nd Int ACM SIGIR Conf on Research and Development in Information Retrieval (SIGIR'09). New York: ACM, 2009: 203-210

[4] Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks [C] //Proc of the 4th ACM Conf on Recommender Systems (RecSys'10). New York: ACM, 2010: 135-142

- [5] Yang Bo, Lei Yu, Liu Dayou, et al. Social collaborative filtering by trust [C] //Proc of the 23rd Int Joint Conf on Artificial Intelligence (IJCAI'13). Menlo Park, CA: AAAI, 2013: 2747-2753
 - [6] Guo Guibing, Zhang Jie, Yorke-Smith N. TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings [C] //Proc of the 29th AAAI Conf on Artificial Intelligence (AAAI'15). Menlo Park, CA: AAAI, 2015: 123-129
 - [7] Page L, Brin S, Motwani R, et al. The PageRank citation ranking: Bringing order to the Web [R]. Stanford InfoLab. 1999 [2017-02-05]. <http://ilpubs.stanford.edu/8090/422/1/1999-66.pdf>
 - [8] Resnick P, Iacovou N, Suchak M, et al. GroupLens: An open architecture for collaborative filtering of netnews [C] //Proc of the 1994 ACM Conf on Computer Supported Cooperative Work. New York: ACM, 1994: 175-186
 - [9] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms [C] //Proc of the 10th Int Conf on World Wide Web (WWW'01). New York: ACM, 2001: 285-295
 - [10] Linden G, Smith B, York J. Amazon. com recommendations: Item-to-item collaborative filtering [J]. Internet Computing, 2003, 7(1): 76-80
 - [11] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering [C] //Proc of the 14th Conf on Uncertainty in Artificial Intelligence (UAI'98). San Francisco, CA: Morgan Kaufmann, 1998: 43-52
 - [12] Ungar L H, Foster D P. Clustering methods for collaborative filtering [C] //Proc of the AAAI Workshop on Recommendation Systems. Menlo Park, CA: AAAI, 1998, 1: 114-129
 - [13] Xue GuiRong, Lin Chenxi, Yang Qiang, et al. Scalable collaborative filtering using cluster-based smoothing [C] //Proc of the 28th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval (SIGIR'05). New York: ACM, 2005: 114-121
 - [14] Yu Yonghong, Wang Can, Gao Yang, et al. A coupled clustering approach for items recommendation [C] //Proc of the 17th Pacific-Asia Conf on Knowledge Discovery and Data Mining (PAKDD'13). Berlin: Springer, 2013: 365-376
 - [15] Hofmann T. Latent semantic models for collaborative filtering [J]. ACM Trans on Information Systems, 2004, 22(1): 89-115
 - [16] Hofmann T. Collaborative filtering via Gaussian probabilistic latent semantic analysis [C] //Proc of the 26th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval (SIGIR'03). New York: ACM, 2003: 259-266
 - [17] Sarwar B, Karypis G, Konstan J, et al. Analysis of recommendation algorithms for e-commerce [C] //Proc of the 2nd ACM Conf on Electronic Commerce. New York: ACM, 2000: 158-167
 - [18] Wang Hao, Wang Naiyan, Yeung Dit-Yan. Collaborative deep learning for recommender systems [C] //Proc of the 21st ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining (KDD'15). New York: ACM, 2015: 1235-1244
 - [19] Li Sheng, Kawale J, Fu Yun. Deep collaborative filtering via marginalized denoising auto-encoder [C] //Proc of the 24th ACM Int on Conf on Information and Knowledge Management (CIKM'15). New York: ACM, 2015: 811-820
 - [20] Liang Huizhi, Baldwin T. A probabilistic rating auto-encoder for personalized recommender systems [C] //Proc of the 24th ACM Int Conf on Information and Knowledge Management (CIKM'15). New York: ACM, 2015: 1863-1866
 - [21] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems [J]. Computer, 2009, 42(8): 30-37
 - [22] Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model [C] //Proc of the 14th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining (KDD'08). New York: ACM, 2008: 426-434
 - [23] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization [J]. Nature, 1999, 401(6755): 788-791
 - [24] Weimer M, Karatzoglou A, Le Q V, et al. Maximum margin matrix factorization for collaborative ranking [C] //Proc of the 21st Annual Conf on Neural Information Processing Systems (NIPS'07). New York: Curran Associates, 2007: 1-8
 - [25] Mnih A, Salakhutdinov R. Probabilistic matrix factorization [C] //Proc of the 21st Annual Conf on Neural Information Processing Systems (NIPS'07). New York: Curran Associates, 2007: 1257-1264
 - [26] Yu Kai, Zhu Shenghuo, Lafferty J, et al. Fast nonparametric matrix factorization for large-scale collaborative filtering [C] //Proc of the 32nd Int ACM SIGIR Conf on Research and Development in Information Retrieval (SIGIR'09). New York: ACM, 2009: 211-218
 - [27] McPherson M, Smith-Lovin L, Cook J M. Birds of a feather: Homophily in social networks [J]. Annual Review of Sociology, 2001, 27: 415-444
 - [28] Marsden P V, Friedkin N E. Network studies of social influence [J]. Sociological Methods & Research, 1993, 22(1): 127-151
 - [29] Guo Hongyi, Liu Gongshen, Su Bo, et al. Collaborative filtering recommendation algorithm combining community structure and interest clusters [J]. Journal of Computer Research and Development, 2016, 53(8): 1664-1672 (in Chinese)
- (郭弘毅, 刘功申, 苏波, 等. 融合社区结构和兴趣聚类的协同过滤推荐算法 [J]. 计算机研究与发展, 2016, 53(8): 1664-1672)

[30] Yang Xiwang, Steck H, Liu Yong. Circle-based recommendation in online social networks [C] //Proc of the 18th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining (KDD'12). New York: ACM, 2012: 1267-1275

[31] Tang Jiliang, Hu Xia, Gao Huiji, et al. Exploiting local and global social context for recommendation [C] //Proc of the 23rd Int Joint Conf on Artificial Intelligence (IJCAI'13). Menlo Park, CA: AAAI, 2013: 2712-2718

[32] Kleinberg J M. Authoritative sources in a hyperlinked environment [J]. Journal of the ACM, 1999, 46(5): 604-632

[33] Pan Rong, Zhou Yunhong, Cao Bin, et al. One-class collaborative filtering [C] //Proc of the 8th IEEE Int Conf on Data Mining (ICDM'08). Los Alamitos, CA: IEEE Computer Society, 2008: 502-511

[34] Hu Yifan, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets [C] //Proc of the 8th IEEE Int Conf on Data Mining (ICDM'08). Los Alamitos, CA: IEEE Computer Society, 2008: 263-272

[35] Zhao Tong, McAuley J, King I. Leveraging social connections to improve personalized ranking for collaborative filtering [C] //Proc of the 23rd ACM Conf on Information and Knowledge Management (CIKM'14). New York: ACM, 2014: 261-270



Yu Yonghong, born in 1978. PhD and associate professor. His main research interests include recommendation algorithm and social network analysis.



Gao Yang, born in 1972. Professor and PhD supervisor. His main research interests include reinforcement learning and machine learning (gaoy@nju.edu.cn).



Wang Hao, born in 1983. PhD and assistant researcher. His main research interests include recommendation algorithm and reinforcement learning (wanghao.hku@gmail.com).



Sun Shuanzhu, born in 1973. Senior engineer. His main research interests include recommender systems and energy conservation (15905166613@139.com).