推荐系统中谁可以协同新用户?

张 莉 余 磊
(对外经济贸易大学信息学院 北京 100029)

摘 要 协同过滤作为被成功应用于推荐系统的技术之一,得到了各领域学者的关注。然而随着网络平台新用户和项目的不断增加,协同推荐面临严重的"冷启动"问题的挑战。首先基于用户流行度和长尾分布建立用户推荐能力的度量方法,然后利用用户推荐能力筛选出一个用于推荐的全局核心用户子集,来解决推荐系统的"冷启动"问题。实验结果显示,将构建的全局核心用户集合用于协同推荐,在不降低推荐效果的基础上,可显著降低寻找相似用户的时间复杂度,因而可以将其用于解决推荐实时性问题。

关键词 协同过滤,核心用户,长尾分布,用户流行度
中图法分类号 G203 **文献标识码** A

Who Can Collaborate New Users in Recommendation System?

ZHANG Li YU Lei
(School of Information Technology & Management, University of International Business and Economics, Beijing 100029, China)

Abstract As a successful technology used in the recommender system, collaborative filtering has been widly concerned by scholars in various fields. However, with the increasing of new users and items, collaborative recommendation is facing serious challenge of "cold start". This study measured the recommending ability of user based on popularity and long-tailed distribution, and then constructed a global core user set for recommadition using user popularity, which can be used to solve "cold start" problems in recommendation systems. In additional, experimental results show that the core use set used for collaborative recommending can reduce complexity of looking for similar users without lowing the recommendation performance. So it also can be used to improve real-time recommendation.

Keywords Collaborative filtering, Core users, Long-tailed distribution, User popularity

1 引言

随着互联网技术的发展,信息呈爆炸式增长,推荐系统作为解决"信息过载"的有效技术应运而生。协同过滤作为目前推荐系统被成功应用的推荐技术,其基本思想是相似用户具有相似的兴趣爱好。但是由于缺少相关消费记录,无法计算新用户和其它用户的相似性,使得传统的系统过滤算法不能实现对新用户的推荐。另一方面,随着网络平台用户和项目的不断增加,用户行为呈现长尾现象,相似用户的搜索面临严峻的计算复杂性挑战[1]。为此,国内外学者提出了许多新的或改进算法试图解决上面的"冷启动"和推荐实时性问题,如基于模型和内容的推荐技术、基于聚类的协同过滤等[2]。但很少有研究关注不同用户子集对推荐性能的影响。本文主要研究不同用户子集对协同推荐算法性能的影响,试图在不降低算法推荐性能的基础上,构造一个个用于推荐的核心用户子集,以缩小协同过滤算法中相似用户的搜索范围,降低算法的计算复杂度,解决"冷启动"问题。

本文的主要贡献是基于处于长尾分布不同位置的用户对协同过滤算法的性能影响进行研究,构造了一种用户推荐能力度量方法,并基于此选择出全局核心用户子集用于推荐。

2 相关研究

冷启动是协同过滤算法中被研究者们广泛关注的问题,也被称为新用户问题和新项目问题。传统的协同过滤算法是根据用户的评分信息产生推荐的,而新用户或新项目没有评分信息,所以传统的协同过滤算法无法为新用户和新项目找到相似邻居产生推荐,从而产生"冷启动"问题。国内外学者针对"冷启动"问题进行了大量研究,这些研究主要有两个分支。一是通过特定的方法填充新用户的评分矩阵,然后再用传统的协同过滤方法进行推荐。均值法、众数法、信息熵法经常被用来填充新用户评分矩阵[3],这些方法是以牺牲用户的个性化需求为代价的,所以违背了个性化信息服务的宗旨。也有部分学者基于用户和商品的附加属性,如人口统计属性、商品的内容属性,基于回归分析预测新用户对商品的评分,然后根据预测评分进行推荐[4]。二是利用用户和商品的内容信息扩充评分矩阵,因为增加了相应的内容信息使得用户(项目)的相似性计算成为可能,因而这一类研究主要是改进协同过滤算法中的相似性计算方法[5]。随着社会网络的发展,一

本文受国家社科基金项目(13BTQ027)资助。

张 莉 博士,副教授,主要研究方向为智能信息技术、电子商务、社会网络分析等,E-mail:tasummer@sina.com;余 磊(1988—),男,硕士生,主要研究方向为社会网络分析。

部分学者将社会网络的朋友关系、会员关系或社会网络标签数据与用户评分数据融合，改进传统 CF 算法，并利用社会网络的朋友属性解决 CF 中的冷启动问题[6,7]；周涛等将物质扩散与热传导理论引入到个性化推荐算法，提出了基于用户-产品二部图的资源分配模型，并且认为在协同推荐算法中可以将用户拥有的资源作为其权重，并利用权重调整用户间的相似性[8]；Zhang Li 等利用社会网络用户结点的度表示用户的推荐能力[9]，根据用户结点的度重新排列相似用户顺序。

上述的研究都试图设计一种新的或改进的协同推荐方法解决"冷启动"问题，一般都是基于用户全集计算相似性或推荐能力，随着用户数量级的递增，算法计算复杂性呈数量级递增。尽管也有部分学者利用分类或聚类技术将目标用户划分到某一用户分组，只是在同一个用户分组中查找其最相似用户，降低了算法的时间复杂性，但这些研究没有充分利用用户的推荐能力[2]。另一方面，已有研究证实在用户-项目二部图中，若用评价的项目数作为用户结点的度，则用户结点度符合幂律分布（长尾分部）。但根据协同过滤算法中用户相似性度量方法（如 Pearson 系数、余玄相似）可知：若一个用户结点的度越大，则其相似用户越多，其越容易进入目标用户的相似用户集合，即度较大的用户（位于长尾分布的首部）往往会被用来进行协同目标用户；反之，若用户结点的度较少，其相似用户也很少，利用这种稀疏关系很难发现用户间的相似性，即很少被用来推荐。所以，若一个用户的度处于中等水平，则对于描述用户的推荐能力更有利。上述研究忽略了处于长尾分布中间用户的作用。

受文献[10]提出的全局最近邻思想启发，本研究在分析处于长尾分布不同位置用户作用的基础上，试图构造一个全局核心用户子集。该集合只包含部分用户，但基于该集合的协同推荐算法的性能基本保持不变，为解决在线推荐系统的推荐实时性和冷启动问题提供一种新思路。

3 用户推荐能力度量方法

Morid M. A. 等对关键用户进行了建模[11]。本文选取与用户影响力有关的用户流行度进行研究。

(1)评分项目数

I_{u_i} 表示用户 u_i 评价项目的集合，则评价项目数 $\| I_{u_i} \|$ 可以表示用户的度，度越大的用户将会有更大的概率与其他用户评价共同产品。文献[9,11]也分析了结点的度对推荐性能的影响，认为结点的度越大，用户的推荐能力越强。

(2)用户流行度

参考文献[11]的描述，通过评估用户所评项目的平均评价频度，衡量一个用户是偏好流行（被评价次数多）项目还是非流行（很少被评价）项目，即将用户所评分项目的被评价频率定义为用户流行度，如式(1)所示。

$$AvgFreq(u_i)=\frac{\sum_{a_j \in I_{u_i}} Freq(a_j)}{\| I_{u_i} \|} \tag{1}$$

其中，$Freq(a_j)$是项目 a_j 的流行度，即被评分的次数。文献[11]认为用户流行度与用户推荐能力相关，其研究结果显示用户流行度越低，其推荐能力越强。

(3)协同过滤中的长尾问题

由式(1)可知，用户的度与流行度负相关，所以可以认为，用户的度越大，其流行度的值越低。图 1 显示了美国 GroupLens实验室提供的 MovieLens 数据集（100kB）的用户的度和流行度之间的关系，以评分项目数 $\| I_{u_i} \|$ 为纵轴，以评分项目流行度 $AvgFreq(u_i)$为横轴。由图 1 可知，用户的流行度和度呈现明显的长尾分布，并且二者负相关。

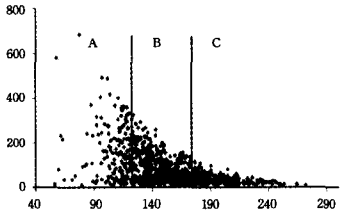

图 1 MovieLens 数据集的用户度与流行度的分布特征

根据已有用户相似性度量方法可知：结点度大的用户处于长尾分布的首部，更容易被用来推荐；反之，若用户结点的度较少，其相似用户也很少，利用这种稀疏关系很难发现用户间的相似性，即很少被用来推荐。但对于长尾分布中间部分用户的推荐能力，相关的研究很少。

(4)用户推荐能力度量

首先，为了利用处于长尾分布中间部分的用户信息，用式(2)表达用户的推荐能力。

$$Qual(u_i)=\frac{d_s}{d_s+|d_s-\log(d_{u_i})|} \tag{2}$$

其中，d_{u_i} 可以表示用户的度或流行度，$d_s=\max_{u_i \in U}(d_{u_i})/2$ 表示用户度或流行度的中等水平。由此公式可以知道，具有较高或较低度的用户，$Qual(u_i)$值较小；类似地，具有较低或较高流行度的用户，$Qual(u_i)$值也会比较小，而有中等水平度或者流行度的用户具有较高 $Qual(u_i)$值。

然后，根据式(2)计算每个用户的推荐能力，并降序排列，选择前 30%～40%的用户组成全局核心用户子集，利用该子集完成传统的基于用户的协同过滤算法。寻找目标用户的 K 个最相似用户是传统的基于用户的协同过滤的核心步骤之一，在此过程中需要计算目标用户与其他用户的相似度，当用户量很大时，其计算复杂性也很大。而核心用户子集只包含了 30%～40%的用户，所以可以大大降低相似度的计算复杂性，从而改善推荐实时性问题。若目标用户是新用户，则选择该子集用户共同评价的前 N 个项目进行推荐，从而解决推荐系统的"冷启动"问题。

4 实验

实验数据采用美国 GroupLens 实验室提供的 MovieLens 数据集（100kB）。数据集中包含了 943 个用户对 1682 部电影的 100000 个评分。数据集被划分成训练集和测试集。训练数据集对应的节点度和用户流行度分布如图 1 所示。从图 1 可以看出节点度和用户流行度符合长尾分布，这有利于全局核心用户子集的构造。

4.1 评价标准

算法性能用推荐准确性和多样性进行度量。系统推荐 N 个项目给用户 u_i，其中用户感兴趣的有 N_i 个项目。

(1)准确性

准确性表示用户对推荐系统所推荐的产品列表感兴趣的比率。对于测试集中的目标用户 u_i，其推荐准确率计算如式(3)所示。实验中，N_i 值等于同时出现在算法推荐列表及用户 u_i 评价列表的产品数目。用测试集中所有用户的推荐准

确率平均值作为算法的推荐准确性，如式(3)所示。

$$Precision(N)=\frac{1}{M}\sum_{u_i}\frac{N_i}{N} \tag{3}$$

其中，M 表示测试目标用户数量。

(2)多样性

多样性表示算法推荐新项目的能力。本文采用文献[12]的多样性的度量方法，如式(4)所示。

$$Diversity=\sum_{i=1}^{M}\sum_{j\neq i}(1-\frac{N_{ij}}{N}) \tag{4}$$

其中，N_{ij} 表示用户 u_i，u_j 推荐列表中相同的项目数。

4.2 实验步骤及结果分析

实验的主要目的是验证本文提出的用户推荐能力度量方法对协同过滤算法性能的影响。实验时选择用户流行度计算用户推荐能力，并按照用户流行度大小划分成 3 个大小相等的集合 A、B、C，如图 1 所示。实验中选取最近邻用户数 K=20、30、40、50、60，采用余弦相似性计算用户间的相似性，推荐前 30 个项目。

首先验证处于长尾分布不同位置的用户对协同推荐效果的作用，即采用图 1 所示的集合 A、B、C 完成传统基于用户的协同过滤算法，分别标识为 Aucf、Bucf、Cucf，结果如图 2、图 3 所示。

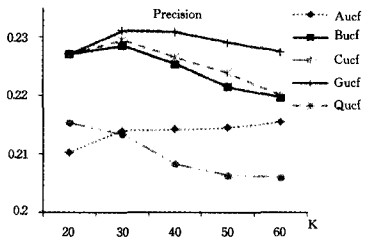

图 2　推荐准确性对比分析

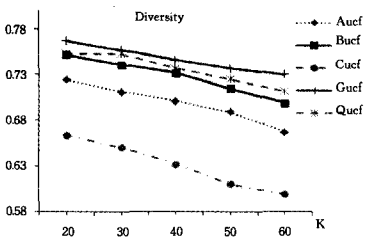

图 3　推荐多样性对比分析

由图 2、图 3 可知，处于长尾分布不同位置上的用户对协同过滤算法的性能影响不同，使用中间部分用户(集合 B)可以得到最好的推荐性能，而处于尾部的用户推荐性能最差。

其次，验证本文提出的用户推荐能力度量方法的性能。利用式(2)计算用户的推荐能力并降序排列，选取前 40%的用户组成核心用户子集，并基于此子集进行协同推荐，用 Qucf 标识，与基于全部用户的协同过滤算法(用 Gucf 标识)的性能对比如图 2、图 3 所示。由图 2、图 3 可知：Qucf 在推荐准确性方面与 Gucf 具有相似的变化趋势，比 Gucf 的推荐性能略差，但 Qucf 只使用了 40%的用户，相似性的计算复杂度远低于 Gucf。

结束语　随着网络技术的发展，推荐系统得到了计算机科学、社会网络、电子商务等领域学者的关注，出现了许多性能较好的推荐算法，但很少有研究关注处于长尾分布不同位置上的用户集合对推荐算法性能的影响。本文研究发现处于长尾分布不同位置的用户子集对协同过滤算法性能的影响不同，并提出了一种新的用户推荐能力的度量方法。实验结果表明，在极大地降低协同过滤算法计算复杂性的同时，本文算法尽可能地保持推荐系统的准确率和多样性。该算法可应用在用户量很大的实时推荐环境，也可以利用构造的全局核心用户信息直接对平台新用户推荐，从而解决推荐的“冷启动”问题。因而，下一步工作考虑采用实际中电子商务平台数据验证本文算法的性能，以及基于 A、B 两个集合的其他用户子集构造算法。

参 考 文 献

[1] Su Xiao-yuan, Taghi M K. A Survey of Collaborative Filtering Techniques[J]. Advances in Artificial Intelligence, 2009(1): 1-19

[2] Fidel C, V'ictor C, Diego F, et al. Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-Performance Recommender Systems[J]. ACM Transactions on the Web, 2011, 5(1): 2-33

[3] 孙冬婷，何涛，张福海. 推荐系统中的冷启动问题研究综述[J]. 计算机与现代化，2012(5)：59-63

[4] Park S T, Chu W. Pairwise preference regression for cold-start recommendation[C]// Proceedings of the Third ACM Conference on Recommender Systems. ACM, 2009: 21-28

[5] Qiu T, Chen G, Zhang Z K, et al. An item-oriented recommendation algorithm on cold-start problem[J]. EPL(Europhysics Letters), 2011, 95(5)

[6] Sahebi S, Cohen W W. Community-based recommendations: a solution to the cold start problem[C]// Workshop on Recommender Systems and the Social Web(RSWEB). 2011

[7] Konstas I, Stathopoulos V, Jose J M. On social networks and collaborative recommendation[C]// Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2009: 195-202

[8] Zhou T, Ren J, Medo M, et al. Bipartite network projection and personal recommendation[J]. Physical Review E, 2007, 76(4)

[9] Zhang Li, Teng Pi-qiang, Qin Tao. Using Key Users of Social Network to Solve Cold Start Problem in Collaborative Recommendation Systems[J]. Information Technology Journal, 2013, 12(22)

[10] Boumaza A, Brun A. From neighbors to global neighbors in collaborative filtering: an evolutionary optimization approach[C]// Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference. ACM, 2012: 345-352

[11] Morid M A, Shajari M, Golpayegani A H. Who are the most influential users in a recommender system? [C]// Proceedings of the 13th International Conference on Electronic Commerce. ACM, 2011: 19

[12] Zeng W, Zeng A, Liu H, et al. Uncovering the information core in recommender systems: Scientific Reports 4: 6140[R]. 2014