AttentionRank$^+$:一种基于关注关系与多用户行为的图推荐算法

刘梦娟[1)] 王 巍[2)] 李杨曦[2)] 罗绪成[1)] 秦志光[1)]

[1)](电子科技大学网络与数据安全四川省重点实验室 成都 610054)

[2)](电子科技大学计算机科学与工程学院 成都 611731)

摘 要 该文提出一种基于关注关系和多用户行为的图推荐算法 AttentionRank$^+$,目的是为网络系统用户提供感兴趣的物品推荐.算法思路如下:首先根据用户对物品的多种反馈建立"用户-物品"反馈图,根据用户间的关注行为建立用户兴趣图;分别从每个用户节点出发,在反馈图上完成一轮 Random Walk,得到每个用户节点与反馈图上各节点间的相似度;将用户节点与物品节点的相似度信息在兴趣图上进行扩散,计算通过关注关系扩散后用户节点与物品节点间新的相似度;重复上述 Random Walk 和信息扩散的过程,直到反馈图上用户节点与各节点间的相似度收敛到稳定值;最后根据用户节点与物品节点间的相似度信息,计算每个用户的物品推荐列表.该文采用包含关注、收藏、上传等用户行为的 YouKu 数据集对推荐算法进行评价,实验结果表明 AttentionRank$^+$ 能够在用户行为稀疏的情况下,为用户提供高质量的视频推荐.

关键词 关注;多用户行为;随机游走;协同过滤;推荐

中图法分类号 TP399 **DOI 号** 10.11897/SP.J.1016.2017.00634

AttentionRank$^+$: A Graph-Based Recommendation Combining Attention Relationship and Multi-Behaviors

LIU Meng-Juan[1)] WANG Wei[2)] LI Yang-Xi[2)] LUO Xu-Cheng[1)] QIN Zhi-Guang[1)]

[1)](*Network and Data Security Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China, Chengdu* 610054)

[2)](*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu* 611731)

Abstract In this paper, we present a graph-based recommendation, AttentionRank$^+$, which considers not only the multi-behaviors but also the attention relationship between different users by extending the basic Random Walk algorithm. First, we build a weighted user-item graph based on the multi-behaviors, and then we build the attention graph based on the follower-followee relations between users that have similar interests. After that, AttentionRank$^+$ conducts a Random Walk on the weighted user-item graph and calculates the similarities between the target user node and any other node. If a user (*follower*) pays attention to other users (*followees*), we assume that the similarities between the follower and other nodes can be affected by his/her followees. So the similarity information of a user node can be spread along the interest edges to the followers on the attention graph. Each follower updates his/her own similarity information, and the new similarities are taken as the initial values for the next Random Walk. Repeat the above process until the similarity information of each user node converge to stable values. Finally, AttentionRank$^+$ creates a recommendation list of items for the target user according to the similarities between the user

收稿日期:2015-05-26;**在线出版日期**:2016-03-02. **本课题得到国家自然科学基金**(61202445,61272527,61300090,61133016)**资助**.
刘梦娟,女,1979 年生,博士,副教授,主要研究方向为数据挖掘、分布式计算. E-mail: mjliu@uestc.edu.cn. **王 巍**,男,1993 年生,硕士研究生,主要研究方向为数据挖掘、推荐算法. **李杨曦**,男,1987 年生,硕士研究生,主要研究方向为数据挖掘、推荐算法. **罗绪成**,男,1974 年生,博士,副教授,主要研究方向为计算机网络、人工智能. **秦志光**,男,1956 年生,博士,教授, 主要研究领域为信息安全、数据挖掘.

node and the other nodes in descending order. We evaluate the performance of AttentionRank$^+$ by using a YouKu dataset containing users follow, upload and collection records. The extensive experimental results show that AttentionRank$^+$ is capable of providing users with personalized and high quality video recommendation even when user behaviors are sparse.

Keywords　attention; multi-behaviors; random walk; collaborative filtering; recommendation

1　引　言

随着互联网数据的爆炸式增长，如何从海量数据中提取有效信息提供给用户是当前互联网应用亟待解决的问题. 个性化信息推荐是解决这一问题的有效方法，通过学习用户的历史行为，提取用户的兴趣特征，从而为用户推荐感兴趣的内容. 目前个性化信息推荐已经被各大互联网平台广泛采用，例如 Facebook、腾讯 QQ 等社交平台向用户推荐好友和圈子，YouTube、YouKu 等视频网站向用户推荐视频，淘宝、京东等电子商务网站向用户推荐商家和商品等. 已有的推荐算法主要包括：利用用户行为数据进行推荐[1]、利用用户和物品特征信息进行推荐[2]、利用时间[3]、位置[4]等上下文信息进行推荐、利用社交网络数据进行推荐[5-6]等.

目前应用最广泛的推荐算法是基于物品的协同过滤算法. 该算法根据用户对物品的历史反馈记录，计算物品与物品之间的相似程度，以此给目标用户推荐与其曾经喜欢的物品最相似的物品. 该算法的前提是认为物品之间的相似度与同时喜欢它们的用户数有关，系统中共同喜欢两个物品的用户数越多，则两个物品的相似度越高. 基于物品的协同过滤算法存在的问题是当用户的反馈数据稀疏时，推荐准确性会大幅降低，且不易于与其他信息融合. 图推荐算法[7-14]是一种更为灵活地利用用户反馈数据的推荐算法. 它将用户对物品的历史反馈通过二分图表示出来，然后在二分图上展开 Random Walk，以此计算出用户节点与物品节点间的相似度，从而为用户推荐与其最相似的物品. 由于二分图上不仅可以包含用户和物品节点，还可以包含用户和物品的标签节点，以及用户执行反馈时的上下文等信息节点，因此图推荐算法除了给用户推荐感兴趣的物品以外，还可以扩展推荐具有共同兴趣的好友、用户的兴趣标签等信息.

基于用户反馈数据的推荐算法存在一个共同问题，即当系统中用户反馈非常稀疏时，会导致推荐性能下降. 目前，越来越多的网络平台支持用户之间建立关注关系，例如 Facebook 的好友关注功能，YouKu 等视频网站的用户订阅功能等. 这种关注行为被认为是关注者和被关注者可能具有共同兴趣的一种隐含表达. 因此可以利用这种关注关系提高推荐算法的性能，特别是解决用户反馈稀疏或者冷启动的问题. 同时，随着网络平台的功能越来越丰富，用户可以在平台上对物品产生各种反馈，例如视频网站中用户可以产生浏览、播放、收藏、评价、点赞、上传等反馈，不同反馈类型反映出用户对视频兴趣度的差异，因此可以通过量化不同用户反馈的权重来提高推荐性能.

综上所述，本文提出一种基于关注关系和多用户反馈的图推荐算法 AttentionRank$^+$. 该算法的主要贡献包括：(1) 针对用户对物品的多种行为反馈，提出一种将多种反馈归一化为一种用户反馈的表示方法，并用于计算反馈图中“用户-物品”边的权值；(2) 在反馈图中，提出一种基于 Hammock 宽度的“用户-用户”边构建方法，以提高兴趣相似用户之间的游走概率；(3) 提出一种基于朴素贝叶斯分类器的用户兴趣图构建方法；(4) 提出一种基于反馈图 Random Walk 和兴趣图相似度信息扩散的节点间相似度的计算方法. 论文利用 YouKu 数据集验证所提出算法的性能，实验结果证明，即使在用户行为稀疏的情况下，AttentionRank$^+$ 能够在准确率、召回率、覆盖率等多项指标上取得较好的性能.

2　相关工作

随着推荐系统在产业界获得巨大成功，推荐算法的研究逐渐受到各国学者的重视. 其中最重要、应用最广泛的是基于用户行为数据的协同过滤算法 (Collaborative Filtering，CF). 目前，CF 算法主要分为两大类：建立邻域的方法和建立模型的方法. 其中，建立邻域的方法可进一步分为基于用户的协同过滤算法 (UserCF) 和基于物品的协同过滤算法 (ItemCF). UserCF 的基本思想是：通过计算用户间

的相似度，找到与目标用户最相似的用户构成用户邻域，给目标用户推荐和他兴趣最相近的邻域用户所喜欢的物品[15-16]. 与 UserCF 不同，ItemCF 的基本思想是通过计算物品之间的相似度，找到与目标物品最相似的物品构成物品邻域，给用户推荐和其之前喜欢物品相似的物品[17-18]. 目前，ItemCF 已被广泛应用在 Netflix、亚马逊、YouTube 等大型网站.

文献[19]将上述两种邻域方案结合起来提出一个基于标准差的混合协同推荐方案 SD-HCF，用以实现向目标用户推荐所需要的物品(web service). 该方案的基本思路是分别计算用户之间的相似度和物品之间的相似度，然后通过 UserCF 和 ItemCF 分别计算出目标用户对每个物品的打分值，将两个分值进行综合，最后根据目标用户对物品的综合打分按照由高到低的顺序进行推荐. 此外，作者还提出了 IF-UCF 方案，用以预测每个用户通过特定 web services 提供者访问 web services 的频率，从而实现为每个 web service 提供者推荐用户的功能. 邻域方案的优点是简单、易于实现，不需要考虑用户和物品的具体内容，但是运行效率低于建立模型的方案，且不能很好地解决用户反馈数据稀疏的问题.

建立模型的推荐方法主要包括隐语义模型的方案、图模型的方案、基于贝叶斯网络的方案等. 其中，隐语义模型方案又称为矩阵分解方案，其核心思想是通过矩阵分解，将用户和物品映射到隐含特征空间中，用隐含特征来刻画用户与隐含特征、隐含特征与物品之间的关系，从而提高推荐准确率[20-26]. 基本方法[23]如下：首先将评分矩阵 R 映射到两个低维的隐含空间 P 和 Q 当中，然后通过 P 和 Q 的乘积来重构 R，这样原始矩阵 R 中没有评分的物品也有了相应的得分，通过删除用户已经评分的物品，对剩余物品的得分进行排序，即可得到每个用户最终的推荐列表. 根据上述原理，用户 u 对物品 i 的打分值 $\hat{r}_{ui}$ 可以用式(1)进行近似预测：

$$\hat{r}_{ui}=\boldsymbol{P}_u^{\mathrm{T}}\boldsymbol{Q}_i,\ \boldsymbol{R}\approx\boldsymbol{P}^{\mathrm{T}}\boldsymbol{Q} \tag{1}$$

其中：P 是用户的隐含特征矩阵，P 中的第 u 行表示用户 u 的隐含特征向量，记为 $\boldsymbol{P}_u(\boldsymbol{P}_u\in\mathbb{R}^f)$，$f$ 为隐含特征的维度；矩阵 Q 是物品的隐含特征矩阵，Q 中的第 i 行表示物品 i 的隐含特征向量，记为 $\boldsymbol{Q}_i(\boldsymbol{Q}_i\in\mathbb{R}^f)$. 除了基本的矩阵分解方法外，研究者还提出了基于奇异值分解[24]、基于非负矩阵分解[25]、基于概率矩阵分解[26]等推荐方法.

图模型方案是另一种通过建立模型来实现推荐的方法，它不仅能有效地计算图中节点的相似性[7]，还能发掘节点之间传递的关联性[8]. 最早的图推荐算法是文献[9]提出的基于 Random Walk 的 TSPR (Topic-Sensitive PageRank)算法. 该算法采用一阶 Markov-chain 计算游走概率，方法如下：分别从每个用户节点出发，在“用户-物品”二分图上进行一轮 Random Walk；每当到达一个节点时，需要判断是以概率 β 继续向下游走，还是以 $1-\beta$ 概率返回出发点重新游走；重复游走过程，直到二分图上用户与物品的游走概率(即相似度)收敛到稳定值；最终，根据用户对物品的游走概率生成每个用户的物品推荐列表. TSPR 算法实现简单，具有较高的推荐准确率，但是算法的时间复杂度较高，且也存在数据稀疏导致的性能下降问题. 文献[10]在 TSPR 的基础上，综合考虑用户的长期偏好和短期偏好，提出一种可改善推荐准确率的图推荐算法.

文献[11]对 TSPR 算法进行了扩展，提出了能够适应不同应用场景的混合推荐算法 PathRank. 该算法在原始“用户-物品”二分图的基础上，增加了与物品和用户相关的内容或者上下文信息作为节点和关联边，通过改进的 Random Walk 在扩展图上进行游走，最后得到一个考虑多种因素的混合推荐列表. 文献[12]提出一种基于图模型的多维度推荐算法，在二分图中不仅包含物品和用户节点，而且包含时间和地点信息，因此在推荐时可以根据不同场景需求，结合不同的节点信息进行推荐，例如可以仅依据用户偏好进行推荐，也可以依据时间场景下的用户偏好进行推荐，或者依据地点场景下的用户偏好进行推荐. 文献[13]利用用户的点击与查询信息，建立“用户-查询-视频”三部图，通过在三部图中进行概率游走，得到用户节点到各视频节点的访问概率，作为视频推荐的基础. 由于用户与视频之间的游走路径变得更多，因此数据稀疏性问题得到了一定程度的减轻.

另一方面，为了缓解数据稀疏性及冷启动问题，研究者将社交因素和信任信息引入协同过滤算法中. 文献[5]将好友信息引入到 UserCF 中，首先使用皮尔森系数计算用户相似度矩阵；然后基于相似度选择最邻近用户，如果最邻近用户集中有目标用户的好友，则增加具有好友关系的邻近用户权重；最后预测目标用户对物品的打分. 实验结果表明，改进后的算法性能优于仅使用用户反馈的协同推荐算法，也优于仅利用好友关系进行的推荐算法. 文献[6]

首先分析了好友购买行为对用户偏好的影响；然后提出基于社交关系和贝叶斯后验概率模型的排名算法SBPR，实验证明利用好友关系不仅可以提高推荐性能，而且能够一定程度解决冷启动问题.

文献[27]通过定义用户之间的信任度，建立用户信任传播网络，使用信任度代替用户间的相似度权值，计算目标用户最信任的邻近用户，基于邻近用户对物品的打分来计算目标用户对物品的偏好. 文献[28]提出一种基于信任度的聚类方法，将位于相同簇中的用户作为邻近用户，基于邻近用户对物品的打分，预测目标用户对物品的打分. 文献[29]提出了一种基于图聚类的协同过滤方法，该方法通过将用户构成一张无向图，在无向图中寻找最稀疏子图，并以此子图中各节点为中心进行用户的聚类，最后利用与目标用户位于相同类的其他用户的反馈数据进行推荐，其中用户相似度是利用皮尔森系数和信任度的加权和来计算的.

综上，与本文研究最相关的是基于 Random Walk 的图推荐算法，然而已有的图推荐算法通常都只考虑了简单的用户反馈或者好友关系来计算节点相似度，没有考虑关注行为对用户兴趣的隐含体现，也很少有算法将多种用户反馈融合到一起进行考虑. 本论文提出的 AttentionRank$^+$ 算法是首次将关注行为和用户多种反馈结合起来的图推荐算法.

3　算法设计

3.1　概　述

本文在基本的 TSPR 算法的基础上，提出一种新的基于图模型的节点相似度计算方法. 首先仍然使用 Random Walk 在“用户-物品”反馈图上计算每个用户节点与物品节点的相似度；然后将得到的用户节点与物品节点的相似度信息在用户兴趣图上进行扩散，扩散的规则是将被关注用户与所有用户以及物品节点的相似度信息传递给关注用户，关注用户将从被关注用户处获取的相似度信息和自己在反馈图上执行一轮 Random Walk 得到的相似度信息进行融合，形成一个新的关注用户与所有节点的相似度列表；以每个用户节点新的相似度信息为基础，重复执行 Random Walk 和信息扩散的过程，直到每个用户与反馈图上各节点的相似度收敛到稳定值为止. 最后，根据用户与所有物品的相似度收敛值，计算每个用户的物品推荐列表. 算法流程如图 1 所示.

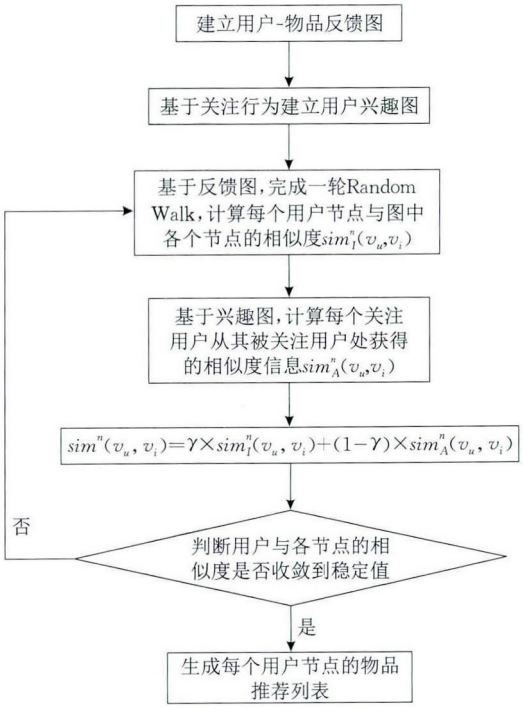

图 1　算法流程图

3.2　构建“用户-物品”反馈图

在 AttentionRank$^+$ 中，首先需要根据用户对物品的多种反馈建立“用户-物品”反馈图. 反馈图由 $G(V,E)$ 表示，节点集合 $V=V_I\cup V_U$ 包括用户节点集合 V_U 和物品节点集合 V_I，“用户-物品”边 $e(v_u,v_i)$ 表示用户 v_u 对物品 v_i 有反馈，$w(v_u,v_i)$ 为边的权重，表示用户对物品的感兴趣程度. 在现有的基于 Random Walk 的图推荐算法中，通常只简单考虑单一用户反馈，例如将用户对物品的评分作为反馈，边的权值为评分值. 这种设计没有体现出当前网络系统的多种用户反馈以及不同反馈体现出的兴趣差异度. 考虑在 YouKu 这样的视频网站中，一个用户观看视频和一个用户观看并且收藏(评价)该视频所体现出的兴趣是有差异的. 为此本论文提出一个新的基于反馈发生频率的“用户-物品”边权重计算方法. 该方法能体现用户对物品的不同反馈所隐含的用户兴趣差异性.

一种直观的考虑是，当系统中某种用户反馈类型发生频率越高，该类反馈所能体现的用户兴趣度就越低；发生频率越低，所能体现的用户兴趣度就越高. 例如视频观看和视频收藏，收藏反馈所体现出的感兴趣程度明显高于观看反馈. 基于这一考虑，设计“用户-物品”边的权重计算方法如下：假设系统中有 N 种用户反馈类型，每种反馈类型发生的总次数记为 $(F_1,F_2,\cdots,F_N)$，首先按照式(2)计算每种反馈类

型对应的归一化后的权重，记为$(f_1, f_2, \cdots, f_N)$，$f_1+f_2+\cdots+f_N=1$；然后根据用户v_u对物品v_i的历史反馈记录，计算边$e(v_u, v_i)$的权值$w(v_u, v_i)$，如式(3)所示，这里$num_m(v_u, v_i)$是用户v_u对物品v_i执行第m种反馈类型的次数，通常$num_m(v_u, v_i)$的取值为0或1，表示用户是否执行过这种反馈，因此“用户-物品”边的权重范围为(0,1].

$$f_i = tmp_i \Big/ \sum_{j=1}^{N} tmp_j,\ tmp_i = \Big(\sum_{j=1}^{N} \log F_j\Big) \Big/ \log F_i \quad (2)$$

$$w(v_u, v_i) = \sum_{m=1}^{N} num_m(v_u, v_i) \times f_m \quad (3)$$

在AttentionRank$^+$的“用户-物品”反馈图中，不仅设计了基于用户多种反馈类型的“用户-物品”边，还增加了反映用户共同反馈紧密程度的“用户-用户”边，以提高具有共同兴趣的用户节点之间的游走概率.“用户-用户”边的建立方法借鉴了文献[30]中*Hammock*宽度的概念，即如果两个用户共同反馈的物品数越多，则说明两个用户的兴趣相似度越高，在两个用户之间建立一条无向边，可以增加Random Walk时从用户节点到与其兴趣相似的用户节点，以及这些节点的反馈物品节点的游走概率. 用户相似度的计算可以采用皮尔森相关系数、余弦相似度等方法. 本论文提出一种基于“用户-物品”边权重的计算方法：假设用户x和用户y有共同反馈的物品集合为$\{i_1, i_2, \cdots, i_S\}$，这里共同反馈的物品数为$S$，则用户相似度计算如式(4)所示：

$$usersim(x, y) = \sum_{i \in \{i_1, i_2, \cdots, i_S\}} w(v_x, v_i) + w(v_y, v_i) \quad (4)$$

如果$usersim(x, y)$大于设定的相似度阈值k，则在两个用户节点之间建边，阈值k称为*Hammock*宽度.“用户-用户”边的权重计算方法如式(5)：

$$w(v_x, v_y) = \alpha \times \frac{usersim(x, y)}{S} \quad (5)$$

在本文中设置$\alpha=0.5$，用于调节“用户-用户”边与“用户-物品”边的权重保持一致，限制取值范围为(0,1]. 因此，最终建立的“用户-物品”反馈图是一个包含两类边的无向有权图. 为了使描述更为清晰，本论文引入一个实例对算法的关键步骤加以说明. 假设某视频网站有如下用户反馈记录：

A用户观看视频a，c，收藏视频a；

B用户观看视频a，b，c，d，收藏视频c，d；

C用户观看视频c，d，收藏视频d.

这里3个用户A、B、C总共有两种反馈类型：收藏和观看，因此可构建“用户-物品”反馈图如图2(a)所示，图中“用户-物品”边的权值可根据式(3)计算：

$$w(A,a)=w(B,c)=w(B,d)=w(C,d)=1.0,$$
$$w(A,c)=w(B,a)=w(B,b)=w(C,c)=0.4.$$

图2(b)是添加了“用户-用户”边的示意图，假设$\alpha=0.5, k=2$，则在A、B和B、C之间添加两条边. 根据式(5)可计算：

$$w(A,B)=0.7,\ w(B,C)=0.85.$$

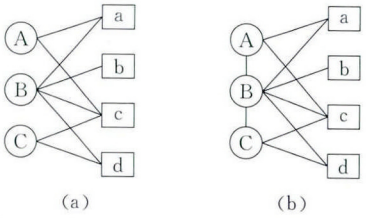

图2 用户-物品反馈图的例子

3.3 构建用户兴趣图

AttentionRank$^+$算法的特点是不仅基于反馈图来计算用户与物品节点的相似度，还结合用户之间的关注行为，考虑被关注用户(*followee*)的行为对关注用户(*follower*)行为的影响. 为此，算法提出根据用户的关注行为，构建用户兴趣图，被关注用户节点与反馈图中每个节点的相似度信息可沿着兴趣图中的边向关注用户扩散. 假设用户兴趣图表示为$AG(AV, AE)$，图中的节点是存在关注行为的用户节点集合$AV(AV \in V_U)$，如果用户x关注了用户y，则建立一条从被关注用户节点v_y指向关注用户节点v_x的边. 兴趣图是一个有向无权图，每个被关注用户节点与反馈图中各节点的相似度信息可沿着兴趣图中的边向关注用户节点扩散. 图3是一个用户兴趣图的例子，表示用户A关注了用户B和C，B关注了用户A和C，C没有关注任何人. 因此用户C在反馈图中获得的与各节点的相似度信息可沿着兴趣边扩散到用户A和B，而用户节点A和B会根据收到的相似度信息对自己与反馈图中各节点的相似度进行更新.

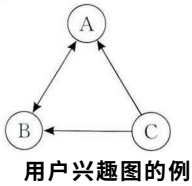

图3 用户兴趣图的例子

关注用户节点从兴趣图的信息扩散过程获得的节点相似度信息计算方法如下：假设用户节点v_u在兴趣图中关注的用户集合为$IN(v_u)$，$|IN(v_u)|$表示被v_u关注的用户节点个数，$sim_I(v_y, v_i)$表示被关注用户节点v_y在反馈图上与节点v_i的相似度信息，

则经过兴趣图扩散相似度信息后，用户节点 v_u 获得的反馈图中各节点的相似度信息计算方法如式(6)所示：

当 $|IN(v_u)|\neq 0$：

$$sim_A(v_u,v_i)=\frac{1}{|IN(v_u)|}\times\sum_{v_y\in IN(v_u)}sim_I(v_y,v_i) \quad (6)$$

需要说明的是如果用户 u 没有关注其他用户，$rank_A(v_u,v_i)$ 的计算如式(7)所示，即设置为用户节点 v_u 初始时与反馈图上各节点的相似度，这样设计的目的是保证每一轮 Random Walk 和信息扩散后，每个用户节点与反馈图上各节点的相似度之和为 1，并且使用户 u 在没有关注其他用户时，更多地关注与其连通距离更短的物品，从而更好地体现用户兴趣. 后续实验也证明了该设计的有效性.

当 $|IN(v_u)|=0$：

$$sim_A(v_u,v_i)=\begin{cases}1, & i=u\\ 0, & i\neq u\end{cases} \quad (7)$$

结合图 2 和图 3 的实例，A 节点关注了 B、C 节点，则 B、C 节点与反馈图中各节点的相似度信息将通过关注边扩散给 A 节点，因此 A 节点通过兴趣图的信息扩散过程获得的相似度如下所示：

$$sim_A(v_A,v_i)=\frac{1}{2}\times(sim_I(v_B,v_i)+sim_I(v_C,v_i)).$$

C 节点没有关注任何用户，因此 C 节点通过兴趣图的信息扩散过程获得的与反馈图中各节点的相似度信息为

$$sim_A(v_C,v_i)=\begin{bmatrix}A & B & C & a & b & c & d\\ 0 & 0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}.$$

3.4　Random Walk 和扩散过程

根据图 1 所示的 AttentionRank$^+$ 算法的基本步骤，在用户反馈图和兴趣图建立以后，首先基于反馈图进行 Random Walk：每个用户以自身为出发点，每当到达一个节点时，需要判断是以概率 β 继续向下游走，还是以 $1-\beta$ 概率返回出发点重新游走；如果继续游走，则从与当前节点相连的所有节点中，按权值比例，随机选择一个节点作为下一个游走节点；在完成一轮 Random Walk 后，各个节点被用户节点访问到的概率即为该用户节点与反馈图中各个节点的相似度. 假设 $sim_I^n(v_u,v_i)$ 表示完成第 n 轮 Random Walk 后，用户节点 v_u 与反馈图中每个节点 v_i 的相似度，$v_i\in V$，则 $sim_I^n(v_u,v_i)$ 的计算方法如式(8)所示：

初始时设置：

$$sim_I^0(v_u,v_i)=\begin{cases}1, & i=u\\ 0, & i\neq u\end{cases},$$

$$sim_I^n(v_u,v_i)=$$

$$\begin{cases}(1-\beta)+\beta\times\sum\limits_{v_v\in LINK(v_i)}\dfrac{sim_I^{n-1}(v_u,v_v)\times w(v_i,v_v)}{\sum\limits_{v_x\in LINK(v_v)}w(v_v,v_x)}, & i=u\\ \beta\times\sum\limits_{v_v\in LINK(v_i)}\dfrac{sim_I^{n-1}(v_u,v_v)\times w(v_i,v_v)}{\sum\limits_{v_x\in LINK(v_v)}w(v_v,v_x)}, & i\neq u\end{cases} \quad (8)$$

其中 $LINK(v_i)$ 表示在反馈图中与节点 v_i 直接相连的节点集合，$v_v\in LINK(v_i)$；$LINK(v_v)$ 表示与节点 v_v 直接相连的节点集合，$v_x\in LINK(v_v)$.

在用户-物品反馈图上完成一轮 Random Walk 后，将每个用户节点与反馈图中所有节点的相似度信息 $sim_I^n(v_u,v_i)$ 按照兴趣图中的边进行扩散，每个用户节点 v_u 可从其所有被关注用户节点处，获得该节点与反馈图中每个节点的相似度信息，并利用式(6)或者式(7)计算得到 $sim_A^n(v_u,v_i)$.

最终，每个用户节点与反馈图中各节点的相似度 $sim^n(v_u,v_i)$ 由式(9)计算，它综合考虑了从反馈图和从兴趣图分别得到的节点相似度信息，其中 $\gamma\in[0,1]$ 是调节参数，用于调节两种信息源对最终相似度的影响程度.

$$sim^n(v_u,v_i)=\gamma\times sim_I^n(v_u,v_i)+(1-\gamma)\times sim_A^n(v_u,v_i) \quad (9)$$

最后，判断用户节点与反馈图中每个节点的相似度是否收敛到稳定值. 如果没有收敛，则继续在反馈图上执行新一轮 Random Walk 和扩散过程，并将综合后的相似度信息作为新一轮游走的初始值，如式(10)所示；如果相似度值已经收敛，则终止算法，停止继续 Random Walk 和扩散. 需要注意的是算法的执行过程是一个 Markov 过程，即下一轮 Random Walk 时的初始状态只跟上一轮得到的节点相似度有关.

$$sim_I^n(v_u,v_i)=sim^n(v_u,v_i) \quad (10)$$

本文结合图 2 和图 3 的例子，介绍 Random Walk 和扩散过程中，A 节点与反馈图中各节点相似度的计算过程. 首先计算从用户节点 A 出发，经过 1 轮 Random Walk 后与图 2(b)上各节点的相似度.

初始设置：

$$sim_I^0(A,i)=\begin{bmatrix}A & B & C & a & b & c & d\\ 1 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}.$$

假设 $\beta=0.8$，在反馈图上经过 1 轮 Random Walk 后：

$$sim_I^1(A,a)=0.8\times\left(\frac{1}{2.1}+0\right)\approx 0.381,$$

$$sim_I^1(A,i)=\begin{bmatrix} A & B & C & a & b & c & d \\ 0.2 & 0.267 & 0 & 0.381 & 0 & 0.152 & 0 \end{bmatrix}.$$

根据式(6)可计算在兴趣图上相似度信息扩散后,用户节点 A 获得的节点相似度信息:

$$sim_A^1(A,i)=$$

$$\begin{bmatrix} A & B & C & a & b & c & d \\ 0.065 & 0.251 & 0.178 & 0.037 & 0.037 & 0.163 & 0.27 \end{bmatrix}.$$

假设 $\gamma=0.4$,根据式(10)可计算综合后的相似度信息:

$$sim^1(A,i)=$$

$$\begin{bmatrix} A & B & C & a & b & c & d \\ 0.119 & 0.257 & 0.107 & 0.175 & 0.022 & 0.159 & 0.161 \end{bmatrix}.$$

重复上述过程,直到 A 节点与各节点的相似度收敛到稳定值.

$$sim^{10}(A,i)=$$

$$\begin{bmatrix} A & B & C & a & b & c & d \\ 0.155 & 0.312 & 0.177 & 0.059 & 0.019 & 0.111 & 0.167 \end{bmatrix}.$$

3.5 生成物品推荐列表

在 Random Walk 和扩散过程完成后,得到每个用户节点与反馈图中各节点稳定的相似度值.本节介绍根据节点相似度生成每个用户的物品推荐列表.具体方法如下:删除每个用户 u 已经操作过的物品节点,将余下的物品节点按照与用户节点 v_u 的相似度降序排列;进行 Top-N 推荐时,选取相似度最大的 N 个物品推荐给用户.此外,图推荐算法还非常容易扩展进行具有共同兴趣的用户推荐,只需要选择与目标用户 u 相似度最高的、且未与 u 建立关注关系的 N 个用户生成用户推荐列表即可.

3.6 针对兴趣的关注用户提取

AttentionRank$^+$ 算法的基础是假设利用关注关系能够提高推荐算法的准确率,遗憾的是,本论文通过 YouKu 数据集的实验发现,并非所有用户的推荐准确率都能够提高,对于某些用户,其推荐准确率是下降的,如图 7 中的 A-AR 算法所示.究其原因是因为网络系统中用户间的关注行为具有一定的随意性,并非所有用户都受其关注对象行为的影响.为此本论文设计一个朴素贝叶斯分类器,用以将存在关注行为的用户划分为两类:受关注对象正影响或无影响的用户类 C1、受关注对象负影响的用户类 C2.根据分类结果将 C2 类用户及其对应的关注边从兴趣图中删除.

在设计分类器时,考虑每个用户元组有 2 个属性,记为 $\boldsymbol{u}=(x_1,x_2)$,其中属性 x_1 表示该用户关注其他用户的数量 $|IN(v_u)|$,属性 x_2 表示用户与其被关注对象的平均相似度,定义如式(11):

$$x_2(v_u)=\frac{\sum_{y\in IN(v_u)} sim(v_u,v_y)}{|IN(v_u)|} \tag{11}$$

其中:$sim(v_u,v_y)$ 表示用户节点 v_u 与被其关注用户节点 v_y 的相似度收敛值;$IN(v_u)$ 表示被 v_u 关注的用户节点集合,$|IN(v_u)|$ 表示被关注用户的个数.

为了构建分类器,首先需要建立一组用于构建分类器的训练集,它包括存在关注行为的用户元组以及与其关联的类标号,然而每个用户的关注行为是否对其推荐有效在数据集中是未知的,因此该分类器设计的难点是如何确定每个用户元组的类标号.为此本论文将训练集又细分为训练集和测试集,分别使用 TSPR 和 AttentionRank$^+$(这里将包含全部关注用户建立兴趣图的 AttentionRank$^+$ 算法简称为 A-AR)进行 Top-N 推荐,将测试集中每个用户在两种推荐算法下的性能进行对比:

(1)当用户在 A-AR 下的推荐性能比 TSPR 得到的推荐性能有提升或无明显区别时,则把该用户标记为 C1,即被关注对象的行为对关注者的行为有正面影响或无影响;

(2)当用户在 A-AR 下的推荐性能相比 TSPR 下降时,把该用户标记为 C2.

最终得到完整的包含元组和类标号的训练集.基于训练集,利用朴素贝叶斯分类器[31]可计算所有属性组合区间的分类判断规则(如表 1 所示);从而可根据任意用户 u 的关注用户数量及与被关注对象的平均相似度两个属性,预测其所属的分类.实验结果表明,基于改进后的用户兴趣图,本论文提出的算法能获得较大的性能提升.

表 1 贝叶斯分类规则(实验方案如 4.2 节)

相似度	关注数								
	1～2	2～3	3～4	4～5	5～9	9～21	21～51	51～101	101～280
0～0.05	C1	C2	C2	C2	C2	C2	C2	C2	C2
0.05～0.10	C1	C2	C2	C2	C2	C2	C2	C2	C2
0.10～0.15	C1	C2	C2	C2	C2	C2	C2	C2	C2
0.15～0.20	C1	C2	C1	C1	C2	C1	C1	C1	C2
0.20～0.25	C1	C1	C1	C1	C1	C1	C1	C1	C1
0.25～0.30	C1	C1	C1	C1	C1	C1	C1	C1	C1
0.30～0.35	C1	C1	C1	C1	C1	C1	C1	C1	C1
0.35～0.40	C1	C1	C1	C1	C1	C1	C1	C1	C1
0.40～0.45	C1	C1	C1	C1	C1	C1	C1	C1	C1
0.45～0.50	C1	C1	C1	C1	C1	C1	C1	C1	C1
0.50～0.55	C1	C1	C1	C1	C1	C1	C1	C1	C1

表 1 显示利用用户关注数和用户与被关注用户的平均相似度两个指标作为分类属性是有效的，可以较为清晰地划分出分类规则，特别是当用户与被关注用户的平均相似度能够有效反映关注用户对被关注用户的感兴趣程度时. 例如，当用户与被关注用户的平均相似度达到 0.2 时，无论该用户有多少关注对象，它都属于 C1 类；当用户与被关注用户的平均相似度低于 0.15 时，只有当用户关注数少于等于 2 时，该用户才属于 C1 类. 需要说明的是，这里得到的分类规则对于 YouKu 站点来说是具有普适性的. 在后续实验中，论文采用 YouKu 数据集中不同的数据子集进行训练能得到近似的分类规则，利用表 1 中给出的规则，对其他不重叠的 YouKu 数据集的用户进行分类，也确实能够提取出受关注行为正面影响或无影响的用户.

3.7　算法时间复杂度分析

为了方便分析，首先给出 AttentionRank$^+$ 算法的伪代码如算法 1.

算法 1.　AttentionRank$^+$.

输入：$\boldsymbol{D}_1$，$\boldsymbol{D}_2$，$\boldsymbol{D}_3$

输出：推荐列表

初始化数据：

$\boldsymbol{P}_{1,2,\cdots,m}$是$[m+n,1]$阶矩阵，其中 $\boldsymbol{P}_i[i]=1$，其余为 0

$\boldsymbol{sim}_{1,2,\cdots,m}$的初始值为 $\boldsymbol{P}_{1,2,\cdots,m}$

$\boldsymbol{sim}=[\boldsymbol{sim}_1,\boldsymbol{sim}_2,\cdots,\boldsymbol{sim}_m]^{\mathrm{T}}$

$\boldsymbol{M}$ 为 $\boldsymbol{D}_1$ 的概率转移矩阵，是$[m+n,m+n]$阶矩阵

$\boldsymbol{M}[i,j]=\boldsymbol{D}_1[i][j]/|\boldsymbol{D}_1[i]|$

```
1.  FUNCTION AttentionRank+ (D1, D2, D3, β, γ, steps)
2.    FOR t←1 to steps
3.      FOR i←1 to m
4.        sim_i = (1−β)P_i + β×M×sim_i
5.      END FOR
6.      UPDATE sim
7.      sim_A = D2×sim + D3
8.      FOR i←1 to m
9.        sim_i = γ sim_i + (1−γ) sim_A[i]^T
10.     END FOR
11.     UPDATE sim
12.   END FOR
13. END FUNCTION
```

假设数据集是 m 个用户对 n 个物品的反馈记录，以及 m 个用户之间的关注记录，则算法输入的用户反馈图 G_1 是一个 $(m+n)$ 阶的实对称矩阵 $\boldsymbol{D}_1$，主对角线上元素为 0，如果 G_1 中用户 m_i 对物品 n_j 发生反馈，则

$$\boldsymbol{D}_1[m_i,m+n_j]=\boldsymbol{D}_1[m+n_j,m_i]=w(v_i,v_j).$$

其余元素值为 0；兴趣图 G_2 输入是一个 m 阶方阵 $\boldsymbol{D}_2$，如果 G_2 中用户 m_i 指向用户 m_j，则 $\boldsymbol{D}_2[m_i,m_j]=1/|in(u)|$，其余元素值为 0；针对没有关注其他用户的用户 m_i，则新建一个 $\boldsymbol{D}_1$ 的同型矩阵 $\boldsymbol{D}_3$，$\boldsymbol{D}_3[m_i,m_i]=1$，其余元素值为 0. 进行 Random Walk 时，将 $\boldsymbol{D}_1$ 初始化为概率转移矩阵 $\boldsymbol{M}$. Random Walk 的迭代公式如下：

$$\boldsymbol{sim}_i^t=(1-\beta)\boldsymbol{P}_i+\beta\times\boldsymbol{M}\times\boldsymbol{sim}_i^{t-1},$$

其中 $\boldsymbol{sim}_i^t$ 为 $(m+n)\times1$ 列向量，表示用户 i 迭代后的概率分布，其时间复杂度为 $m(m+n)^2$.

设 $\boldsymbol{sim}=(\boldsymbol{sim}_1,\boldsymbol{sim}_2,\cdots,\boldsymbol{sim}_m)^{\mathrm{T}}$，基于兴趣图的相似度信息扩散过程，在算法中是进行一次矩阵乘法 $\boldsymbol{sim}_A=\boldsymbol{D}_2\times\boldsymbol{sim}$，其时间复杂度为 $m^2(m+n)$.

最后更新扩散后的概率分布，执行一次矩阵加法，复杂度可忽略. 假设 t 轮后访问概率收敛，则算法的时间复杂度为

$$T(m+n)=T(t\times m(m+n)(2m+n))=O((m+n)^3).$$

3.8　算法收敛性分析

论文首先分析基本的 TSPR 算法的收敛性，其迭代公式如下：

$$\boldsymbol{sim}_i=(1-\beta)\boldsymbol{P}_i+\beta\times\boldsymbol{M}\times\boldsymbol{sim}_i$$
$$\Rightarrow(\boldsymbol{I}-\beta\times\boldsymbol{M})\times\boldsymbol{sim}_i=(1-\beta)\boldsymbol{P}_i.$$

因为 $(\boldsymbol{I}-\beta\times\boldsymbol{M})$ 是一个严格的对角占优矩阵，所以 $(\boldsymbol{I}-\beta\times\boldsymbol{M})$ 可逆. 因此

$$\boldsymbol{sim}_i=(1-\beta)(\boldsymbol{I}-\beta\times\boldsymbol{M})^{-1}\times\boldsymbol{P}_i.$$

因此 TSPR 可转化为一步概率转移矩阵求稳态分布，根据文献[32]可证，TSPR 能够收敛.

同理 AttentionRank$^+$ 的迭代公式如下：

$$\boldsymbol{sim}_i=(1-\beta)\boldsymbol{P}_i+\beta\times\boldsymbol{M}\times\boldsymbol{sim}_i,$$
$$\boldsymbol{sim}_A=\boldsymbol{D}_2\times\boldsymbol{sim}+\boldsymbol{D}_3,$$
$$\boldsymbol{sim}_i=\gamma\times\boldsymbol{sim}_i+(1-\gamma)\boldsymbol{sim}_A[i]^{\mathrm{T}}.$$

每一轮迭代更新，可保证 Random Walk 与扩散满足以下条件：(1) $\boldsymbol{sim}_i$ 中的值和为 1，且 $\boldsymbol{sim}_i$ 中的值都大于等于 0；(2) 关注图上概率扩散后仍然保证结果满足 Random Walk 的条件，即 $\boldsymbol{sim}_i$ 中的值和为 1，且 $\boldsymbol{sim}_i$ 中的值都大于等于 0. 因此 AttentionRank$^+$ 具有遍历性，故此算法是收敛的.

此外，论文也从实验角度观察了算法的收敛性. 基于论文中图 2(b)和图 3 给出的用户-物品反馈图和兴趣图进行 Random Walk 和信息扩散，观察用户节点 A 在不同迭代次数时与反馈图中各节点的相似度，如图 4 所示. 可以发现当迭代到第 10 轮时，就已经能够得到一个稳定的相似度分布了.

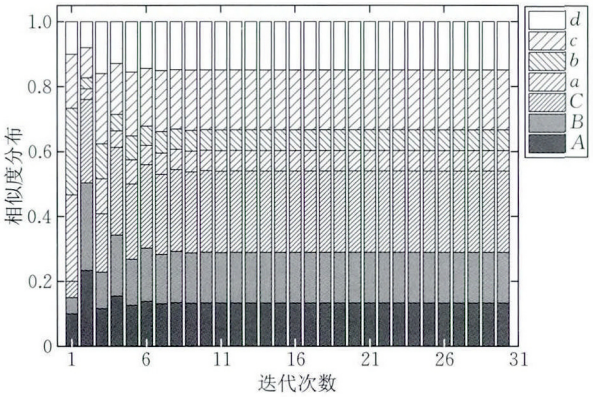

图 4 用户 A 与反馈图中各节点的相似度分布

4 实验及结果分析

4.1 实验数据集及评价指标

论文实验采用 YouKu 数据集①完成，该数据集是从国内知名视频网站 YouKu② 爬取的. 初始数据集包括 113 372 个用户，617 987 条用户关注记录，6 156 133部用户上传和收藏的视频. 由于只包含了每个用户对视频上传和收藏的反馈记录，没有对观看视频的反馈记录，因此初始数据集中用户反馈是非常稀疏的，经过反复过滤后得到本论文目前使用的两个数据集，如表 2 所示. 本文对数据集 I 中的用户反馈记录进行了分析，图 5 是每部视频对应的感兴趣的用户数的分布情况，其中只有少数视频拥有大量的感兴趣用户，而 60.83% 的视频对应的用户反馈数小于等于 5. 图 6 是每个用户的反馈数分布情况. 观察可知，数据集中用户的反馈数分布均服从

表 2 两个数据集的记录

数据集	用户数	关注记录	视频数	反馈记录
数据集 I	2877	10071	3510	94018
数据集 II	4430	12398	4107	138446

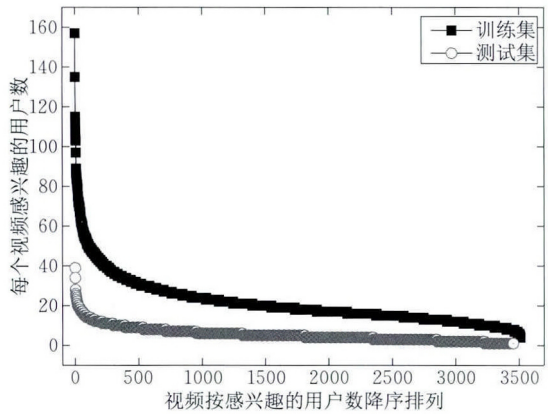

图 5 每部视频对应的反馈用户数分布

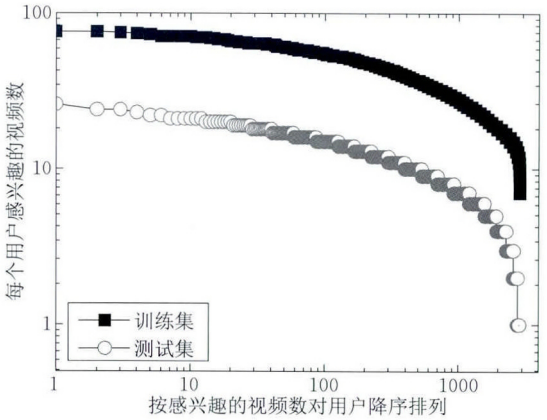

图 6 每个用户执行的反馈数分布

长尾分布，特别是在测试集中，用户反馈数少于 5 次的用户占总用户数的 45.67%，因此在 YouKu 数据集中用户反馈数据是非常稀疏的.

论文使用准确率(*precision*)、召回率(*recall*)、覆盖率(*coverage*)这 3 个指标[33]对推荐算法的性能进行评价. 准确率描述推荐列表中有多少比例的物品是用户确实喜欢的；召回率描述测试集中用户喜欢的物品中有多少比例是包含在推荐列表中；覆盖率是指在全部物品中，推荐列表的物品所占的比例，该指标用于反映推荐算法发掘长尾的能力，覆盖率越高，推荐给用户的物品越多样，越具有新颖性. 评价指标定义如式(12)～(14)所示，其中 V_U 是数据集中的用户集合，V_I 是物品集合，$R(u)$ 表示给用户 u 推荐的 N(在本文的实验中 $N=15$)个物品集合，$T(u)$ 表示用户 u 在测试集中喜欢的物品集合.

$$precision=\frac{\sum_{u\in V_U}|R(u)\cap T(u)|}{\sum_{u\in V_U}|R(u)|} \tag{12}$$

$$recall=\frac{\sum_{u\in V_U}|R(u)\cap T(u)|}{\sum_{u\in V_U}|T(u)|} \tag{13}$$

$$coverage=\frac{|\bigcup_{u\in V_U}R(u)|}{|V_I|} \tag{14}$$

本论文包含 5 组实验，分别用于验证贝叶斯分类器的普适性，多类型用户反馈、*Hammock* 宽度、相似度调节参数以及稀疏数据对推荐算法性能的影响. 实验中使用了 9 种推荐算法作为对比，如表 3 所示. 为了简化，本节将 AttentionRank$^+$ 算法简称为 AR 算法，基于全部关注用户建立兴趣图的 AR 算

① YouKu 数据集，http://pan.baidu.com/s/1qWmgWZM
② YouKu 视频网站，http://www.youku.com/

法简称为 A-AR，基于贝叶斯分类器提取的关注用户建立兴趣图的 AR 算法简称为 T-AR 算法.

表 3　9 种推荐算法的说明

算法名称	说明
TSPR	基本的 Topic-Sensitive PageRank 算法[9]
H-TSPR	在反馈图上添加“用户-用户”边的 TSPR 算法
A-AR	基于全部关注用户建立兴趣图的 AR 算法
T-AR	基于提取关注用户建立兴趣图的 AR 算法
ItemCF	基于物品的协同过滤算法[17]
Social-CF	融入社交关系的基于用户的协同过滤算法[5]
BPR	基于贝叶斯后验概率学习模型的推荐算法[34]
SBPR	融入社交关系的 BPR 算法[6]
RankALS	基于矩阵分解模型的推荐算法[23]

4.2　贝叶斯分类器的普适性验证

实验 1 用于验证基于朴素贝叶斯算法提取的分类器对于 YouKu 数据集构建用户兴趣图是否具有普适性. 该实验基于数据集 II 完成，首先将 4430 个用户的关注记录分为互不相同的两部分（两个子集中用户关注记录是不重叠的）：子集 II-1，包含 2681 个用户的 8203 条关注记录，以及对 4107 个物品的 101 343 条反馈记录；子集 II-2，包含 1749 个用户的 4195 条关注记录，以及对 4099 个物品的 57 691 条反馈记录. 然后将子集 II-1 的反馈记录按照 4∶1 的比例划分为训练集和测试集，并根据 3.6 节中所述的方法基于训练集得到提取关注用户的贝叶斯分类规则，如表 1 所示. 最后将子集 II-2 的反馈记录也按照 4∶1 的比例划分为训练集和测试集，并直接使用表 1 中给出的分类规则对子集 II-2 的训练集中的用户进行分类，得到用户兴趣图.

为了对比，实验 1 在训练模型时使用了 H-TSPR 和 T-AR 两个算法，参数设置如下：$k=15$，$\gamma=0.5$，$\beta=0.8$. 实验结果如表 4 所示. 结果显示基于子集 II-1 提取的有效用户分类器，对于子集 II-2 的有效用户提取同样适用，因此，本论文提出的基于用户关注数和用户与被关注用户平均相似度的贝叶斯分类方法具有一定的普适性.

表 4　不同数据集下使用相同贝叶斯分类器的结果对比

算法	数据集	准确率/%	召回率/%	覆盖率/%
H-TSPR	数据集 II-1	4.04	9.00	23.93
T-AR	数据集 II-1	4.79	10.67	50.06
H-TSPR	数据集 II-2	3.59	8.23	22.22
T-AR	数据集 II-2	3.96	9.07	48.01

4.3　多用户反馈类型对算法性能的影响

实验 2 用于分析不同用户反馈类型对推荐性能的影响. 实验基于数据集 I 完成，随机将数据集中的用户反馈记录按照 5∶5 的比例划分为训练集与测试集，参数设置与实验 1 相同. 分别在考虑用户反馈类型差异和不考虑用户反馈类型差异两种情况下测试了 TSPR 和 T-AR 算法的推荐性能，如果不考虑反馈类型差异，则将用户-物品反馈图中边的权值均设为 1；如果考虑反馈的类型差异，则按照 3.2 节的式(2)、(3)来计算“用户-物品”边的权值. 实验结果如表 5 所示. 结果表明，考虑用户反馈的不同类型确实能够在一定程度上提升推荐的准确率和召回率，这是因为在 YouKu 场景下，不同反馈类型在反应用户对物品的喜好程度上确实具有差异性.

表 5　多类型用户反馈对算法性能的影响

算法	准确率/%	召回率/%	覆盖率/%	有无考虑用户反馈的不同类型
TSPR	8.70	7.97	42.74	无
TSPR	8.72	7.99	42.88	有
T-AR	9.66	8.86	69.89	无
T-AR	9.71	8.90	70.31	有

4.4　*Hammock* 宽度对算法性能的影响

实验 3 考察不同 *Hammock* 宽度值在反馈图中建立“用户-用户”边对算法性能的影响，从而发现适合 YouKu 数据集的最佳宽度值. 实验基于数据集 I 完成，分别从数据集 I 中随机选择 80% 的反馈记录以及全部的关注记录作为训练集，20% 的反馈记录作为测试集. 实验将 *Hammock* 宽度 k 从 1～20 以步长 5 为单位增加，为了进行对比，实验考察了 TSPR、H-TSPR、A-AR、T-AR 共 4 种推荐算法的性能. 实验结果如图 7～图 9 所示.

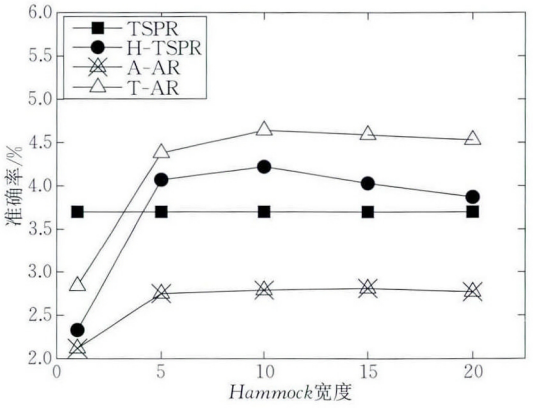

图 7　*Hammock* 宽度对推荐准确率的影响

实验结果显示，当 $k=1$ 时，3 种建立“用户-用户”边的推荐算法的准确率和召回率均低于不建立“用户-用户”边的 TSPR 算法，这是因为阈值过低，在建立反馈图时，会添加大量的“用户-用户”边，而其中绝大部分用户的兴趣相似度可能很低，同时会完全改变原来基于用户反馈建立的图结构. 而随着

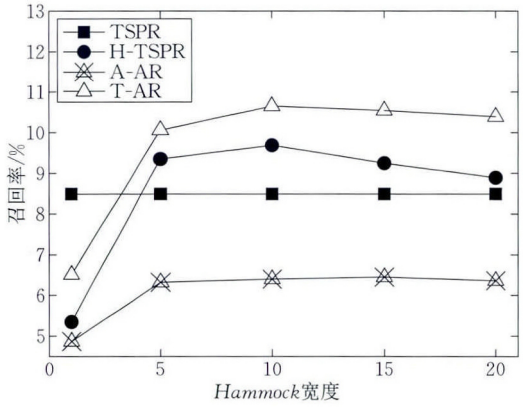

图 8 *Hammock* 宽度对推荐召回率的影响

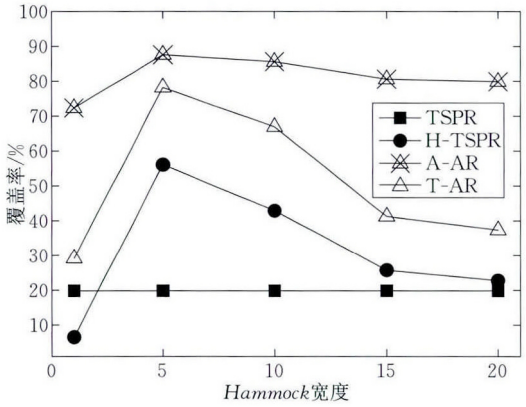

图 9 *Hammock* 宽度对推荐覆盖率的影响

宽度值 k 增加，H-TSPR、A-AR 和 T-AR 这 3 种算法的推荐性能均呈现先增加，达到峰值之后再呈下降的趋势，其中 H-TSPR 和 T-AR 的准确率和召回率都优于 TSPR. 这是因为，随着 k 增加，反馈图中建立的“用户-用户”边的数量会随之减少，但是加边的用户之间的相似度会逐渐升高，加边使得相似用户之间的游走距离变短，所以游走到相似用户的有反馈的物品的距离也相应变短，目标用户更容易通过其相似用户来发现其感兴趣的物品，所以推荐结果准确度也会提高，直到达到最优效果 $k=10$.

当 k 值继续增加，H-TSPR 和 T-AR 的准确率和召回率会略微下降. 这是因为 k 值越大，反馈图中满足相似度要求的用户节点对就越少，添加的“用户-用户”边就越少，直到没有一对用户节点要求时，反馈图就退化为原始的只考虑用户反馈记录的图结构，因此，H-TSPR 的结果与 TSPR 的结果最终变得一致. 而 T-AR 还包含了基于兴趣图的相似度信息扩散过程，因此，T-AR 的推荐性能还是优于 H-TSPR. A-AR 算法的准确率和召回率均大幅低于 T-AR，这是因为 A-AR 使用全部关注用户来构建兴趣图，而在 YouKu 数据集中，并非所有存在关注行为的用户都对其被关注用户的反馈物品感兴趣，因此简单的基于所有关注用户构建兴趣图，进行相似度信息的扩散，不但不能为用户提供其感兴趣的内容，反而会使得用户不感兴趣的内容被推荐给用户，从而使得推荐准确度下降. T-AR 算法通过贝叶斯分类器提取真正有共同兴趣的关注用户和被关注用户，然后基于提取用户来构建兴趣图，确实能获得更好的推荐性能.

由图 9 可见，所有的添加“用户-用户”边的改进算法，其覆盖率均比基本的 TSPR 算法要高. 因为 Random Walk 的原理是，将起始用户节点的访问概率随机扩散到与其相连通的节点上去，而访问概率主要集中在与起始节点连通距离短的节点上，因此图中的边越多，存在的连通路径就越多，更有利于将节点的访问概率扩散到更大的节点范围，从而使推荐列表中的物品更多样化，使推荐覆盖率大幅提升. 综上分析，通过合理设置 *Hammock* 宽度值，本论文提出的 T-AR 算法能够在各项指标上超越基本的 TSPR 算法. 当 *Hammock* 宽度为 10 时，T-AR 在准确率、召回率指标上均达到最大值.

4.5 相似度调节参数 γ 对算法性能的影响

实验 4 考察式(9)中的调节参数 γ 的变化对推荐算法性能的影响，从而分析出适合于 YouKu 数据集的 γ 设置. γ 用于反映在综合考虑用户自己反馈记录以及用户关注对象反馈记录来计算用户与反馈图中各节点相似度时的比重. 当 γ 取 1 时，T-AR 和 A-AR 算法将只使用用户反馈信息来进行计算，因此结果与 H-TSPR 一致. 随着 γ 值逐渐变小，从用户关注对象处得到的节点相似度信息在计算用户自身与反馈图中各节点相似度时占的比重逐渐变大，当 γ 取 0 时，将完全依赖从关注对象处获得的相似度信息来计算自己与各节点的相似度值. 实验 4 使用与实验 3 相同的数据集，设置 γ 取值在[0,1]范围内，以 0.2 为步长进行实验，实验结果如表 6 所示.

从表 6 中可以看出，当 $\gamma=1$ 时，不考虑关注行为对节点相似度的影响，因此 T-AR 和 A-AR 退化为 H-TSPR，所有指标都相等；当 $\gamma=0$ 时，T-AR 和 A-AR 只使用从被关注用户处获得的相似度信息来计算自己与物品的相似度，完全不考虑自己对物品的反馈记录，因此推荐的准确率和召回率都非常低. 此外，T-AR 的推荐性能高于 A-AR，这是因为 A-AR 使用全部关注用户来构建兴趣图，但是并非所有用户都对其被关注用户感兴趣的内容感兴趣，如果使用这部分被关注对象的相似度，来计算用户自

表 6　γ 对算法推荐性能的影响

γ	准确率/%				γ	召回率/%				γ	覆盖率/%			
	TSPR	H-TSPR	A-AR	T-AR		TSPR	H-TSPR	A-AR	T-AR		TSPR	H-TSPR	A-AR	T-AR
0.0	—	—	0.42	0.52	0.0	—	—	0.97	1.20	0.0	—	—	64.42	70.88
0.2	—	—	2.39	4.57	0.2	—	—	5.49	10.48	0.2	—	—	92.22	89.69
0.4	—	—	2.54	4.65	0.4	—	—	5.83	10.67	0.4	—	—	91.68	86.01
0.5	—	—	2.60	4.69	0.5	—	—	5.97	10.75	0.5	—	—	90.91	83.45
0.6	—	—	2.62	4.52	0.6	—	—	6.02	10.37	0.6	—	—	89.80	80.60
0.8	—	—	2.79	4.64	0.8	—	—	6.40	10.65	0.8	—	—	85.61	66.92
1.0	3.7	4.22	4.22	4.22	1.0	8.49	9.69	9.69	9.69	1.0	19.83	42.91	42.91	42.91

己与其他节点的相似度，将使用户的推荐列表出现他并不感兴趣的视频，而使他真正感兴趣的视频被淹没. T-AR 算法通过贝叶斯分类器能够将真正对被关注用户反馈内容感兴趣的用户提取出来，从而建立有效的用户兴趣图，而基于关注边扩散获得的被关注用户节点与各节点的相似度信息，确实对提高推荐性能有较大帮助. 当 γ 取 0.5 时，T-AR 的推荐准确率和召回率达到峰值.

对于覆盖率指标，引入关注行为的推荐算法的覆盖率明显高于不考虑关注行为的 TSPR 和 H-TSPR 算法. 这是因为在考虑关注行为后，用户节点与物品节点的相似度不仅考虑了该用户自己对物品的反馈记录，还考虑了其被关注对象对物品的反馈，因此使得推荐物品的覆盖范围更加广泛. 综上分析，T-AR 算法在推荐准确率、召回率、覆盖率等关键指标上的表现优异，说明引入用户关注行为构建兴趣图，基于关注边进行相似度信息的扩散，确实能够更加准确地发现用户真实的兴趣，给目标用户推荐其感兴趣的内容.

4.6　各种推荐算法的性能对比

表 7 是在 T-AR 最佳参数设置下与其他推荐算法的性能对比. 除了基于图模型的 4 种推荐算法之外，还增加对比了 2 种协同过滤算法以及 3 种基于模型的推荐算法. ItemCF 是基于物品的协同过滤算法；social-CF 是考虑朋友关系的基于用户的协同过滤算法；BPR 是基于贝叶斯后验概率模型的推荐算法；SBPR是BPR的改进算法，它将社交关系引入到了物品建模的过程中，将物品分为用户操作过的物品、用户未操作过但用户好友操作过的物品以及用户跟好友均未操作过的物品三类，然后通过对这三类物品的两两正负例元组训练得到最终的模型，用于推荐；RankALS 是一种基于矩阵分解的推荐算法，通过用户和物品的特征向量的乘积来重构矩阵获得推荐.

表 7　最佳参数设置下的实验性能对比表

算法	准确率/%	召回率/%	覆盖率/%
TSPR	3.70	8.49	19.83
H-TSPR	4.22	9.69	42.91
A-AR	2.60	5.97	90.91
T-AR	4.69	10.75	83.45
ItemCF	4.51	10.36	73.90
Social-CF	5.24	12.02	79.86
BPR	2.21	5.07	12.45
SBPR	0.84	2.31	0.65
RankALS	2.41	5.30	19.69

从表 7 中可以看出，T-AR 算法的性能较之基本的 TSPR 和 H-TSPR 算法有较大幅度的提升，特别是在覆盖率上，T-AR 算法比 TSPR 提升了 3.21 倍，也高于 ItemCF 和 social-CF 两种基于邻域的协同过滤算法，这说明在挖掘长尾资源方面，T-AR 算法具有良好的性能. 这是因为通过兴趣图中的关注边，被关注用户与物品的相似度信息得到了一定程度的扩散，使得最终的推荐列表中的视频除了与用户自身的兴趣相关以外，也具有一定的多样性. 在准确率和召回率方面，T-AR 算法略高于 ItemCF，低于social-CF，这说明 YouKu 数据集更适合于使用那种着重于反映和用户兴趣相似的小群体的热点内容的推荐算法，例如 Social-CF，而不太适合采用反映个人用户历史兴趣的推荐算法，如 ItemCF.

BPR 和 SBPR 的推荐性能均大幅低于其他算法，这是因为这种基于贝叶斯后验概率模型的推荐算法需要根据特定的应用环境对模型参数进行调节，所以直接将该学习算法应用到本论文的 YouKu 数据集中，推荐结果并不是很理想. 社交关系对 SBPR 的建模影响巨大，在本文的 YouKu 数据集实验中，直接使用关注关系代替原本的社交关系来建模，得到的推荐效果很差，远低于不应用关注关系的 BPR. 这是因为，在 YouKu 数据集原始的用户关注关系中，绝大部分都不能体现用户之间的共同兴趣，基于非共同兴趣的用户行为进行推荐，会极大地损害推荐性能. RankALS 的推荐性能优于 BPR 和 SBPR，但是比其他 6 种推荐算法的性能都低.

4.7 更稀疏的用户反馈对算法性能的影响

为了进一步测试 T-AR 算法在稀疏用户反馈数据集中的性能，实验 5 将数据集 I 重新按照 5∶5 的比例划分为训练集和测试集，新的训练集包括 2877 个用户、3510 部视频，47096 条反馈记录，10071 条关注记录，因此平均每个用户的反馈数为 16；平均每部视频的反馈数为 13. 与 4∶1 划分的训练集相比，这个训练集的数据更为稀疏(4∶1 划分的训练集中，平均每个用户的反馈数为 26，平均每部视频的反馈数为 21).

实验结果如表 8 所示，在更为稀疏的用户反馈数据集中，所有算法的准确率都有所提升，而召回率均在一定程度上下降. 这是因为重新划分数据集后，测试集中的用户反馈数增加，从而使得推荐视频被命中的数量增加，因此准确率上升. 虽然测试集中推荐视频的命中数增加，但是用户在测试集中的实际反馈数也增加，且高于命中数增加值，因此整体(除 RankALS 方案以外)的召回率有所下降.

表 8 更稀疏的用户行为对算法性能的影响

算法	准确率/%	召回率/%	覆盖率/%
TSPR	8.91	8.18	42.08
H-TSPR	9.03	8.30	46.97
A-AR	5.38	4.94	89.39
T-AR	9.73	8.93	71.72
ItemCF	5.79	6.30	79.27
Social-CF	9.08	8.34	88.62
BPR	4.80	4.39	31.11
SBPR	1.95	2.23	0.61
RankALS	7.15	6.51	16.48

前 6 种基于历史反馈的算法中，ItemCF 的性能在稀疏反馈数据集上最差，因为用户反馈数据过少，其基于物品的相似性计算准确率就会变低，Social-CF 在稀疏反馈数据集上也存在这个问题，用户相似度计算性能变差，所以其表现并不如 T-AR. 而T-AR通过添加反馈图上的“用户-用户”边，增加了 Random Walk 的路径数(即节点之间的连通性)，且通过在兴趣图上进行相似度信息的扩散，有助于增加稀疏行为用户获取自己感兴趣的视频信息的可能性，可以在一定程度上解决用户冷启动问题. 后 3 种基于模型的算法中，仍然是 SBPR 表现最差，原因还是 YouKu 数据集的社交关系并不是完全有效，而 SBPR 过分依赖于社交关系. RankALS 在稀疏行为数据集上表现较好，这正说明了基于矩阵分解的推荐算法对稀疏数据具有良好的适应性.

5 总结及未来工作

本论文在大量实验的基础上提出一种基于关注行为和多用户反馈的图推荐算法 AttentionRank^{+}. 该算法充分考虑了目前网络系统中，用户多种反馈行为对用户兴趣差异性的体现，而且通过融入反映用户收看兴趣的关注行为，能够为反馈记录稀疏的用户提供高质量的物品推荐. 通过从 YouKu 视频网站爬取的实际用户反馈记录和关注记录作为数据集，对所提出算法的性能进行了比较和分析. 实验结果表明，T-AR 确实能够在用户行为稀疏的情况下，有效提高视频推荐的性能. 利用关注行为对现有推荐算法的性能进行提升是目前推荐算法研究的热点之一，目前论文只是采用简单的贝叶斯分类器对存在关注行为的用户进行划分，提取出与被关注用户兴趣相似的用户，未来希望能够设计出更为有效地针对关注关系进行划分的分类器. 此外，本论文只是在一个规模较小的 YouKu 数据集上进行测试，在未来的工作中，将在更大规模、更多样化的数据集上对算法性能进行验证.

参 考 文 献

[1] Davidson J, Liebald B, Liu J, et al. The YouTube video recommendation system//Proceedings of the 4th ACM Conference on Recommender Systems. Barcelona, Spain, 2010: 293-296

[2] Zhang Zi-Ke, Zhou Tao, Zhang Yi-Cheng. Personalized recommendation via integrated diffusion on user—item—tag tripartite graphs. Physica A: Statistical Mechanics and its Applications, 2010, 389(1): 179-186

[3] Baltrunas L, Amatriain X. Towards time-dependant recommendation based on implicit feedback//Proceedings of the Workshop on Context-Aware Recommender Systems (CARS 2009) in ACM Recsys. New York, USA, 2009: 1-5

[4] Wang Ying-Zi, Yuan Nicholas Jing, Lian De-Fu, et al. Regularity and conformity: Location prediction using heterogeneous mobility data//Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Sydney, Australia, 2015: 1275-1284

[5] Liu Feng-Kun, Lee Hong-Joo. Use of social network information to enhance collaborative filtering performance. Expert Systems with Applications, 2010, 37(7): 4772-4778

[6] Zhao Tong, McAuley Julian, King Irwin. Leveraging social connections to improve personalized ranking for collaborative

filtering//Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. Shanghai, China, 2014: 261-270

[7] Fouss F, Pirotte A, Renders J-M, Saerens M. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(3): 355-369

[8] Gori M, Pucci A. ItemRank: A random-walk based scoring algorithm for recommender engines//Proceedings of the 20th International Joint Conference on Artificial Intelligence. Hyderabad, India, 2007: 2766-2771

[9] Haveliwala T H. Topic-sensitive PageRank: A context-sensitive ranking algorithm for Web search. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(4): 784-796

[10] Xiang L, Yuan Q, Zhao S, et al. Temporal recommendation on graphs via long- and short-term preference fusion//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, USA, 2010: 723-732

[11] Lee Sang-Keun, Park Sung-Chan, Kahng Min-Suk, Lee Sang-Goo. PathRank: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems. Expert Systems with Applications, 2013, 40(2): 684-697

[12] Lee S, Kahng M, Lee S. Flexible recommendation using random walks on implicit feedback graph//Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication. Seoul, Korea, 2011: 1-6

[13] Chen Bi-Sheng, Wang Jing-Dong, Huang Qing-Hua, Mei Tao. Personalized video recommendation through tripartite graph propagation//Proceedings of the 20th ACM International Conference on Multimedia. Nara, Japan, 2012: 1133-1136

[14] Yu Yan, Qiu Guang-Hua. Algorithm of friend recommendation in online social networks based on local random walk. Systems Engineering, 2013, 31(2): 47-54(in Chinese)
(**俞琰，邱广华. 基于局部随机游走的在线社交网络朋友推荐算法. 系统工程**, 2013, 31(2): 47-54)

[15] Shardanand U, Maes P. Social information filtering: Algorithms for automating "word of mouth"//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Denver, USA, 1995: 210-217

[16] Bellogin A, Parapar J. Using graph partitioning techniques for neighbour selection in user-based collaborative filtering//Proceedings of the 6th ACM Conference on Recommender Systems. Dublin, Ireland, 2012: 213-216

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

[18] Jojic O, Shukla M, Bhosarekar N. A probabilistic definition of item similarity//Proceedings of the 5th ACM Conference on Recommender Systems. Chicago, USA, 2011: 229-236

[19] Cao Jie, et al. Hybrid collaborative filtering algorithm for bidirectional Web service recommendation. Knowledge and Information Systems, 2013, 36(3): 607-627

[20] Navgaran D Z, Moradi P, Akhlaghian F. Evolutionary based matrix factorization method for collaborative filtering systems//Proceedings of the 21st Iranian Conference on Electrical Engineering. Mashhad, Iran, 2013: 1-5

[21] Zhuang Yong, Chin Wei-Sheng, Juan Yu-Chin, Lin Chih-Jen. A fast parallel SGD for matrix factorization in shared memory systems//Proceedings of the 7th ACM Conference on Recommender systems. Hong Kong, China, 2013: 249-256

[22] Gemulla R, Nijkamp E, Haas P J, Sismanis Y. Large-scale matrix factorization with distributed stochastic gradient descent//Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego, USA, 2011: 69-77

[23] Takács G, Tikk D. Alternating least squares for personalized ranking//Proceedings of the 6th ACM Conference on Recommender Systems. Dublin, Ireland, 2012: 83-90

[24] Vozalis M G, Margaritis K G. Applying SVD on generalized item-based filtering. International Journal of Computer Science & Applications, 2006, 3(3): 27-51

[25] Zhang S, Wang W H, Ford J, Makedon F. Learning from incomplete ratings using non-negative matrix factorization//Proceedings of the 6th SIAM International Conference on Data Mining. Bethesda, USA, 2006: 549-553

[26] Salakhutdinov R, Mnih A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo//Proceedings of the 25th International Conference on Machine Learning. New York, USA, 2008: 880-887

[27] Massa P, Avesani P. Trust-aware recommender systems//Proceedings of the 2007 ACM Conference on Recommender Systems. Minnesota, USA, 2007: 17-24

[28] DuBois T, Golbeck J, Kleint J, Srinivasan A. Improving recommendation accuracy by clustering social networks with trust//Proceedings of the ACM Recsys Workshop Recommender Systems and the Social Web. New York, USA, 2009: 1-8

[29] Moradi P, Ahmadian S, Akhlaghian F. An effective trust-based recommendation method using a novel graph clustering algorithm. Physica A: Statistical Mechanics and its Applications, 2015, 436(15): 462-481

[30] Mirza B J, Keller B J, Ramakrishnan N. Studying recommendation algorithms by graph analysis. Journal of Intelligent Information Systems, 2003, 20(2): 131-160

[31] Duda R O, Hart P E, Stork D G. Pattern Classification. 2nd Edition. Hoboken: Wiley, 2000

[32] Haveliwala T, Kamvar S, Jeh G. An analytical comparison of approaches to personalizing pagerank. Stanford University, Stanford: Technique Report: 596, 2003

[33] Herlocker J L, Konstan J A, Terveen L G, Riedl J T. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 2004, 22(1): 5-53

[34] Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L. BPR: Bayesian personalized ranking from implicit feedback// Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence. Chicago, USA, 2009: 452-461

LIU Meng-Juan, born in 1979, Ph.D., associate professor. Her research interests include data mining and distributed computation.

WANG Wei, born in 1993, M. S. candidate. His research interests include data mining and recommendation.

LI Yang-Xi, born in 1987, M. S. candidate. His research interests include data mining and recommendation.

LUO Xu-Cheng, born in 1974, Ph. D., associate professor. His research interests include computer network and artificial intelligence.

QIN Zhi-Guang, born in 1956, Ph. D., professor. His research interests include information security and data mining.

Background

Information overload is one of the most critical problems, and personalized recommendation system is a powerful tool to solve this problem. In recent years, studies of recommendation algorithms have sprung up in the field of data mining. Taking both user behavior and follower relationship into consideration, they allow users who face a huge amount of information to find out information they are interested in. Now, a variety of recommendation algorithms have been used in different application scenarios, including recommendation with user behaviors, tags, time, location, context-aware, social network data, etc.

Collaborative filtering (CF) is the most successful technique in the design of recommender systems, where a user will be recommended with items that people with similar tastes and preferences in the past. Despite its success, the performance of CF is strongly limited by the scarcity of data resulted from: (1) the huge number of items that is far beyond user's ability to evaluate even a small fraction of them; (2) users do not inceptively wish to rate the purchased/viewed items. Besides the fundamental user-item relations, some accessorial information can be exploited to improve the algorithmic accuracy.

In our algorithm, different user behaviors and follower relationship are both considered when computing the nodes similarity for improving the recommendation performance. It is suitable for making recommendations in social network, such as video sharing sites, e-commerce system with subscription service and so on.

This work is supported by the National Natural Science Foundation of China under Grants (Nos. 61202445, 61272527, 61300090, 61133016). The projects aims to analyze the user behavior and establish the user's interest model, thus improve the recommendation performance of videos. We have successfully applied for two patents in the areas of video sharing and recommendation algorithm. AttentionRank algorithm supposed in this paper aims to solve the problems about modeling for multiple user behavior, combining graph recommendation algorithm and social network relationship in an effective way. In the end, we proceed to quantify how much the user behavior and interestingness is related. It hopefully achieves more research results in fields of building Multi-behaviors model, discovering the rule between social relation and video sharing and users' interest changes.