基于有向图分割的推荐算法①

黄 波, 严宣辉, 林建辉

(福建师范大学 数学与计算机科学学院, 福州 350007)

摘 要: 利用资源分配的原理提出一个基于有向图分割的推荐算法. 通过二部图网络结构与资源分配方法的结合, 建立了物品间关系的有向图, 再利用非对称非负矩阵分解(Asymmetric Nonnegative Matrix Factorization, ANMF)分割此有向图, 并将物品根据分割结果得出的物品间关联关系进行分类, 并以此设置物品间的关联权重, 最终实现对用户的 Top-N 物品推荐方案. 实验结果表明, 提出的算法提高了推荐准确率, 并且能在一定程度上提高推荐多样性, 降低推荐物品的流行性.

关键词: 推荐算法; 有向图; ANMF; 推荐权重

Recommendation Algorithm Based on the Partition of Directed Graph

HUANG Bo, YAN Xuan-Hui, LIN Jian-Hui

(School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China)

Abstract: By using the principle of resource allocation, we propose a recommendation algorithm which is based on the partition of directed graph. The items directed graph is established by combining with the bipartite graphs network structure and resource allocation method, and is partitioned by the method of Asymmetric Nonnegative Matrix Factorization. Then we classify items by the relationship between them, set connection weights between the items and implement a recommendation from the Top-N items to the user. Experimental results show that the proposed algorithm can improve the recommendation accuracy and the recommendation diversity, and reduce the popularity of recommendation to a certain extent.

Key words: recommendation algorithm; directed graph; ANMF; recommendation weight

1 引言

个性化推荐系统是指根据用户的历史行为, 建立用户兴趣模型, 去揭示用户可能感兴趣的物品, 从而向用户进行推荐. 一个完整的推荐系统通常由以下 3 个部分组成: 收集用户信息的行为记录模块, 分析用户喜好的模型分析模块和推荐算法模块. 其中个性化推荐算法是个性化推荐系统中最核心和最重要的部分, 其很大程度上决定了推荐效果的好坏和推荐系统的性能. 当前比较流行的个性化推荐算法包括协同过滤推荐算法[1-3]、基于内容的推荐算法[4]、基于网络结构的推荐算法[5-9]、基于矩阵分解的推荐算法[10-13]等.

协同过滤推荐算法是应用最广泛的推荐技术之一, 包括基于用户的协同过滤[1]和基于物品的协同过滤[2], 基于用户的协同过滤推荐算法先计算用户之间的相似度, 然后找出与目标用户相似度高的用户作为目标用户的最近邻居, 并基于这些最近邻居对目标用户进行推荐; 基于物品的协同过滤推荐算法从物品的角度进行分析, 寻找与目标物品相似的物品集合, 然后进行推荐. 协同过滤方法不受复杂的非结构化数据的影响, 但存在冷启动和数据稀疏等问题.

基于内容的推荐算法[4]提取用户喜爱的物品的特征信息, 找出与这些特征信息相似度最高的物品推荐给用户. 基于内容的推荐算法的推荐结果直观、容易理解, 但是对于复杂的非结构数据, 例如图像、视频和音乐等, 没有有效的特征提取方法, 且不能发现用户新的兴趣点和存在冷启动问题.

① 收稿时间:2015-04-08;收到修改稿时间:2015-05-28

基于用户物品组成的二分图网络结构, Haveliwala 提出了基于二分图网络结构的随机游走算法[5], 用户在二分图网络上经过多次随机游走后, 访问每个物品节点的概率会收敛到一个稳定的数值, 最终的推荐列表中物品的权重就是物品节点访问的概率; 李芳等[6]提出一种基于随机游走的多维数据推荐算法, 在随机游走算法的基础上加入用户的上下文信息, 能提高推荐算法的准确性; Zhou 等人[7-9]根据物质扩散的原理和二分图网络结构提出基于资源分配的推荐算法, 该方法能提高推荐的准确性和部分地解决了数据稀疏性的问题.

因为矩阵分解能将高维的数据矩阵进行降维处理, 解决算法的空间复杂度问题, 所以矩阵分解被应用到个性化推荐算法中. Daniel 等人提出 SVD 分解(奇异值分解)的方法[10], 对用户物品评分矩阵缺失值进行预测; Xiang 等人提出三维矩阵的分解方法[11], 添加了用户物品的上下文信息; Yehuda 等人在 SVD 模型的基础上引入了时间效应[12], 考虑了用户选择物品的时间顺序等; Chen 等人[13]提出利用正交的非负矩阵分解结合协同过滤算法, 提高了推荐结果的准确性. 但矩阵分解技术有以下两个缺点, 矩阵分解降低维度的同时导致了信息的损失, 效果难以保证; 矩阵分解过程需要不断迭代求解, 在一定程度上增加了算法的时间复杂性.

本文利用资源分配的原理提出一个基于有向图分割的推荐算法, 主要有以下两个方面的贡献: (1)在个性化推荐领域, 基于图的模型一般是指用户物品的二部图网络结构, 但大部分图的算法不能应用于二部图网络结构, 本文创新性地利用资源分配的原理, 建立了物品间关系的有向图, 拓宽了图的算法在推荐领域的应用; (2)本文提出利用具有聚类性质和强解释性的非对称非负矩阵分解方法来分割物品间关系的有向图, 从而将物品聚类, 解决了传统的 k-means 聚类方法对噪声数据敏感和难解释性的问题, 使推荐结果具有强解释性和高准确性.

本文提出的推荐算法的主要结构为: 利用资源分配原理, 建立了物品间关系的有向图, 再利用非对称非负矩阵分解(Asymmetric Nonnegative Matrix Factorization, ANMF)分割物品间关系的有向图, 将物品根据它们之间的关联关系进行分类, 在对用户进行 Top-N 物品推荐时, 提高同类物品之间的推荐权重. 为验证算法的性能, 我们在多个数据集上进行了实验和分析, 从实验结果看, 本文所提出的算法在推荐准确率、推荐多样性和降低推荐物品的流行性三个指标上都有较好的表现.

2 问题定义

2.1 利用资源分配的方法建立物品有向图

二分图网络结构是推荐系统中重要内容, 令 $G(V, E)$表示用户物品二分图, 其中 $V=V_U \cup V_I$ 由用户顶点集合 V_U和物品顶点集合 V_I组成. 对于数据集中每一个二元组(u , i),图中都有一个对应的边 $e(v_u,v_i)$,其中 $v_u \in V_U$ 是用户 u 对应的顶点, $v_i \in V_I$是物品 i 对应的顶点. 如图 1(A)所示, 是一个简单的用户物品二分图模型, 其中圆形节点表示用户, 方形节点表示物品, 圆形节点和方形节点之间的边表示用户选择过该物品.

基于以上二分图网络结构, Zhou 等人根据物质扩散的原理提出基于资源分配的推荐算法[7-9],该方法不但在算法推荐准确性上优于经典的协同过滤算法, 而且在算法复杂性上也明显低于经典的协同过滤算法. 资源分配是指每个物品将自己所拥有的资源通过二分图的边平均分配给选择过它的用户; 反过来, 每个用户又将自己所分到的资源再次通过二分图的边平均分配给他所选择过的物品. 如图 1 为资源分配过程的示意图, 第一步, 假设初始时, 物品 a 有 a 个资源, 由图 1(A)的二分图结构可知, 物品 a 只与用户 x 和 y 有连接边, 所以物品 a 将其所拥有的资源平均分配给用户 x 和 y, 即分别分配给 x、y 用户 $a/2$ 个资源, 同理, 分别将物品 b、c、d 和 e 的资源平均分配给与其有连接边的用户, 可得第一步分配结果如图 1(B)所示. 第二步, 由第一步可得此时用户 x 拥有 $a/2+c/3$ 个资源, 因为用户 x 只与物品 a 和 c 有连接边, 所以将用户 x 所拥有的资源平均分配给物品 a 和 c, 即分别分配给 a 和 c 物品 $a/4+c/6$ 个资源, 同理, 分别将用户 y、z 和 w 拥有的资源平均分配给与其有连接边的物品, 可得资源分配的最终结果如图 1(C)所示.

根据以上资源分配过程, 最终的资源分布可表示为(1)式所示的矩阵形式, 用符号表示即为 $A'=WA$, 其中(1)式中 5×5 的权重矩阵 W 是列标准化的, 且是不对称的, 第 i 行 j 列的元素表示第 j 个物品最终分配给第 i 个物品多少资源. 所以可以根据权重矩阵 W_{ij} 构造物品间关系的有向图, 第 i 行 j 列的元素即表示有向边 $e(v_i ,v_j)$的权重值, 两个物品间边的权重值越大, 表示

两个物品联系越紧密，否则相反.

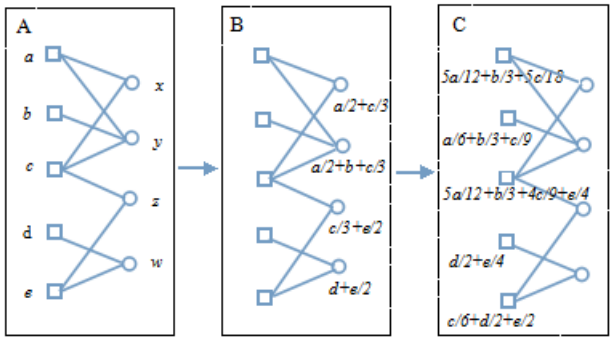

图 1　资源分配过程

$$\begin{pmatrix} a' \\ b' \\ c' \\ d' \\ e' \end{pmatrix} = \begin{pmatrix} 5/12 & 1/3 & 5/18 & 0 & 0 \\ 1/6 & 1/3 & 1/9 & 0 & 0 \\ 5/12 & 1/3 & 4/9 & 0 & 1/4 \\ 0 & 0 & 0 & 1/2 & 1/4 \\ 0 & 0 & 1/6 & 1/2 & 1/2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} \tag{1}$$

下面我们推导权重矩阵 W 的计算公式，假设用 $o_1, o_2, ..., o_n$ 表示物品和 $u_1, u_2, ..., u_m$ 表示用户.

1)资源分配过程中第一步是从物品到用户，将物品的资源平均分配给选择过它的用户，可用公式(2)表示，其中 $f(o_i)$表示物品的初始资源，$k(o_i)$表示物品 i 的流行度，a_{il}表示用户 l 是否有选择过物品 i，如果有选择过，$a_{il}=1$，否则 $a_{il}=0$.

$$f(u_l)=\sum_{i=1}^{n}\frac{a_{il}f(o_i)}{k(o_i)} \tag{2}$$

2）资源分配的第二步是从用户返回到物品，用户将所分得的资源平均分配给他选择过的物品，可用公式(3)表示，其中 $k(u_l)$表示用户 l 的兴趣度.

$$\begin{aligned} f'(o_i)&=\sum_{l=1}^{m}a_{il}f(u_l)/k(u_l) \\ &=\sum_{l=1}^{m}\frac{a_{il}}{k(u_l)}\sum_{j=1}^{n}\frac{a_{jl}f(o_j)}{k(o_j)} \end{aligned} \tag{3}$$

将公式(3)表示成公式(4)的形式，也即公式(1)的符号化

$$f'(o_i)=\sum_{j=1}^{n}w_{ij}f(o_j) \tag{4}$$

3）根据公式(3)和公式(4)，权重矩阵 W_{ij} 可表示为公式(5)所示

$$W_{ij}=\frac{1}{k(o_j)}\sum_{l=1}^{m}\frac{a_{il}a_{jl}}{k(u_l)} \tag{5}$$

从公式(5)权重矩阵 W_{ij} 的表示式子可以得出，物品 j 对物品 i 分配资源的值大小取决于：同时选择过物品 i 和物品 j 的用户数、同时选择过物品 i 和物品 j 的用户 l 的兴趣度、物品 j 的流行度. 当同时选择过物品 i 和物品 j 的用户数越多，同时选择过物品 i 和物品 j 的用户 l 的兴趣度越低和物品 j 的流行度越低，则物品 j 对物品 i 分配的资源值越大，所以构造物品间关系的有向图时，物品 j 到物品 i 的有向边的权值越大，否则相反.

2.2 分割有向图的非对称非负矩阵分解法

因为矩阵分解具有降维、可解释能力强以及具有和聚类方法相似的功能等，所以被广泛应用到个性化推荐算法中. 在通常的矩阵分解方法中，原始的大矩阵 V 被近似分解为低秩的 $V=WH$ 形式，因子 W 和 H 中的元素可为正或负，即使输入的初始矩阵元素是全为正的，迭代过程也不能保证分解结果的非负性. 从计算的观点看，分解结果中存在负值是正确的，但负值元素在实际问题中往往是没有意义的. 因此，探索矩阵的非负分解方法一直是很有意义的研究问题，正因如此，Lee 和 Seung 所提出的非负矩阵分解方法(Nonnegative Matrix Factorization , NMF)[14,15]得到人们的广泛关注，它克服了传统矩阵分解结果存在负值的问题. 并且 Ding 等人证明了非负矩阵分解(NMF)和 k-means 聚类方法有相似的关系和更容易解释聚类结果的特点[16]，而且分解方法具有收敛速度快、左右非负矩阵存储空间小的特点，所以它能将高维的数据矩阵降维处理，适合处理大规模数据.

因为由资源分配方法得到不对称的权重矩阵 W 表示物品间关系的有向图，所以本文运用非对称非负矩阵分解(Asymmetric Nonnegative Matrix Factorization ,ANMF)进行有向图分割，从而将物品进行分类.

问题可以形式化地描述为：已知一个非对称非负矩阵 W，求非负矩阵 X 和 S，使得：

$$W=XSX^T \tag{6}$$

其中 $X\in R_+^{n\times q}$ 表示矩阵 W 分解的结果，X_{ij} 表示第 i 个元素属于第 j 个类的可能性；$S\in R_+^{q\times q}$ 为非对称非负矩阵，用来标准化矩阵 X.

矩阵分解实际上是一个优化问题，如公式(7)所示，通过最小化度量矩阵分解偏差的损失函数来找到近似解. 假设将权重矩阵 W(非对称非负矩阵)分解为非负矩阵 X 和 S 的乘积 XSX^T，矩阵分解的效果可以用 W 与 XSX^T 的欧氏距离值来评判，两者间的欧氏距离值越小，表明矩阵分解的效果越好，否则相反.

$$\min_{X,S}\ell(W,XSX^T), s.t. X\in R_+^{n\times q}, S\in R_+^{q\times q} \tag{7}$$

解决这类问题可以采用梯度下降法, 即首先固定 S, 对目标函数针对 X 求偏导,可以得到矩阵 X 的迭代公式; 然后固定 X, 对目标函数针对 S 求偏导, 可以得到矩阵 S 的迭代公式. 根据得到的迭代公式分别对非负矩阵 X和 S进行若干次迭代, 可使得权重矩阵 W 分解为XSX^T的损失评价函数(7)收敛, 从而将权重矩阵 W 分解为非负矩阵 X 和 S 的乘积 XSX^T.

可以推导得出非负矩阵 X 和 S 的迭代公式, 如公式(8)和公式(9)所示, Wang 等[17]证明了公式(8)和(9)的正确性, 并且证明了使用公式(8)和(9)多次迭代能使得权重矩阵 W 分解为 XSX^T 的损失评价函数(7)收敛. 通过迭代所得的矩阵 X 的值表示对有向图分割的结果, X_{ij} 表示第 i 个物品属于第 j 个物品分类的概率.

$$X_{ik} \leftarrow X_{ik}\left(\frac{\left[W^TXS+WXS^T\right]_{ik}}{\left[XSX^TXS^T+XS^TX^TXS\right]_{ik}}\right)^{1/4} \tag{8}$$

$$S_{kl} \leftarrow S_{kl}\frac{\left[X^TWX\right]_{kl}}{\left[X^TXSX^TX\right]_{kl}} \tag{9}$$

公式(1)中的权重矩阵 W 为物品有向图的矩阵表示, 假设要将该有向图分割为 2 个子有向图结构, 我们使用迭代公式(8)和(9), 将有向图进行分割, 如图 2 所示为有向图分割的结果(连接箭头粗细表示相应的连接权重值大小), 可得非对称非负矩阵分解可以将有向图准确分割.

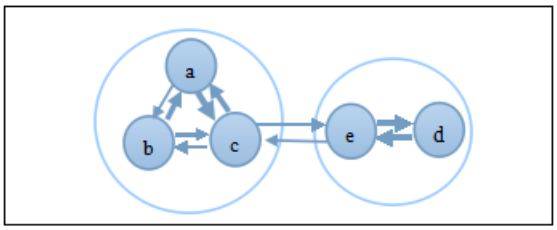

图 2 物品有向图分割

3 算法描述

本文算法的主要原理是利用资源分配的方法, 建立了物品间关系的有向图, 再利用非对称非负矩阵分解分割该有向图, 将物品根据它们之间的关联关系进行分类; 再根据公式(10)计算每个用户对每个物品的可能喜欢程度, 其中, 当物品 i 和 j 属于同一个类时, 资源推荐的权重 θ 的值为 λ, 当不属于同一个类时, 资源推荐的权重值为(1-λ), V_{il} 表示用户 l 对物品 i 的可能喜欢值, W_{ij} 表示物品 j 对物品 i 的资源分配值, R_{jl} 表示用户 l 对物品 j 的评分. 最终向用户推荐可能喜欢程度最高的 Top-N 个物品.

$$V_{il}=\sum_{j=1}^{n}\theta W_{ij}R_{jl} \tag{10}$$

根据以上算法的主要原理, 本文算法(WNBI, Weight Network-Based Inference)的详细步骤描述如下:

Input: 用户和物品的评分矩阵 $R_{n\times m}$, 目标用户 u_l

Output: 目标用户 u_l 的推荐列表

Step1(构造有向图): 根据公式(5), 计算得到物品间关系的有向图矩阵表示 $W_{n\times n}$.

Step2(分割有向图): 根据公式(8)和(9), 经迭代计算矩阵 $X_{n\times q}$ 和 $S_{q\times q}$ 使公式(7)收敛, 从而将该有向图分割成 q 个子有向图.

Step3(物品分类): 将物品根据步骤 2 分割的子图信息, 将物品分成 q 类.

Step4(计算用户对物品的喜欢值): 根据公式(10), 分别计算用户对全部物品的喜欢值.

Step5(Top-N 推荐): 将物品根据用户对其的喜欢值进行降序排列, 向用户推荐排列最前面的 N 个物品.

假设实验数据集中有 m 个用户, n 个物品, Step1 中构造有向图的时间复杂度为 $O(m\cdot n^2)$; Step2 中分割有向图的时间复杂度为 $O(itera\cdot n^2\cdot q)$,其中 $itera$ 为矩阵分解的迭代次数, q 为分割子图数; Step3 中物品分类的时间复杂度为 $O(n\cdot q)$; Step4 中计算用户对物品的喜欢值的时间复杂度为 $O(m\cdot n^2)$;Step5 中进行 Top-N 推荐的时间复杂度为 $O(n\cdot N)$. 综上所述, 算法的时间复杂度为 $O(itera\cdot n^2\cdot q)$,其中 n 为物品数,$itera$ 为矩阵分解的迭代次数, q 为分割子图数.

4 实验分析

4.1 实验数据集

在本文中, 我们采用个性化推荐中广泛使用的实验数据集 Movielens 和 EachMovie 来测试算法的准确性、多样性和流行性.

本文实验使用的 Movielens 数据集包括 943 个用户(即 m=943)对 1682 部电影(即 n=1682)的 10 万个评分记录. 其中, 每个用户至少评价过 20 部电影, 评分值为 1 到 5 之间的整数, 评分越高表示用户越喜欢该电影. 为了测试本文的算法, 我们假设用户对某电影的评分大等于 3 时, 表示他喜欢该电影. 原数据集有 10 万条评分记录, 其中有 82520 条评分值大等于 3, 所以我们将这些大等于 3 的 82520 条记录随机划分为两部分: 90%的记录为训练集, 10%的记录为测试集, 训练集假设为已知的信息, 测试集假设为未知的, 用来

进行预测评分.

本文实验使用的EachMovie数据集包括5000个用户(即 m=5000)对 1682 部电影(即 n=1682)的 93816 个评分记录, 评分值为 0、0.2、0.4、0.6、0.8、1, 其中评分值越大表示用户越喜欢该电影. 为了测试本文的算法, 我们假设用户对某电影的评分大等于 0.6 时, 表示他喜欢该电影. 在数据集中有 73884 条评分值大等于 0.6, 所以我们将这些大等于 0.6 的 73884 条记录随机划分为两部分: 90%的记录为训练集, 10%的记录为测试集, 训练集假设为已知的信息, 测试集假设为未知的, 用来进行预测评分.

4.2 参数设置

根据本文算法的原理, 算法的实验结果会随着参数 q(分割子图数)和参数 λ(同类物品之间资源推荐权重)的设置值不同而变化, 我们首先通过准确性指标来设定最佳的参数 q 和参数 λ. 本文以下给出的全部实验结果, 是经过 20 次独立实验所得结果的平均值, 且每次分割物品间关系的有向图是经过 500 次迭代所得.

本文采用命中率(hit)来测试算法的准确性, 如公式(11)所示, H 表示测试的命中数, T 表示测试数. 命中率越高表示算法的准确性越高, 否则相反.

$$hit=\frac{H}{T} \tag{11}$$

为了得出使推荐准确率最高的参数 λ 值, 我们先设置参数 q=8, 当 Top-N 推荐列表长度为 10 时, 在实验数据集Movielens和EachMovie中, 算法的命中率与参数 λ 的关系分别如图 3 和图 4 所示.

从图 3 和图 4 观察可知, 当参数 λ 小于 0.8 时, 算法推荐命中率随着参数 λ 的增大而提高, 当参数 λ 为 0.8 时, 算法的推荐命中率达到最高, 当参数 λ 大于 0.8 时, 算法的推荐命中率随着参数 λ 的增大而降低, 所以可得当参数 λ 为0.8时, 本文提出的算法的推荐准确性最好. 同时验证了同类物品之间资源推荐更可靠, 所以提高同类物品之间资源的推荐权重, 能提高算法的推荐准确性; 但又由于数据的稀疏性, 所以不能仅仅只利用同类物品之间的资源推荐, 不然当同类物品都没有被用户选择过的情况, 就会导致冷启动问题.

接下来, 为了得出最佳分割子图数, 我们设置同类物品之间资源推荐权重 λ 值为 0.8, 考虑到 q 值如果取太大, 会导致分割子图过细, 一些相似的物品被划分到不同类中; 反之则导致分割子图过粗, 不能将不同类的物品区分开. 所以我们假设 q 取值范围为 5 到 15 之间, 当 Top-N 推荐列表长度为 10 时, 在实验数据集 Movielens 和 EachMovie 中, 算法命中率与参数 q 的关系分别如图 5 和图 6 所示.

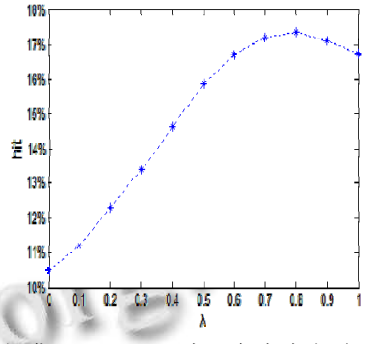

图 3　数据集 Movielens 中, 命中率与参数 λ 关系

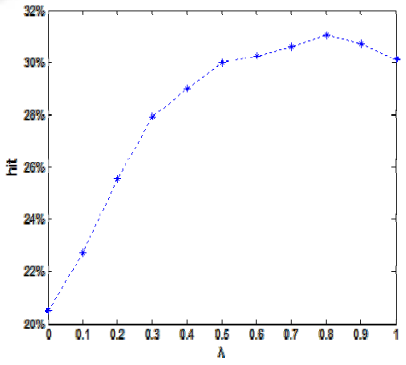

图 4　数据集 EachMovie 中, 命中率与参数 λ 关系

从图 5 和图 6 观察可知, 算法的推荐命中率会随着参数 q(分割子图数)改变而改变. 由图 5 得, 在数据集 Movielens 中, 当分割子图数为 10 时, 算法的推荐命中率最高; 由图 6 得, 在数据集 EachMovie 中, 当分割子图为 9 时, 算法的推荐命中率最高. 两个数据集分割子图数的不同, 表明本文提出的利用非对称非负矩阵分割有向图的方法能根据不同数据集的特点准确将物品分类.

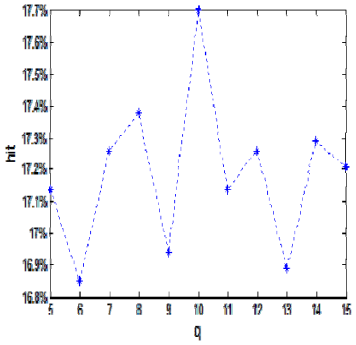

图 5　数据集 Movielens 中, 命中率与参数 q 的关系

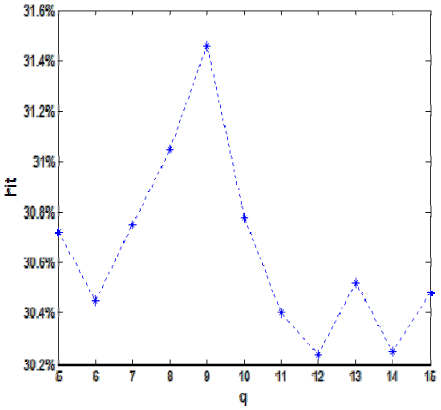

图 6 数据集 Eachmovie 中, 命中率与参数 q 的关系

从以上实验可知, 在数据集 Movielens 中, 当设置参数 λ 为 0.8, 参数 q 为 10 时, 算法的准确性最高, 所以在下一节对比实验中, 算法在数据集 Movielens 中的测试时, 设置参数 λ 为 0.8, 参数 q 为 10; 同理, 算法在数据集 EachMovie 中测试时, 设置参数 λ 为 0.8, 参数 q 为 9.

4.3 对比实验

为了验证本文算法的有效性, 将本文算法(WNBI)的推荐结果与参考文献 7 中提到的 GRM、UserCF、ItemCF 和 NBI 推荐算法, 分别在推荐准确性、多样性和流行性三个评价指标上进行比较. 其中, GRM 推荐算法是向每个目标用户都推荐最热门的物品; UserCF 推荐算法是指基于用户的协同过滤推荐算法, 算法先计算用户之间的相似度, 然后找出与目标用户相似度高的用户作为目标用户的最近邻居, 并基于这些最近邻居对目标用户进行推荐; ItemCF 推荐算法是指基于物品的协同过滤推荐算法, 算法从物品的角度进行分析, 寻找与目标物品相似的物品集合, 然后进行推荐; NBI 推荐算法是利用用户-物品二部图进行资源分配的推荐算法.

4.3.1 准确性

根据上一节参数设置实验可知, 在数据集 Movielens 中, 我们设置同类物品之间资源推荐权重 λ 值为 0.8 和分割子图数 q 为 10; 在数据集 EachMovie 中, 设置同类物品之间资源推荐权重 λ 值为 0.8 和分割子图数 q 为 9. 采用公式(11)所示的推荐命中率表示推荐算法的推荐准确性, 将本文算法(WNBI)的推荐准确性与 GRM、UserCF、ItemCF 和 NBI 推荐算法的推荐准确性进行对比, 如表格 1 所示.

表 1 本文算法与其它推荐算法的推荐准确性比较

数据集	算法名称	10	20	50	100
Movielens	GRM	10.3%	16.9%	31.1%	45.2%
	UserCF	14.1%	21.6%	37.0%	51.0%
	ItemCF	14.9%	22.8%	39.1%	52.5%
	NBI	16.2%	24.8%	41.2%	55.9%
	WNBI	17.7%	27.3%	43.7%	59.0%
EachMovie	GRM	24.7%	35.5%	54.7%	71.9%
	UserCF	27.3%	37.9%	58.3%	74.2%
	ItemCF	28.4%	38.7%	59.0%	76.6%
	NBI	30.1%	42.0%	62.5%	78.7%
	WNBI	31.5%	45.5%	66.8%	81.8%

从表 1 可以看出, 在数据集 Movielens、EachMovie 中, 当参数设置最优值以及 Top-N 推荐列表为 10,20,50 和 100 时, 本文提出的推荐算法的推荐准确性比其它经典的推荐算法有了较大程度的提高.

4.3.2 多样性和流行性

推荐算法的多样性评价可以通过用户推荐列表之间的平均汉明距离来表示, 对于任意的两个用户 u_i 和 u_j,其推荐列表之间的汉明距离为公式(12)所示.

$$H_{ij}=1-\frac{Q_{ij}}{L} \tag{12}$$

其中, L 表示推荐列表长度, Q_{ij} 表示用户 u_i 和 u_j 在长度为 L 的推荐列表中相同的物品个数; 计算出任意两个用户之间的汉明距离, 然后计算其平均值 H, 用 H 衡量算法的多样性. H 的数值范围为 0 到 1, H 越大表示算法的多样性越好, 否则相反.

在数据集 Movielens 中, 我们设置同类物品之间资源推荐权重 λ 值为 0.8 和分割子图数 q 为 10 时; 在数据集 EachMovie 中, 设置同类物品之间资源推荐权重 λ 值为 0.8 和分割子图数 q 为 9. 将本文算法(WNBI)的推荐多样性与 GRM、UserCF、ItemCF 和 NBI 推荐算法的推荐多样性进行对比, 如表格 2 所示.

表 2 本文算法与其它推荐算法的推荐多样性比较

数据集	算法名称	10	50	100
Movielens	GRM	0.508	0.397	0.337
	UserCF	0.684	0.549	0.441
	ItemCF	0.742	0.669	0.554
	NBI	0.733	0.618	0.520
	WNBI	0.745	0.675	0.596
EachMovie	GRM	0.299	0.187	0.136
	UserCF	0.376	0.294	0.223
	ItemCF	0.512	0.445	0.408
	NBI	0.495	0.399	0.383
	WNBI	0.563	0.484	0.468

从表 2 可以看出, 在数据集 Movielens 和 EachMovie 中, 当 Top-N 推荐列表为 10,50 和 100 时, 本文提出的推荐算法的推荐多样性比其它经典的推荐算法有了较大程度的提高.

用推荐列表中的 L 个物品的平均流行度<K>来评价算法所推荐物品的流行性. 平均流行度越小, 表示即使不是非常流行的物品也能被推荐, 说明算法的推荐效果越好, 因为流行度高的物品, 即使推荐系统不推荐, 用户也能通过其它渠道比偏冷门的物品更容易获得物品的相关信息, 所以推荐流行度相对较低的品, 能更好地发挥推荐系统的作用.

在数据集 Movielens 中, 我们设置同类物品之间资源推荐权重 λ 值为 0.8 和分割子图数 q 为 10 时; 在数据集 EachMovie 中, 设置同类物品之间资源推荐权重 λ 值为 0.8 和分割子图数 q 为 9. 将本文算法(WNBI)的推荐流行性的实验结果与 GRM、UserCF、ItemCF 和 NBI 推荐算法的推荐流行性进行对比, 如表 3 所示.

表 3　本文算法与其它推荐算法的推荐物品流行性比较

数据集	算法名称	10	50	100
Movielens	GRM	353.50	258.00	214.09
	UserCF	334.79	246.23	204.13
	ItemCF	303.96	214.46	189.76
	NBI	315.05	233.43	193.68
	WNBI	301.61	212.21	186.90
EachMovie	GRM	854.14	611.33	481.22
	UserCF	805.45	553.56	432.96
	ItemCF	748.90	498.76	377.45
	NBI	775.81	527.17	383.61
	WNBI	737.38	482.85	369.16

从表 3 可以看出, 在数据集 Movielens 和 EachMov- ie 中, 当 Top-N 推荐列表为 10,50 和 100 时, 本文提出的推荐算法的推荐流行性比其它经典的推荐算法的推荐流行性有一定的降低.

5　结论

本文利用资源分配的原理, 建立了物品间关系的有向图, 再利用非对称非负矩阵分解(Asymmetric Nonnegative Matrix Factorization, ANMF)分割物品间关系的有向图, 将物品根据它们之间的关联关系进行分类; 当对用户进行推荐时, 赋予同类物品之间的推荐权重适当大于不同类物品之间的推荐权重. 由实验分析可知, 当物品间关系的有向图被分割成的子图个数为最佳时(数据集 Movielens 中为 10, 数据集 EachMovie 中为 9), 并且赋予同类物品之间的推荐权重与不同类物品之间的推荐权重之比为 4:1 时, 本文提出的算法能取得最优的推荐准确率, 并且该推荐准确率大于 GRM(最热门推荐)、协同过滤和 NBI 推荐算法的推荐准确率, 并且本文分析了在取得最优推荐准确率的参数设置下, 该算法的推荐多样性和推荐物品的流行性, 实验表明本文提出的算法不但能提高推荐准确率, 并且能在一定程度上提高算法的推荐多样性, 降低推荐物品的流行性.

在个性化推荐领域, 基于图的模型一般是指用户物品的二部图网络结构, 本文创新性地利用资源分配的原理, 建立了物品间关系的有向图, 拓宽了图的算法在推荐领域的应用. 并且由于非负矩阵分解具有聚类性质和强解释性, 本文使用非对称非负矩阵分解来分割物品间关系的有向图, 克服了 k-means 聚类方法的难解释性, 相信本文方法的提出对推荐算法的发展有一定的意义. 在接下去的工作中, 将进一步研究能否建立用户间关系的有向图, 研究其能否用于进一步改进推荐算法.

参考文献

1 HerLoc0ker LJ, Konstan AJ, Riedl TJ. Empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information Retrieval, 2002,5(4):287–310.

2 Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms. Proc. of the 10th International Conference on World Wide Web. New York: ACM Press, 2001: 285–295.

3 Chen YL, Cheng LC. A novel collaborative filtering approaach for recommending ranked items. Expert Systems with Applications, 2008,34(4):2396–2405.

4 Balabanovic M, Shoham Y. content-based collaborative recommendation. Communications of the ACM, 1997,40(3): 66–72.

5 Haveliwala TH. Topic-sensitive PageRank. Proc. of the Eleventh International World Wide Web Conference, 2002, 101(3):494–502.

6 李芳,李永进.一种基于随机游走的多维数据推荐算法.计算机科学,2013,40(11):304–307.

7 Zhou T, Ren J, Medo M, et a1. Bipartite network projection and personal Recommendation. Physical Review.E, 2007,76

(4):046115.

8 Zhou T, Jiang LL, Su RQ, et a1. Effect of initial configuration on network based recommendation. Europhysics Letters, 2008,81(5):58004.

9 Zhou T, Su RQ, Liu RR, et a1. Ultra accurate personal recommendation via eliminating redundant correlations. Physics, 2009,2(5):4127.

10 Daniel B, Michael J. Learning Collaborative Information Filters. Proceedings of the Fifteenth International Conference on Machine Learning, San Francisco: Morgan Kaufmann Publishers Inc press, 1998:46–54.

11 Xiang L, Yang Q. Timedependent models in collaborative filtering based recommender system. ACM International Conference on Web Intelligence and Intelligent Agent Technology. New York: ACM Press, 2009: 450–457.

12 Yehuda K. Collaborative Filtering with Temporal Dynamics. KDD, 2010,53(4):89–97.

13 Chen G, Wang F, Zhang CS. Collaborative Filtering using orthogonal nonnegative matrix tri-factorization. Information Processing and Management, 2009,45(3):368–379.

14 Lee DD, Seung HS. Algorithms for non-negative matrix factorization. Advances in neural information processing system. 2001,13:556–562.

15 Lee DD, Seung HS. Learning the parts of objects by non-negative matrix. Factorization. Nature, 1999,401:788– 791.

16 Ding C, He XF, Simon HD. On the equivalence of nonnegative matrix factorization and spectral clustering. Siam International Conference on Data Mining. 2005: 606 –610.

17 Wang F, Li T, Wang X, et al. Community discovery using nonnegative matrix. Factorization. Data Mining and Knowledge Discovery, 2011,22(3):493–512.