·人工智能及识别技术· 文章编号：1000-3428(2019)07-0222-07 文献标志码：A 中图分类号：TP301

一种自适应的混合协同过滤推荐算法

杨佳莉[1]，李直旭[1,3]，许佳捷[1]，赵朋朋[1]，赵 雷[1]，周晓方[1,2]

（1. 苏州大学 计算机科学与技术学院，江苏 苏州 215006；
2. 昆士兰大学 信息技术与电子工程学院，澳大利亚 布里斯班 4067；
3. 广东省大数据分析与处理重点实验室，广州 510006）

摘 要： 为解决协同过滤算法在处理数据量较大时存在推荐效率低的问题，提出一种自适应混合协同推荐算法。根据待推荐用户活跃度和目标物品新鲜度调节模型权重，基于张量分解计算物品间的相似度，通过短路径枚举叠加生成预测结果。实验结果表明，与 CBCF 算法相比，该算法推荐准确率提高了 28.6% 。

关键词： 推荐系统；张量分解；协同过滤算法；自适应混合；短路径

中文引用格式： 杨佳莉，李直旭，许佳捷，等. 一种自适应的混合协同过滤推荐算法[J]. 计算机工程，2019，45(7)：222-228.
英文引用格式： YANG Jiali, LI Zhixu, XU Jiajie, et al. An adaptive hybrid collaborative filtering recommendation algorithm[J]. Computer Engineering, 2019, 45(7): 222-228.

An Adaptive Hybrid Collaborative Filtering Recommendation Algorithm

YANG Jiali[1], LI Zhixu[1,3], XU Jiajie[1], ZHAO Pengpeng[1], ZHAO Lei[1], ZHOU Xiaofang[1,2]

(1. College of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China;
2. School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane 4067, Australia;
3. Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, China)

【Abstract】 In order to solve the problem that the collaborative filtering algorithm has low recommendation efficiency when processing a large amount of data, an adaptive hybrid collaborative recommendation algorithm is proposed. The algorithm adjusts the weight of the model based on the to-be-recommended user activity and the freshness of target items. The similarity between items is calculated based on the tensor decomposition. The prediction result is generated based on short path enumeration superposition. Experimental results show that compared with the CBCF algorithm, the proposed algorithm improves the recommendation accuracy by 28.6% .

【Key words】 recommendation system; tensor decomposition; collaborative filtering algorithm; adaptive hybrid; short path

DOI： 10.19678/j.issn.1000-3428.0051041

0 概述

目前，推荐算法可分为基于内容[1-2]和协同过滤[3-4]2 种类型。基于内容的推荐算法给用户推荐已购物品相似度较大的物品，但仅考虑到物品内容在一维空间中的相似度，导致推荐效果较差。协同过滤推荐算法是目前主流的推荐算法，如 k 近邻算法和随机游走算法[5-7]，但无法适用于用户历史记录较少的情况，即无法解决数据稀疏性问题，对没有历史记录的用户更加难以推荐，且在数据集规模较大时，计算开销大。

针对上述问题，本文提出一种自适应混合协同过滤推荐（Adaptive Hybrid Collaborative Filtering recommendation，AHCF）算法。运用基于张量的相似度计算物品间的相似度，以目标用户为起点，通过多步路径跳转找到候选物品，使用贝叶斯概率计算评分得到物品排名，从而得到物品推荐结果。

1 相关工作

个性化物品推荐是推荐系统的核心问题。物品推荐算法可分为基于内容的推荐算法和协同过滤推荐算法。协同过滤推荐算法主要通过考察用户对物品的购买记录、评论信息和标签信息等内容来计算

基金项目： 国家自然科学基金(61632016)；江苏省高等学校自然科学研究重大项目(17KJA520003)。
作者简介： 杨佳莉(1993—)，女，硕士研究生，主研方向为推荐系统、机器学习；李直旭(通信作者)、许佳捷、赵朋朋，副教授；赵 雷、周晓方，教授。
收稿日期： 2018-04-02 **修回日期：** 2018-06-05 **E-mail：** zhixuli@suda.edu.cn

用户与用户之间及物品与物品之间的相似度,从而更好地为用户推荐相关商品[8]。然而,用户之间或者物品之间相似性依赖历史记录。当一个系统刚开始启动时,协同过滤算法存在稀疏性问题,因此结合协同过滤推荐算法与基于内容的推荐算法被提出。一种方法是分别用多种推荐方法得到推荐结果,再采用某种算法将其混合。如投票机制[9-10]。一种方法是以某种推荐策略为框架混合另外的推荐策略,如在协同过滤推荐的框架内混合基于内容的推荐[11]。然而,上述混合方式都不能缓解系统稀疏性的问题。

基于内容的推荐算法是将物品表示成词向量,并度量物品间的相似度,给用户推荐最可能感兴趣的物品。然而一维向量表示方法忽略了内容各个属性间的关联关系。例如,在电影数据集上,导演詹姆斯·卡梅隆总和科幻性电影联系在一起,当一部电影由詹姆斯·卡梅隆导演,即使没有指明其类型,但如果有关联关系,仍可以推断这是一部科幻电影。而这种关联关系不能通过一维相似度计算。张量分解是一种特征抽取方法。文献[8]对用户物品标签三元组构造张量,使用 tucker 分解求解得到用户的抽象表示,核心张量蕴含了各属性间的关联,且考虑了物品与标签的共现关系。因此可利用张量分解来挖掘内容之间的内部关系。

协同推荐方法通常考虑建立一个由用户和物品构成的二部图[3,12],并在此二部图上进行用户与物品之间的相似度计算。文献[13]提出经典 SimRank 算法,其基本思想是"跟相似物品相关的物品也是相似的",它定义了节点间相似度度量方法,然后用迭代方法求得最终所有节点的相似度。文献[5]提出 3 种基于随机游走的评分算法,分别计算从一个用户出发经过三步或五步游走后得到的物品排名信息。对于加入标签信息的三部图上的协同过滤算法也有大量的研究。比如文献[14]将二部图上的基于传播模型的推荐算法拓展到三部图中,定义一个简单的线性叠加来结合用户与物品、物品与标签二部图的传播,该集成方法能够提供更准确、多样和新奇的推荐。文献[15]提出基于标签的物品推荐算法,通过先降维再融合,将用户、物品和标签的三维关系划分为 3 个二维关系,最后再将这些关系进行融合。然而,基于图结构的协同过滤算法的缺点在于数据量较大时对内存的要求较高且时间效率较差,如 SimRank、随机游走[6]等都需要足够的内存空间和计算资源以处理大量节点的邻接矩阵。使用抽样的方法能够解决效率问题[5],但抽样会导致推荐质量的下降。文献[16]提出一种从目标用户出发的三步路径法,使用贝叶斯范式计算候选物品评分,降低了计算开销。本文将该方法拓展到三部图上,其有较好的推荐性能。

2　自适应混合协同推荐算法

问题定义如下:用户集合 $U=\{u_1,u_2,\cdots,u_m\}$,物品集合 $I=\{i_1,i_2,\cdots,i_n\}$(每个物品都具有特定属性)和标签集合 $T=\{t_1,t_2,\cdots,t_k\}$ 构成三部图,如图 1 所示,其多元关系包括:用户打标签的行为,用户对物品的购买记录以及物品被标签标注的行为。推荐任务是基于三部图上的多元信息为每个用户个性化推荐可能感兴趣的物品。自适应混合协同推荐模型框架如图 2 所示。

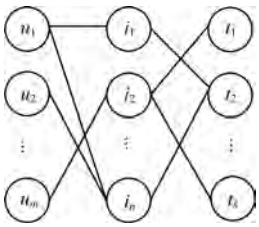

图 1　用户、标签及物品构成的三部图

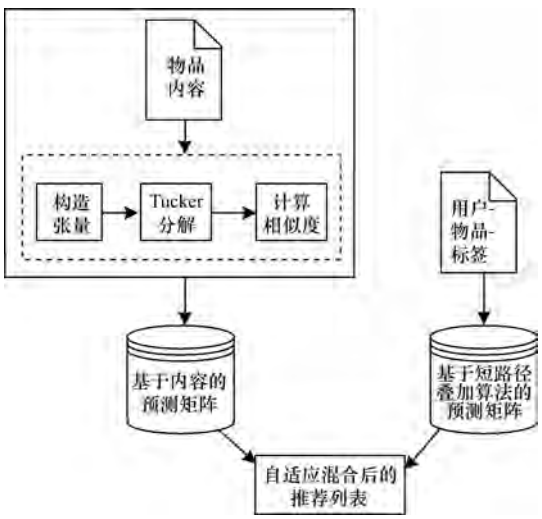

图 2　自适应混合协同推荐模型原理框架

2.1　自适应调整的加权混合模型

协同过滤方法对于评论和购买记录较少的不活跃用户以及刚进入系统的新物品推荐效果较差。为此,引入基于内容的物品相关性计算来自适应地缓解冷启动问题。自适应主要指算法能根据目标用户的活跃度和推荐物品的新鲜度来自动调节基于内容的推荐在混合模型中的权重。用户的活跃度和物品的新鲜度定义如下:

定义 1(用户 u 的活跃度)一个用户 u 的活跃度反映了用户的活跃程度,历史记录越多,活跃度越高,其定义为:

$$\eta(u)=\frac{2}{1+\mathrm{e}^{-N_r(u)/\alpha}}-1 \tag{1}$$

其中,$N_r(u)$ 表示用户 u 的物品历史记录个数,α 是一个调节参数,用于调整用户记录数对活跃度的影响程度。

定义 2(物品 i 的新鲜度)一件物品 i 的新鲜度

反映了该物品受用户的关注程度,它被购买记录越多,新鲜度越低,其定义为:

$$\varphi(i)=2-\frac{2}{1+\mathrm{e}^{-N_r(i)/\beta}} \tag{2}$$

其中,$N_r(i)$表示物品 i 的被交易记录个数,β 也是一个调节参数,用于调整物品被交易记录数对物品新鲜度的影响程度,其由数据集确定。

对一个物品 i 而言,将其推荐给用户 u 的得分可计算如下:

$$r(u,i)=\omega(u,i)\cdot r_t(u,i)+(1/\omega(u,i))\cdot r_c(u,i) \tag{3}$$

其中,$r_t(u,i)$为基于内容的推荐将物品 i 推荐给用户 u 的推荐得分,$r_c(u,i)$为协同推荐将物品 i 推荐给用户 u 的推荐得分,$\omega(u,i)$是混合基于内容的推荐算法的自适应权重。

$\omega(u,i)$主要考虑用户的活跃度和物品的新鲜度以及它们的组合关系对权重值的影响。随着用户的活跃度提高或者物品新鲜度的降低,基于内容的推荐评分所占权重 $\omega(u,i)$的分数逐渐降低,其计算公式如下:

$$\omega(u,i)=\frac{\gamma\cdot\varphi(i)}{\eta(u)} \tag{4}$$

其中,γ 是一个调整因子,本文设为 1。

2.2 基于张量的内容推荐分析

在混合模型中,基于内容的关联度推荐是对协同过滤方法的重要补充,因此挖掘物品的内容信息十分关键。文献[8]方法仅基于词向量空间模型,把所有关键词映射到一个向量空间,且每一个物品可以表示成 $i_i=\{w_{1i},w_{2i},\cdots,w_{ni}\}$,那么物品 i_i 与 i_j 的相似度可以用余弦相似度来度量。但这种一维空间上的词向量表达不仅浪费空间,且无法解析和发现物品本身的不同特征之间的关系。本文使用一种基于张量分解的方法来对物品之间的相似性进行计算,将物品中包含的多维共现信息表示成张量,然后用张量分解方法提取物品的抽象特征。具体来说,首先将物品表示成一个 N 阶张量 $\boldsymbol{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$,每一维表示一个属性。例如电影可以有类型、导演、演员等多个属性的表示。I_i 表示属性 i 的所有取值的集合大小。张量 $\boldsymbol{X}$ 与矩阵 $\boldsymbol{U}\in\mathbb{R}^{J\times I_i}$的 mode-i 乘法用 $\boldsymbol{X}\times_i\boldsymbol{U}$ 表示,且结果是一个 $I_1\times I_2\times\cdots\times I_{i-1}\times J\times I_{i+1}\times\cdots\times I_N$ 大小的张量。本文使用 Tucker 分解方法[17]将物品张量 $\boldsymbol{X}$ 分解成一个与原张量维度相同的核心张量 $\boldsymbol{\vartheta}\in\mathbb{R}^{J_1\times J_2\times\cdots\times J_N}$和每个维度上的因子矩阵 $\boldsymbol{A}^{(i)}\in\mathbb{R}^{I_i\times J_i}$进行 mode-i 矩阵乘法近似表达,即:

$$\boldsymbol{X}\approx\boldsymbol{\vartheta}\times_1\boldsymbol{A}^{(1)}\times_2\boldsymbol{A}^{(2)}\times\cdots\times_N\boldsymbol{A}^{(N)} \tag{5}$$

核心张量是一个和原张量维度一样但是每一维上的特征小得多的张量,它代表了各个属性的隐式因子间的关系,每个维度上的因子矩阵 $\boldsymbol{A}^{(i)}$代表该属性上的特征与隐式因子间的关系。例如,电影网络可以表示成 1 个 4 阶张量,$\boldsymbol{X}\in\mathbb{R}^{I_1\times I_2\times I_3\times I_4}$,其中每个维度分别表示电影、导演、演员、类型。经过 Tucker 分解后,核心张量是一个 $d\times d\times d\times d$ 的张量,且 d 是比较小的常数,而每一个因子矩阵是 $d\times I_n$ 大小的矩阵,表示张量在某一个属性维度上的子空间。假设 $\boldsymbol{A}^{(1)}$代表了电影子空间因子矩阵,那么用 $\boldsymbol{\vartheta}\times_1\boldsymbol{A}^{(1)}$来表示所有电影的特征张量,它是一个 $I_1\times d\times d\times d$ 大小的张量,每一个电影可以用 $d\times d\times d$ 的特征张量表示。假设有电影 $\boldsymbol{T}_1$ 和 $\boldsymbol{T}_2$,那么 $\boldsymbol{T}_1$ 和 $\boldsymbol{T}_2$ 的相似度为:

$$sim(\boldsymbol{T}_1,\boldsymbol{T}_2)=\frac{\sum_{i,j,k}t_{ijk}^1\cdot t_{ijk}^2}{\|\boldsymbol{T}_1\|\ \|\boldsymbol{T}_2\|} \tag{6}$$

其中,t_{ijk}表示张量中第(i,j,k)个元素,从而可以计算出所有物品之间的相似度。

2.3 三部图上的短路径枚举算法

本文数据集有用户、物品以及标签 3 种实体,且 3 种实体间有不同的关系连接。为解决协同推荐方法的效率随着用户及物品规模的增长而降低的问题,本文采用一种短路径枚举叠加方法。该方法以目标用户为起点,通过多步路径跳转找到候选物品,使用贝叶斯概率计算评分得到物品排名从而进行推荐,大幅提高了基于图模型的推荐算法效率且保证了推荐质量。三部图模型涉及的关系较多,每个用户的邻居节点包括物品和标签,用户之间的连接可以通过物品或者标签,也可以同时通过一个物品标签对。因此,本节将讨论并比较包含多类型关系的三部图之上的 2 种短路径枚举方法。图 3 所示为三部图上的多种短路径枚举法的拓展图模型。

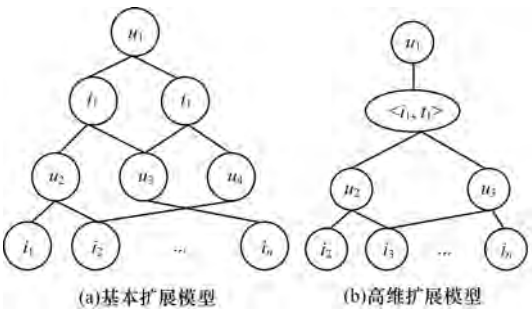

图 3 三部图上的多种短路径枚举法的拓展图模型

从图 3(a)可以看出,由目标用户出发,根据该用户的邻接节点,包括选择过的物品和打过的标签,并找到共享这些物品或标签的用户,再将这些用户的所有邻接物品找出来。对于每一条路径上的目标物品,用贝塔分布来计算他们的评分,对于目标物品相同的路径,叠加该目标物品的得分。物品的评分是该物品的正面评分的概率。假设物品 v 的正面概

率 θ_v 的分布服从贝塔分布,即 $\theta_v \sim Beta(a,b)$,那么 $\frac{a}{a+b}$ 即为所有物品正面概率的平均值,而对于某一个物品 v 来说,它的正面率又与其本身的正面评分与负面评分相关,假设该物品正面和负面评分的分别个数有 $|R_v^+|$ 和 $|R_v^-|$,并且为避免该物品没有评分的情况,调整该物品的正面率如式(7)所示。

$$p_v = \frac{a + |R_v^+| + 1}{a + b + |R_v^+| + |R_v^-| + 1} \tag{7}$$

通过上述方法,可以统计图信息,根据候选物品的路径评分叠加得出所有待推荐物品的得分。对于用户 u 的从物品 v 出发到用户 w 再到物品 x 一条路径 $\rho(u,v,w,x)$,其得分为:$S_{\rho(u,v,w,x)} = P_x$。对于经过标签的路径同理可求。

分享共同物品或者共同标签的用户数量实际上是很普遍的,将他们选取为路径中的节点无法判断用户与目标用户是真的兴趣相投还是只是因为偶然性。因此,考虑一种共享物品标签对的路径方法,从而提出第 2 种高维扩展方法。如图 3(b)所示,将路径中的第 1 步到达节点更改为物品与标签二元组,可以认为共享物品和标签的二元组信息的用户是更相似的,这种物品与标签共现的信息很大程度上去除了大量偶然相关用户,提高推荐物品的相关性。这种情况下第 2 个节点的评分与基本扩展类似,从而该条路径的评分是终点物品的评分。假设用户 u 通过 $<i_k,t_i>$ 二元组找到用户 v,计算用户 v 的历史物品 i_l 的得分为:$S_{\rho(u,<i_k,t_i>,v,i_l)} = P_{i_l}$。

图 4 描述了短路径枚举方法的示意图,$\boldsymbol{A}$ 代表用户与物品标签的邻接矩阵,对于用户与物品(标签)有交互的点设置为 1,没有交互的设置为 0,$\boldsymbol{A}^{\mathrm{T}}$ 代表 $\boldsymbol{A}$ 的转置,$\boldsymbol{P}$ 是用户与物品的交互矩阵,矩阵相乘之后即为最终的预测矩阵 $\boldsymbol{R}$。

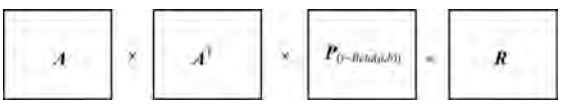

图 4　短路径枚举法算法说明

2.4　算法描述

本文算法伪代码具体如下:

输入　物品的内容信息,用户物品评分矩阵 $\boldsymbol{M}$,用户物品邻接矩阵 $\boldsymbol{A}$

输出　所有用户的推荐结果 r

```
//生成基于内容的推荐评分矩阵
1. 构造物品的内容张量 T;
2. ϑ ×1 A(1) ×2 A(2) ×⋯ ×N A(N) = Tucker(T);
3. item = ϑ ×1 A(1);
4. itemSimMatrix = calSimMatrix(item);
5. canMatrixByCon = calCanList(itemSimMatrix,M);
//生成短路径枚举法推荐评分矩阵
6. itemScore = cal(M);
7. P = construct(M,itemScore);
8. canMatrixBySP = A × A^T × P;
//自适应混合,输出推荐结果 r
9. for j = 1 to m do
10. calculate users' activeness;
11. for k = 1 to n do
12. calculate items' freshness;
13. calculate w(j,k);
14. for i = 1 to m do
15. for j = 1 to n do
16. if M(i,j) = 0
17. r(i,j) = w(i,j) × canMatrixByCon(i,j) + (1/w(i,j)) × canMatrixBySP(i,j);
```

具体步骤描述如下:

步骤 1　根据物品的内容构造张量,例如根据电影名称、导演、演员、类型构造一个四维张量,Tucker 分解后得到电影的低维稠密向量表示,第 4 行和第 5 行与传统基于内容的推荐类似,通过构造物品相似度矩阵,生成预测结果。

步骤 2　使用基于短路径枚举的方法生成预测结果。具体来说,利用统计学方法为每个物品计算评分,并使用图 4 计算最终的预测矩阵。

步骤 3　将前 2 个步骤的结果自适应混合。第 9 行~第 13 行计算所有用户的活跃度和所有物品的新鲜度以及对应的权重。对于每一个用户,遍历其所有未交互的物品,根据权重计算自适应评分,生成最终推荐结果。

3　实验结果与分析

3.1　实验设置

3.1.1　数据集描述

本文分别在 Movielens、doubanbook 这 2 个真实数据集上进行实验,前者为公开数据集,后者爬取自豆瓣读书。表 1 所示为 2 个数据集的相关信息描述。另外,对于基于内容的属性选取部分,Movielens 数据集中选取的属性是电影名称、导演、演员和类型,doubanbook 选取的属性是书籍名称、作者、类别、关键词。

表 1　2 种数据集描述

名称	数据集	
	Movielens	doubanbook
用户数	2 113	466
物品数	10 197	25 032
标签数	13 222	3 845

本文使用留出法进行评估,数据集中每个用户抽取 80% 的记录作为训练集,其余作为测试集。

3.1.2　评价指标

本文使用的评价指标包括:Top-N 结果的准确率 P_N 以及平均准确率均值(MAP)。

$$P_N = \frac{|L_u^N \cap T_u^+|}{N} \tag{8}$$

其中，L_u^N 代表用户 u 的 Top-N 顺序推荐列表，T_u^+ 代表用户 u 测试集中正面评价物品集，N 代表推荐数目。

$$MAP=\frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{j=1}^{m_i}p(j)\times rel(j)}{|T_i^+|} \tag{9}$$

其中，N 代表用户的数量，$p(j)$ 是截止排名从 1 ~ j 的准确率，$rel(j)$ 是一个指数函数，m_i 是用户 i 的推荐列表总数。

3.2 基于张量的内容对比

在基于内容的推荐部分，将本文提出的基于张量的内容推荐与传统的基于词向量的内容推荐(RSV)进行比较。如表 2 和表 3 所示，本文算法在 2 个数据集上准确率分别平均提升 19.12% 和 34.4%，平均准确率均值分别提升 91.5% 和 49.2%。其中，MAP 是用户数量为 10 的平均准确率均值。该结果验证了基于张量挖掘内容相似度的算法的有效性。

表 2 Movielens 上 2 种算法性能对比

算法	P_3	P_5	P_8	P_{10}	MAP_{10}
RSV 算法	0.137	0.123	0.104	0.07	0.012 9
本文算法	0.153	0.145	0.123	0.09	0.024 7

表 3 doubanbook 上 2 种算法性能对比

算法	P_3	P_5	P_8	P_{10}	MAP_{10}
RSV 算法	0.026	0.023	0.020	0.020	0.019 1
本文算法	0.036	0.032	0.027	0.025	0.028 5

3.3 短路径方法与以往方法的对比

在基于短路径的推荐算法的性能评估部分，本文提出的 2 种短路径枚举算法(SP1 和 SP2)基于用户与物品以及标签的交互，与基于协同过滤的方法相似，因此将 SP1 和 SP2 与经典 UserKNN 和随机游走(RW)算法进行对比。UserKNN[18] 是一个标准的基于用户 k 近邻的协同过滤算法。本文实验将 k 设置为 3。随机游走算法通过构造用户、物品和标签的邻接矩阵，并根据典型的随机游走的原理实现推荐。SP1 和 SP2 是分别是 2.3 节中提出的 2 种短路径枚举算法。

表 4 和图 5 分别给出 4 种方法在推荐数为 10 的准确率和运行时间开销对比。从表 4 可以看出，本文提出的 2 种短路径枚举算法在 2 个数据集上准确率都明显优于 UserKNN 和 RW。其中，SP1 算法在 2 个数据集上较 UserKNN 提升了 225.2% 和118.8%，SP2 算法在 2 个数据集上有提升但效果不明显。SP1 效果要好于 SP2，这是因为 SP1 找到的路径列表其实是包含 SP2 包含的路径列表，它的统计信息更能反映用户的偏好，但显然需要更多的时间开销。图 5 给出 4 种算法在时间性能上的比较。可以看出，SP1 和 SP2 的时间性能优势明显，在 Movielens 中 SP1 和 SP2 时间优化最高，较 UserKNN 和 RM 分别达到了 9.53 倍、13.09 倍和 21.89 倍、29.17 倍。UserKNN 由于对于每个用户都需要遍历整个数据集的用户计算相似度花费大量时间，RW 由于需要在大规模矩阵中多次迭代计算产生了大量的时间消耗，SP1 和 SP2 用 3 个矩阵相乘得到最后的预测评分矩阵。

表 4 不同算法准确率对比结果

算法	准确率	
	Movielens	doubanbook
UserKNN 算法	0.111	0.016
RW 算法	0.131	0.013
SP1 算法	0.358	0.035
SP2 算法	0.144	0.018

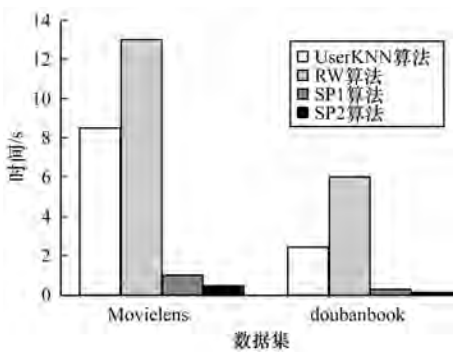

图 5 4 种协同过滤算法的运行时间对比

3.4 混合模型

由于在 3.3 节的实验结果表明了基于短路径枚举算法第一种模型性能优势明显，因此本文使用第 1 种短路径枚举模型作为混合模型中的协同过滤部分。

3.4.1 α 和 β 对准确率的影响

表 6 和表 7 给出在 Movielens 上不同稀疏度数据集中设置不同的 α 和 β 准确率的变化。在表 6 中，抽取总数据集中 20% 的记录作为训练集，此时用户和物品的平均记录数分别为 100 和 20，可以看出，当 α 和 β 设置为 100 和 20 时，准确率达到最高。在表 7 中，抽取总数据集 80% 的记录作为训练集，此时用户和物品的平均记录数约为 300 和 60，可以看出，当 α 和 β 设置为 300 和 60 时，精确度达到最高。因此，在混合计算之前，首先判断当前数据集的用户与物品的平均记录数，将 α 和 β 分别设置为对应值。同理，doubanbook 也根据上述方法设置。

表 6　在 20% 训练集上不同 α 和 β 时的本文算法准确率对比

β	准确率			
	$\alpha=100$	$\alpha=200$	$\alpha=400$	$\alpha=600$
10	0.528	0.528	0.527	0.526
20	0.528	0.524	0.522	0.518
40	0.497	0.465	0.382	0.315
60	0.418	0.302	0.190	0.132

表 7　在 80% 训练集上不同 α 和 β 时的本文算法准确率对比

β	准确率			
	$\alpha=200$	$\alpha=300$	$\alpha=400$	$\alpha=600$
20	0.134	0.134	0.134	0.134
40	0.232	0.231	0.231	0.231
60	0.233	0.232	0.232	0.232
80	0.233	0.233	0.233	0.233

3.4.2　结果分析

本文对比了 3 种混合算法,分别是 CBCF、AHCF-basic 和 AHCF。

CBCF 是一个经典的混合推荐算法[19],其目的是解决数据稀疏性问题,该算法利用基于内容的预测来构造伪评分矩阵,并在伪评分矩阵上进行基于用户的协同推荐。

AHCF-basic 即 AHCF 的简化版,其不考虑用户的活跃度和物品的新鲜度,只是简单地将基于内容评分和基于短路径枚举方法的得分等权加和。

AHCF 是本文提出的自适应混合模型,即考虑用户活跃度和物品新鲜度并且自适应调整 2 种方法比重的算法。

表 8 和表 9 反映了 2 个数据集上 3 种混合方法的精确率比较。可以看出,2 个数据集上 AHCF 算法准确率较高。在 Movielens 中,AHCF 较 CBCF 和 AHCF-basic在P_1上提升幅度最大,分别提升21.4% 和 30.8%。在 doubanbook 中,AHCF 较 CBCF 和 AHCF-basic 在 P_9 提升最多,可达 28.6% 和 63.6%。

表 8　Movielens 上 3 种算法准确率对比结果

算法	P_1	P_3	P_5	P_7	P_9
CBCF 算法	0.140	0.159	0.141	0.135	0.120
AHCF-basic 算法	0.130	0.156	0.132	0.127	0.117
AHCF 算法	0.170	0.170	0.156	0.147	0.137

表 9　doubanbook 上 3 种算法准确率对比结果

算法	P_1	P_3	P_5	P_7	P_9
CBCF 算法	0.022	0.021	0.018	0.015	0.014
AHCF-basic 算法	0.021	0.018	0.015	0.013	0.011
AHCF 算法	0.025	0.023	0.021	0.019	0.018

为验证 AHCF 处理数据稀疏性问题,本文依次随机选择原始数据集中每个用户的 10%、20%、30%、50%、80% 的记录作为训练集,剩余部分作为测试集,选择 CBCF,AHCF-basic 和 AHCF 进行比较,且用户数量为 10。图 6 展示了 2 个数据集上 3 种算法的准确率对比结果。可以发现,AHCF 在数据稀疏性不断变化的过程中始终处于领先位置。这是因为当数据集较稀疏时,用户和物品的记录都很少,使用短路径枚举的方法无法获得足够的统计信息。同时,需要增大基于内容的评分比重,用于修正短路径枚举方法带来的偏差。CBCF 由于使用了内容预测的方法来填补了评分矩阵,一定程度上也缓解了稀疏性问题。然而这种填补方法很有可能偏离用户的真正兴趣。同时固定权值的 AHCF-basic 进一步验证了 AHCF 的合理性和有效性。

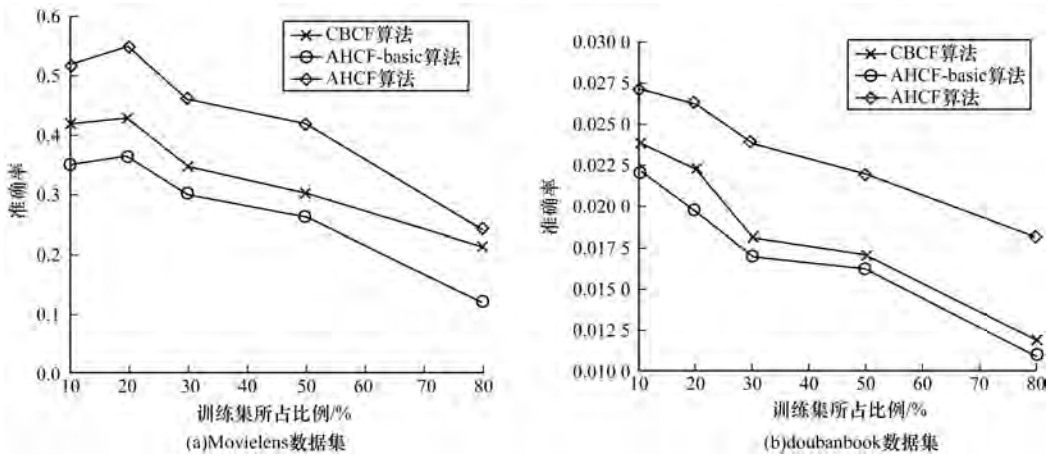

图 6　不同算法在数据稀疏度不同时准确率对比结果

4　结束语

本文提出一种动态自适应混合协同推荐算法。该算法结合基于内容的推荐方法,通过基于张量的相似度计量方法来计算物品间相似度,在此基础上,采用短路径叠加方法大幅提高了算法的效率。在 2 个真实数据集上的实验结果表明,该方法不仅推荐准确率较高,而且能够节省计算开销。

参考文献

[1] GEMMIS M D,LOPS P,SEMERARO G,et al. Integrating tags in a semantic content-based recommender[C]//Proceedings of ACM Conference on Recommender Systems. New York,USA:ACM Press,2008:163-170.

[2] MOONEY R J,ROY L. Content-based book recommending using learning for text categorization[C]//Proceedings of ACM Conference on Digital Libraries. New York,USA:ACM Press,2000:195-240.

[3] 王霞. 协同过滤在电子商务推荐系统中的应用研究[D]. 西安:西北大学,2003.

[4] KUNEGIS J,SCHMIDT S. Collaborative filtering using electrical resistance network models[C]//Proceedings of the 7th Industrial Conference on Advances in Data Mining. Berlin,Germany:Springer,2007:269-282.

[5] COOPER C,LEE S H,RADZIK T,et al. Random walks in recommender systems:exact computation and simulations[C]//Proceedings of the 23rd International Conference on World Wide Web. New York,USA:ACM Press,2014,811-816.

[6] SINGH A P,MEEK C,SURENDRAN A C. Recommendations using absorbing random walks[EB/OL]. [2018-02-20]. http://www.docin.com/p-1421996299.html.

[7] 罗辛,欧阳元新,熊璋,等. 通过相似度支持度优化基于K近邻的协同过滤算法[J]. 计算机学报,2010,33(8):1437-1445.

[8] PENG Jing,ZENG Dajun,ZHAO Huimin,et al. Collaborative filtering in social tagging systems based on joint item-tag recommendations[C]//Proceedings of the 19th ACM International Conference on Information and Knowledge Management. New York,USA:ACM Press,2010:809-818.

[9] CLAYPOOL M,GOKHALE A,MIRANDA T,et al. Combining content-based and collaborative filters in online newspaper[C]//Proceedings of ACM SIGIR Workshop on Recommender Systems. New York,USA:ACM Press,1999:40-48.

[10] PAZZANI M J. A framework for collaborative,content-based and demographic filtering[J]. Artificial Intelligence Review,1999,13(5/6):393-408.

[11] PARK H S,YOO J O,CHO S B. A context-aware music recommendation system using fuzzy Bayesian networks with utility theory[C]//Proceedings of International Conference on Fuzzy Systems and Knowledge Discovery. Berlin,Germany:Springer,2006:970-979.

[12] ZHENG Haitao,YAN Yanghui,ZHOU Yingmin. Graph-based hybrid recommendation using random walk and topic modeling[C]//Proceedings of Conference on Web Technologies and Applications. Berlin,Germany:Springer,2015:573-585.

[13] JEH G,VWIDOM J. SimRank:a measure of structural-context similarity[C]//Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York,USA:ACM Press,2002:538-543.

[14] ZHANG Zike,ZHOU Tao,ZHANG Yicheng. Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs[J]. Statistical Mechanics and Its Applications,2010,389(1):179-186.

[15] TSO-SUTTER K H L,MARINHO L B,SCHMIDT-THIEME L. Tag-aware recommender systems by fusion of collaborative filtering algorithms[C]//Proceedings of ACM Symposium on Applied Computing. New York,USA:ACM Press,2008:1995-1999.

[16] LOPES R,ASSUNCÃO R,SANTOS R L T. Efficient Bayesian methods for graph-based recommendation[C]//Proceedings of the 10th ACM Conference on Recommender Systems. New York,USA:ACM Press,2016:333-340.

[17] KOLDA T G,SUN Jimeng. Scalable tensor decompositions for multi-aspect data mining[C]//Proceedings of the 8th IEEE International Conference on Data Mining. Washington D. C.,USA:IEEE Computer Society,2008:363-372.

[18] BELL R M,KOREN Y. Improved neighborhood-based collaborative filtering[EB/OL]. [2018-03-01]. https://www.ixueshu.com/document/eaab2fa803bce4da318947a18e7f9386.html.

[19] MELVILLE P,MOONEY R J,NAGARAJAN R. Content-boosted collaborative filtering for improved recommendations[C]//Proceedings of the 8th National Conference on Artificial Intelligence. [S.l.]:AAAI Press,2002:187-192.

编辑 赵 辉