软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(12):2852−2864 [doi: 10.13328/j.cnki.jos.004725]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于用户信任和张量分解的社会网络推荐*

邹本友[1,2], 李翠平[1,2], 谭力文[1,2], 陈 红[1,2], 王绍卿[1,2,3]

[1](数据工程与知识工程教育部重点实验室,北京 100872)

[2](中国人民大学 信息学院,北京 100872)

[3](山东理工大学 计算机学院,山东 淄博 255091)

通讯作者: 李翠平, E-mail: licuiping@ruc.edu.cn

摘 要: 社会化网络中的推荐系统可以在浩瀚的数据海洋中给用户推荐相关的信息.社会网络中用户之间的信任关系已经被用于推荐算法中,但是目前的基于信任的推荐算法都是单一的信任模型.提出了一种基于主题的张量分解的用户信任推荐算法,用来挖掘用户在不同的物品选取的时候对不同朋友的信任程度.由于社交网络更新速度快,鉴于目前的基于信任算法大都是静态算法,提出了一种增量更新的张量分解算法用于用户信任的推荐算法.实验结果表明:所提出的基于主题的用户信任推荐算法比现有算法具有更好的准确性,并且增量更新的推荐算法可以大幅度提高推荐算法在训练数据增加后的模型训练效率,适合更新速度快的社会化网络中的推荐任务.

关键词: 推荐系统;社会网络;信任;张量分解;增量更新

中图法分类号: TP311

中文引用格式: 邹本友,李翠平,谭力文,陈红,王绍卿.基于用户信任和张量分解的社会网络推荐.软件学报,2014,25(12):2852−2864. http://www.jos.org.cn/1000-9825/4725.htm

英文引用格式: Zou BY, Li CP, Tan LW, Chen H, Wang SQ. Social recommendations based on user trust and tensor factorization. Ruan Jian Xue Bao/Journal of Software, 2014,25(12):2852−2864 (in Chinese). http://www.jos.org.cn/1000-9825/4725.htm

Social Recommendations Based on User Trust and Tensor Factorization

ZOU Ben-You[1,2], LI Cui-Ping[1,2], TAN Li-Wen[1,2], CHEN Hong[1,2], WANG Shao-Qing[1,2,3]

[1](Key Laboratory of Data Engineering and Knowledge Engineering, Ministry of Education, Beijing 100872, China)

[2](Information School, Renmin University of China, Beijing 100872, China)

[3](School of Computer Science and Technology, Shandong University of Technology, Zibo 255091, China)

Corresponding author: LI Cui-Ping, E-mail: licuiping@ruc.edu.cn

Abstract: In social networks, recommender systems can help users to deal with information overload and provide personalized recommendations to them. The trust relationship of users is used in the social networks' recommender systems. But the state-of-art algorithms only use the single trust relationship which cannot capture the trust to user's friends when looking for different items. This paper proposes a topic-based trust recommendation algorithm using tensor factorization model. As the social information changes rapidly, the state-of-art algorithms often need redo factorization. To address the issue, the paper also presents an effective incremental method to adaptively update its previous factorized components rather than re-computing them on the whole dataset when the data changes. Experiments show that the proposed method can achieve better performance and the incremental method is suitable for the rapid changes in the social networks.

Key words: recommendation systems; social network; trust; tensor factorization; incremental update

* 基金项目: 国家重点基础研究发展计划(973)(2014CB340402, 2012CB316205); 国家高技术研究发展计划(863)(2014AA015204); 国家自然科学基金(61272137, 61033010, 61202114); 国家社会科学基金(12&ZD220)

收稿时间: 2014-05-02; 修改时间: 2014-08-21; 定稿时间: 2014-09-30

推荐系统(recommender systems)[1,2]作为个性化服务研究领域的重要分支,通过挖掘用户与项目之间(user-item)的二元关系,帮助用户从大量数据中发现其可能感兴趣的项目(如 Web 信息、服务、在线商品等),并生成个性化推荐以满足个性化需求.目前,推荐系统在电子商务(如 Amazon、eBay、Netflix、阿里巴巴、豆瓣网、当当网等)、信息检索(如 iGoogle、GroupLens、百度等)以及移动应用、电子旅游、互联网广告等等众多应用领域取得较大进展.

随着社会网络的迅猛发展,社会网络的大数据时代已经来临.在线社会网络中,如果两个用户之间的交互很频繁,表明他们之间的关系强度很大.用户间的信任程度与交互经历相关,当两个用户之间的经历为正关系时,用户间的信任程度会提交;反之,信任程度下降[3].在 eBay 和 Amazon 这样的在线购物网站中,用户间的信任是根据他们之间历史交易的反馈来获得[4],采用用户之间的信任度可以提高物品推荐的满意度[5].

Granovetter[6]引入用户间的链接强度来表示社会网络中的信任关系,强连接关系表示用户间的比较高的信任关系,弱连接表示用户间的信任关系低.Gilbert 和 Karahalios[7]将连接强度的概念运用到社交网络的场景中,对于 Facebook 的数据集,作者将用户之间的信任关系映射为二元模式,即用户之间要么信任要么不信任.但是在现实中,用户之间的信任程度是不一样的,二元模式并不能表现出这个特点.Xiang 等人[8]提出了一种无监督学习的方法来确定社会网络中信任关系的强度大小.Zarghami 等人[9]引入 T-index 的概念来估计用户之间的信任程度,并且根据用户之间的信任程度来给用户推荐新朋友.Jamali 等人[10]提出了一种将用户信任关系与矩阵分解算法相结合的推荐算法,并且应用到社会网络的推荐中.

Ma 等人[11]提出将朋友的信任关系加入到社会化推荐.在用户对物品的评分做预测时,不仅考虑了用户自身的偏好对于评分的影响,而且考虑了用户朋友对用户做评分时候的影响.文中将两种影响用线性叠加的方法做处理,得到用户对物品的评分.用户 u 对物品 i 的评分预测为 $\widehat{R_{u,i}}=\alpha U_u^T V_i+(1-\alpha)\sum_{v\in N_u} T_{u,v}U_v^T V_i$, 其中,$\alpha$表示用户朋友对于用户评分的影响权重.

上述的推荐算法存在以下两个缺陷:

1) 大多数的推荐算法基于一个假设:用户对其邻居的影响是单一的,但是在实际的社会网络中,每个用户的兴趣是不同的,用户对于不同的话题或知识的见解是不同的.比如:用户买手机电脑等数码产品的时候,这个用户在很大程度上会听取对数码产品比较了解的朋友的建议;用户如果要去三亚旅游,会倾向去去过三亚或者旅游爱好者的朋友的建议;用户如果想购买数据挖掘的教程,可能会倾向微博中比较活跃的从事数据挖掘研究的学者的建议;
2) 现有的推荐算法中,大部分都是假设训练数据是静态的,即训练集在模型训练的时候已经固定,如果训练集中有新的数据加入,比如新用户的加入、新物品的加入等,算法要对新的训练集重新计算.对于更新速度快的社会网络的推荐任务,这种算法就会显得不合时宜.

为了解决上述两个传统推荐算法的缺点,本文提出了两种不同的算法:

1) 提出了一种基于主题和张量分解的信任推荐算法,挖掘用户在不同的主题上对朋友的信任关系;
2) 为了解决传统推荐算法中的训练数据更新引起的算法重算问题,提出了一种可以增量更新的张量分解的用户信任推荐算法.

本文第 1 节介绍推荐算法和张量分解的相关基础知识.第 2 节提出基于主题的张量分解的用户信任推荐算法.第 3 节介绍动态更新的张量分解算法用于朋友信任推荐.第 4 节是本文的实验部分,通过与传统基于信任的推荐方法的对比,分析我们提出的算法在有效性和效率的表现.第 5 节是本文的总结.

1 背景知识

本节主要介绍本文用到的相关技术,首先介绍传统的推荐算法和上下文感知的社会化推荐算法,其次介绍张量相关知识以及张量分解相关技术.

1.1 传统推荐算法

传统的推荐算法是通过用户的行为数据建立用户-项目二者之间的二元关系,通过对其挖掘分析得到每个用户潜在的感兴趣的项目,从而进行个性化推荐.Adomavicius 等人[1]将推荐系统定义为:假设 $User=\{u_1,u_2,\ldots,u_N\}$为推荐系统中所有用户的集合,$Item=\{i_1,i_2,\ldots,i_N\}$为推荐系统中所有的项目集合(例如图书、电影等).记 s 为效用函数,衡量项目 i 与用户 u 之间的相关性,例如 $s:User\times Item\rightarrow Rating$,其中,$Rating$ 为整体有序的实数集合(例如在电影的评分推荐中,大都采用 1~5 之间的整数作为用户对电影的喜欢程度),推荐系统的目标就是对给定的用户和项目集合找到使得效用函数 $s(\cdot)$最大的项目,即: $\forall u\in User, s'_u=\arg\max_{i\in I} s(u,i)$.

1.2 上下文感知的社会化推荐算法

虽然传统的推荐算法采用用户-项目模型已经取得了很大的成功,但这些算法都忽略了一点,就是用户所处的上下文(context)信息.在社会化网络中,用户的上下文信息包括时间、地点、用户的社会关系等信息.Adomavicius 等人[1]首次提出上下文感知推荐系统,将传统的用户-项目二维模型 $s:User\times Item\rightarrow Rating$ 扩展为包含上下文信息的多维模型:$s:D_1\times\ldots\times D_n\rightarrow Rating$,或者 $s:User\times Item\times Context\rightarrow Rating$.Karatzoglou 等人[12]提出采用张量分解的推荐算法,将用户、电影、上下文信息构建为用户-项目-上下文的三阶张量,通过张量分解算法,得到用户在不同上下文信息方面对项目的喜好程度.

1.3 张量基本知识

张量(tensor)是高维数组的总称[13],举例来说,一阶张量就是一个向量,二阶张量是矩阵,三阶张量或者更高阶的张量称为高阶张量.图 1 所示为三阶张量 $X\in\mathbb{R}^{I\times J\times K}$.

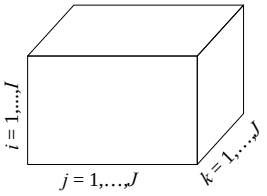


Fig.1 3-Order tensor

图 1 三阶张量图示

定义 1(模式(mode)).定义张量的模式为张量的维度数量大小.例如,张量 $X\in\mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ 的模式为 N,又称为阶.

Table 1 Symbols used in this paper

表 1 本文用到的符号表

符号	定义
X	初始张量
$X_{(i_1,\ldots,i_N)}$	张量 X 的元素
$X_{(\ldots,i_{n-1},:,i_{n+1},\ldots)}$	张量 X 的 n 模式向量
N	张量 X 的模式
$uf(X,n)$或 $X_{(n)}$	张量 X 的 n 模式展开矩阵
$U^{(n)}$	张量分解后的第 n 模式基础矩阵
$\times_n$	张量与矩阵的 n 模式乘积

定义 2(展开矩阵(matrix unfolding)). 将张量 $X\in\mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ 按照第 n 模式展开,并且以矩阵的形式展现,这个矩阵被称为展开矩阵,符号表示为 $uf(X,n)$.张量 X 的第 n 模式展开矩阵表示为: $X_{(n)}\in\mathbb{R}^{I_n\times(I_1I_2\ldots I_{n-1}I_{n+1}\ldots I_N)}$. $X_{(n)}$的列向量称作张量 X 的 n 模式向量.

例如,图 2 以三阶张量 A 为例,将其沿第 1~3 模式展开后得到 3 个展开矩阵 $A_{(1)},A_{(2)},A_{(3)}$如下:

$$A_{(1)}=\begin{bmatrix}1&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&0&0&1\end{bmatrix},A_{(2)}=\begin{bmatrix}1&0&0&0&1&0&0&0&0\\1&0&0&0&0&0&0&0&1\end{bmatrix},A_{(3)}=\begin{bmatrix}1&0&0&1&0&0\\0&1&0&0&0&0\\0&0&0&0&0&1\end{bmatrix}.$$

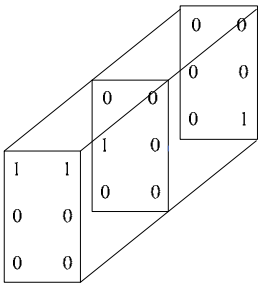


Fig.2 Example of tensor A

图 2 张量 A 的图例

定义 3(模式乘积(mode product)). n 模式乘积为张量 $X\in\mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ 与矩阵 $U\in\mathbb{R}^{J_n\times I_n}$ 在第 n 模式上的乘积,可以表示为 $A\times_n U$,结果为大小为 $\mathbb{R}^{I_1\times\ldots\times J_n\times\ldots\times I_N}$.本文简称为模乘.例如,大小为 3×4×5 的张量 A 与大小为 2×3 的矩阵 1 模乘得到的张量 A'的大小为 2×4×5.关于张量的更多细节内容,见文献[14].

定理 1. 张量 A 与矩阵 U 的 n 模乘的 n 模矩阵展开可以等价转换为第 n 模矩阵展开得到的 $A_{(n)}$与矩阵 U 的乘积:$uf(A\times_n U)=A_{(n)}\times U$.

1.4 张量分解模型

张量分解(又称为高阶奇异值分解、HOSVD)是在矩阵奇异值分解(SVD)的概念上的延伸.文献提出了多种张量分解的模型:Tucker 模型[15]、PARAFAC 模型[16]和 CANDECOMP 模型[17].Tucker 模型是一种高阶主成分分析方法,它将 N 阶张量分解为一个核心张量(core tensor)和 N 个因子矩阵乘积形式:

$$X\approx C\times_1 U^{(1)}\times_2 U^{(2)}\ldots\times_N U^{(N)} \tag{1}$$

其中, $U^{(1)}\in\mathbb{R}^{I_1\times R_1}$,$U^{(2)}\in\mathbb{R}^{I_2\times R_2}$,...,$U^{(N)}\in\mathbb{R}^{I_N\times R_N}$,核心张量 $C\in\mathbb{R}^{R_1\times R_2\times\ldots\times R_N}$.因子矩阵可以被认为是每一维上的主成分,$U^{(i)}(1\le i\le N)$可以通过对张量 X 的 n 模展开矩阵 $X_{(n)}$做 SVD 分解的到的左奇异矩阵得到[14].

图 3 中给出了三阶张量的 Tucker 分解模型的图例,一个三阶张量可以分解为 3 个因子矩阵和一个核心张量.如果 R_1,R_2,R_3 分别小于 I_1,I_2,I_3,我们可以将核心张量 C 看作是对原始张量 X 的压缩,在某些情况下,压缩后的张量的存储空间可以明显小于原始张量所占的存储空间,比如对稀疏张量的压缩处理[18].实际上,PARAFAC 模型和 CANDECOMP 模型看以看做是 Tucker 分解的特例,他们要求分解后的 $R_1=R_2=R_3$.

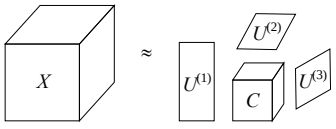


Fig.3 Tucker decomposition of 3-mode tensor

图 3 三阶张量的 Tucker 分解图例

1.5 张量N阶近似

张量 N 阶近似(rank-($R_1,R_2,\ldots,R_N$) approximation):假设 N 阶张量 X 的大小为 $I_1\times I_2\times\ldots\times I_N$,$X$ 的 N 阶近似是找到一个与 X 近似的张量 $\tilde{X}\in\mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$,满足二者的平方差最小: $\tilde{X}=\arg\min_{\tilde{X}}\|X-\tilde{X}\|^2$.

当 $R_n<<I_n(1\le n\le N)$时,张量 $\tilde{X}$ 可以看作是原始张量 X 的稠密化近似,对于稀疏的大数据集来说,可以得到一个很好的压缩近似[13].

当因子矩阵 $U^{(1)} \in \mathbb{R}^{I_1 \times R_1}, U^{(2)} \in \mathbb{R}^{I_2 \times R_2}, ..., U^{(N)} \in \mathbb{R}^{I_N \times R_N}$ 确定时,核心张量 C 可以通过张量 X 与因子矩阵计算得到:

$$C = X \times_1 U^{(1)^T} \times_2 U^{(2)^T} ... \times_N U^{(N)^T} \tag{2}$$

通过公式(2)可以看到:对于求 X 的近似,可以只计算因子矩阵.本文中,我们通过这个方法求张量的核心张量,然后代入公式(1)得到张量的 N 阶近似.

2 基于用户主题信任推荐算法

本节首先介绍基于用户信任与矩阵分解相结合的推荐算法,然后介绍基于用户信任与张量分解模型的推荐算法.

在上一节中提到的基于用户信任的推荐算法中,全部都是只考虑了用户单方的信任关系.以无向图为例,图 4 中给出了算法中所用到的用户信任关系的图例.对于用户 1 来说,他信任的朋友只有用户 2,用户 2 信任的朋友是用户 1 和用户 4,用户 3 的信任朋友为用户 4,用户 4 的信任朋友为用户 2 和用户 3.这种方法可以很直观地找到用户的信任关系,方法简单易用.

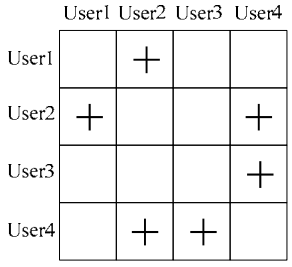


Fig.4 Trust between users

图 4 用户间的信息关系

但是在实际的应用中,比如社会网络中,用户存在多方面的兴趣.因此,用户的朋友中也是与用户兴趣相关,在不同方面的鉴赏能力是不同的.比如,每个用户都有自身擅长的领域,有些用户可能对电子产品熟悉,有些用户对汽车产品熟悉,这些特性决定了每个用户在不同方面会考虑不同朋友的建议.基于这种思想,我们提出了一种基于用户多方影响力的推荐算法.图 5 中给出了基于主题的信任关系框图,在引入了主题后,我们可以看到:在主题 1 层面上,用户间的信任关系有 User1-User2,User3-User4;主题 2 层面上,用户的信任关系为 User1-User2,User2-User4;在主题 3 层面上,用户的信任关系为 User2-User4,User3-User4.

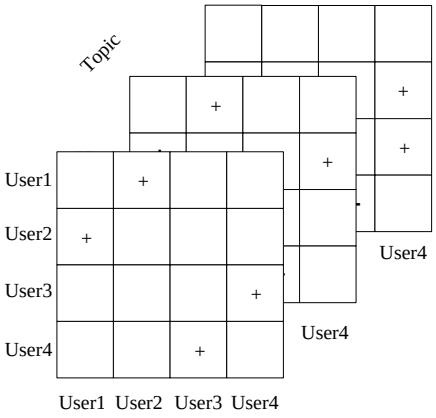


Fig.5 Trust between users based on topic

图 5 基于主题的用户间的信息关系

在对基于主题的用户信任推荐中,我们使用三维张量来模拟用户-用户-主题的三者关联关系.我们根据文献[14]中采用的交替最小二乘法(alternating least squares,简称 ALS)实现张量 N 阶近似,提出的算法为 TrustTensor 算法.交替最小二乘法为迭代算法,每次迭代过程是根据其他 N–1 个基础矩阵求解张量的一个基础矩阵.例如,在第 l+1 次迭代求解基础矩阵 $U_{l+1}^{(n)}$ 过程中,我们根据其他 N–1 个基础矩阵 $U_{l+1}^{(1)},...,U_{l+1}^{(n-1)},U_l^{(n+1)},...,U_l^{(N)}$:

首先通过以下公式计算得到近似张量 A':

$$A' = A \times_1 U_{l+1}^{(1)^T} \times_2 U_{l+1}^{(2)^T} ... \times_{(n-1)} U_{l+1}^{(n-1)^T} \times_{(n+1)} U_l^{(n+1)^T} ... \times_N U_l^{(N)^T};$$

其次,通过对张量 A'的展开矩阵 $uf(A',n)$进行 SVD 计算得到基础矩阵 $U_{l+1}^{(n)}$.算法的伪代码描述如下:

算法 1. 基于主题的 TrustTensor 算法.

1. 输入:用户-用户-项目张量 A,主题数目 R,迭代阈值ε;
2. 输出:核心张量 C,基础特征矩阵 $U^{(1)},U^{(2)},U^{(3)}$.
3. 初始化 $U^{(1)},U^{(2)},U^{(3)}$;
4. 初始化 $C_0 = A \times_1 U^{(1)^T} \times_2 U^{(2)^T} \times_3 U^{(3)^T}$;
5. Let l=0;
6. for each $n \in [1,2,3]$:
7. $A'=A$;
8. for each $m \in [1,n-1]$ and $m \neq n$ Do
9. $A' = A' \times_m U_{l+1}^{(m)^T}$;
10. End for
11. for each $m \in [n,3]$ Do
12. $A' = A' \times_m U_l^{(m)^T}$;
13. End for
14. $(U_{l+1}^{(n)}, \Sigma_{l+1}^{(n)}, V_{l+1}^{(n)}) = SVD(uf(A',n), R)$;
15. End for
16. $C_{l+1} = A \times_1 U_{l+1}^{(1)^T} \times_2 U_{l+1}^{(2)^T} \times_3 U_{l+1}^{(3)^T}$;
17. if $\|C_{l+1}\|^2 - \|C_l\|^2 \leq \varepsilon$ QUIT;
18. Otherwise,$l=l+1$,并且跳转到第 6 行
19. End if

复杂度分析:算法 1 的主要的资源开销是算法每次迭代计算张量 A 的近似张量 A'的模乘运算和对 A'的展开矩阵进行 SVD 分解过程.计算近似张量 A'的复杂度为 $O\left(\sum_{i=1,i\neq n}^{3}\left(I_n R_n \prod_{j=1}^{n-1} R_j \prod_{j=n+1}^{3} I_j\right)\right)$;对 A'的展开矩阵 $A'_{(n)} \in \mathbb{R}^{I_n \times (R_1 ... R_n R_{n+1} ... R_3)}$ 进行 SVD 计算的复杂度为 $O\left(I_n \prod_{j=1,j\neq n}^{3} R_j R_n^2\right)$;求核心张量 C_{l+1} 的复杂度与求 A'的复杂度相同.所以,算法总体复杂度为 $O\left(\sum_{n=1}^{3}\left(\sum_{i=1,i\neq n}^{3}\left(I_n R_n \prod_{j=1}^{n-1} R_j \prod_{j=n+1}^{3} I_j\right) + I_n \prod_{j=1,j\neq n}^{3} R_j R_n^2\right)\right)$.在实际应用中,张量 A 每个维度的大小 I_n 远大于分解因子 R_n,所以算法的复杂度可以粗略记为 $O\left(\prod_{i=1}^{3} I_i\right)$,与张量 A 的大小密切相关.

3 增量更新的主题信任推荐算法

前一节描述了基于主题的用户信任的张量分解推荐算法 TrustTensor,此算法可以描述在不同的主题上下文上用户信任的关系.

对于大多数的社会网络,比如 Facebook、Twitter 以及新浪微博等,每天都有成千上万的新用户注册,网站每天会新增很多的消息、话题等,对于这些新用户和新物品的推荐的冷启动问题已经成为推荐系统重要的挑战.

但是大多数的推荐算法是基于静态数据的,没有考虑到用户和物品的增长,当新用户和物品加入后,推荐系统需要对新的张量进行重新分解,这样,原来分解过的张量数据也要重新计算,浪费了大量的资源.

为了解决这个问题,我们提出了一种动态增量更新的张量分解算法.该方法不需要对原始张量重新计算,只是用到原始张量分解的结果和新加入的用户、物品构成的新张量,对原始分解结果进行动态更新.在介绍动态增量张量分解算法之前,我们先介绍下矩阵增量 SVD 分解.

3.1 增量SVD分解模型

假设有矩阵 $A\in\mathbb{R}^{I_1\times I_2}$,SVD 分解的结果为 $SVD(A)=U_t\Sigma_t V_t^T$.矩阵 $F\in\mathbb{R}^{I_1\times I_2'}$,我们介绍如何对矩阵 A 和矩阵 F 合并后的矩阵 A^*进行增量更新.我们记 $A^*=[A|F], A^*\in\mathbb{R}^{I_1\times(I_2+I_2')}$,根据 SVD 分解的性质,我们知道 U_t和 V_t 为正交矩阵,我们可以根据下面的公式求得Σ_t:

$$\Sigma_t=U_t^TU_t\Sigma_tV_t^TV_t \tag{3}$$

按照文献[19]提到的方法,我们对 A^*做如下的计算:

$$U_t^TA^*\begin{bmatrix}V_t & 0\\ 0 & I_f\end{bmatrix}=[\Sigma_t|U_t^TF] \tag{4}$$

其中,I_f为 $I_2'\times I_2'$ 大小的单位矩阵.我们记 $Y=[\Sigma_t|U_t^TF]$,然后对 Y 进行 SVD 计算.

因为矩阵 Y 的大小为 $R_1\times(R_2+I_2')$,远小于 $I_1\times(I_2+I_2')$,所以对 Y 的 SVD 计算时间会小很多.

我们假设 $Y=U_Y\Sigma_YV_Y^T$,那么:

$$A^*=U_tU_t^TA^*\begin{bmatrix}V_t & 0\\ 0 & I_f\end{bmatrix}\begin{bmatrix}V_t & 0\\ 0 & I_f\end{bmatrix}^T=U_t[\Sigma_t|U_t^TF]\begin{bmatrix}V_t & 0\\ 0 & I_f\end{bmatrix}^T=U_tY\begin{bmatrix}V_t & 0\\ 0 & I_f\end{bmatrix}^T=U_tU_Y\Sigma_YV_Y^T\begin{bmatrix}V_t & 0\\ 0 & I_f\end{bmatrix}^T \tag{5}$$

我们记 $SVD(A^*)=U_{t+1}\Sigma_{t+1}V_{t+1}^T$,根据公式(5),我们可以得到:

$$\begin{aligned}U_{t+1}&=U_tU_Y\\ \Sigma_{t+1}&=\Sigma_Y\\ V_{t+1}^T&=V_Y^T\begin{bmatrix}V_t & 0\\ 0 & I_f\end{bmatrix}^T\end{aligned} \tag{6}$$

复杂度分析:增量 SVD 算法的只要资源消耗是对矩阵 Y 的 SVD 运算,计算 U_{t+1}与 V_{t+1}^T.对 Y 进行 SVD 分解的时间复杂度的大小为 $O(R_1^2(R_2+I_2'))$;计算 U_{t+1} 的时间复杂度为 $O(I_1R_1^2)$;对于 V_{t+1}^T,我们可以对其进行分块计算,时间复杂度为 $O(I_2R_1^2)$,所以总的时间复杂度为 $O(R_1^2(R_2+I_1+I_2+I_2'))$.

3.2 增量张量分解模型

上一节描述了矩阵的增量式 SVD 分解算法(IncSVD),本节将增量分解的思想拓展到高阶张量.

我们以四阶张量为例.假设原始张量为 $A\in\mathbb{R}^{I_1\times I_2\times I_3\times I_4}$, 增量的张量为 $F\in\mathbb{R}^{I_1\times I_2\times I_3\times I_4'}$, 二者在第 4 模式合并后组成的新张量我们记为 $A^*\in\mathbb{R}^{I_1\times I_2\times I_3\times I_4^*}$, 其中, $I_4^*=I_4+I_4'$. 我们假设 $I_1=I_2=I_3=I_4=2, I_4'=1$,那么 $I_4^*=3$.图 6 中为张量 A^*的展开矩阵,为了便于理解,将图的左半部分四阶张量 A^*表示为 I_1 个三阶张量 $A'\in\mathbb{R}^{I_2\times I_3\times I_4^*}$, 假设图中灰色的部分表示新加入的部分.图的右半部分是对张量 A^*按照 1,2,3,4 模式矩阵展开得到的矩阵,其中, $A_{(1)}^*$ 为 $I_1\times(I_2\times I_3\times I_4^*)$ 的矩阵、 $A_{(2)}^*$ 为 $I_2\times(I_3\times I_4^*\times I_1)$ 的矩阵、 $A_{(3)}^*$ 为 $I_3\times(I_4^*\times I_1\times I_2)$ 的矩阵、 $A_{(4)}^*$ 为 $I_4^*\times(I_1\times I_2\times I_3)$ 的矩阵.从图中我们可以看到:新加入的张量经过矩阵展开,结果矩阵 $A_{(1)},A_{(2)},A_{(3)}$的列数会增加,矩阵 $A_{(4)}$的行数会增加.

从图 6 观察到:对于新张量 A^*的第 1 模式的矩阵展开得到 $A_{(1)}^*=[A_{(1)}|F_{(1)}]$;而第 2 模式和第 3 模式的展开矩阵要对$[A_{(2)}|F_{(2)}]$和$[A_{(3)}|F_{(3)}]$进行相应的列变换得到,我们记为 $A_{(n)}^*=[A_{(n)}|F_{(n)}]P_n, n\in\{2,3\}$;对于第 4 模式的展开

矩阵为 $A_{(4)}^{*}=\left[\begin{matrix}A_{(4)}\\F_{(4)}\end{matrix}\right]=[A_{(4)}\,|\,F_{(4)}]^{T}$.

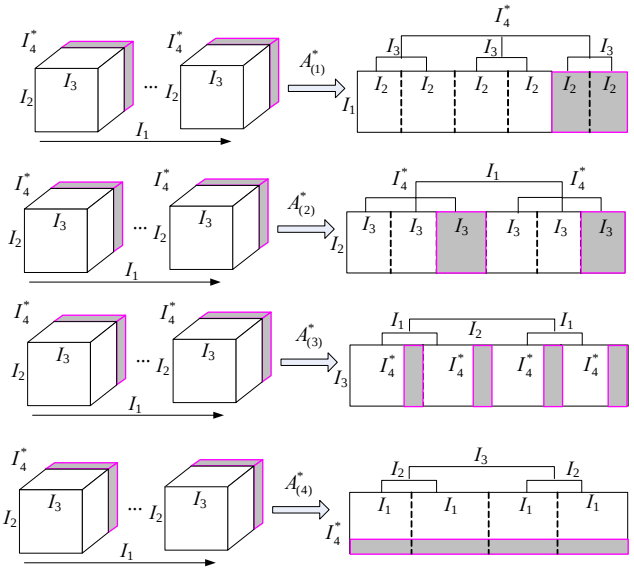


Fig.6 Illustration of unfolding a 4-order tensor

图 6 四阶张量的展开矩阵

我们引入单位矩阵 $G=[E_n|Q_n](1\le n\le I_1I_1\ldots I_{n-1})$,$E_n$ 和 Q_n 将 G 分为 $2I_1I_1\ldots I_{n-1}$ 个列向量,其中,

$$E_n\in\mathbb{R}^{(I_1\ldots I_{n-1}I_{n+1}\ldots I_N^{*})\times(I_{n+1}\ldots I_N)},Q_n\in\mathbb{R}^{(I_1\ldots I_{n-1}I_{n+1}\ldots I_N^{*})\times(I_{n+1}\ldots I_N')}.$$

那么 P_n 可以表示为

$$P_n=[E_1\,|\,E_2\,|\ldots|\,E_{I_1I_2\ldots I_{n-1}}\,|\,Q_1\,|\,Q_2\,|\ldots|\,Q_{I_1I_2\ldots I_{n-1}}]^{T}\tag{7}$$

对于图 6 中的四阶张量 A^{*}:

$$P_2=[E_1\,|\,E_2\,|\ldots|\,E_{I_1}\,|\,Q_1\,|\,Q_2\,|\ldots|\,Q_{I_1}]^{T},E_n\in\mathbb{R}^{(I_1I_3I_4^{*})\times(I_3I_4)},Q_n\in\mathbb{R}^{(I_1I_3I_4^{*})\times(I_3I_4')},$$

$$P_3=[E_1\,|\,E_2\,|\ldots|\,E_{I_1\times I_2}\,|\,Q_1\,|\,Q_2\,|\ldots|\,Q_{I_1\times I_2}]^{T},E_n\in\mathbb{R}^{(I_1I_2I_4^{*})\times(I_4)},Q_n\in\mathbb{R}^{(I_1I_2I_4^{*})\times(I_4')}.$$

上述对四阶张量的更新过程已经详细介绍,根据上述介绍,我们提出一种普适的增量张量分解算法,记为 IncTrustTensor 算法,下面给出算法的伪代码:

算法 2. IncTrustTensor 算法.

1. 输入:张量 A 的特征矩阵 $U_t^{(n)},\Sigma_t^{(n)}$ and $V_t^{(n)}(1\le n\le N)$,新张量 F;
2. 输出:特征矩阵 $U_{t+1}^{(n)},\Sigma_{t+1}^{(n)}$ and $V_{t+1}^{(n)}$.
3. Let $A_{(1)}^{*}=[A_{(1)}\,|\,F_{(1)}]$;
4. $(U_{t+1}^{(1)},\Sigma_{t+1}^{(1)},V_{t+1}^{(1)})=IncSVD(U_t^{(1)},\Sigma_t^{(1)},V_t^{(1)},F_{(1)})$;
5. For each $n\in[2,N-1]$ Do:
6. $A_{(n)}^{*}=[A_{(n)}\,|\,F_{(n)}]P_n$;
7. $(U_{t+1}^{(n)},\Sigma_{t+1}^{(n)},V_{t+1}^{(n)})=IncSVD(U_t^{(n)},\Sigma_t^{(n)},V_t^{(n)},F_{(n)})$;
8. $V_{t+1}^{(n)}=P_n^{T}V_{t+1}^{(n)}$;
9. End for

10. Let $A^{*}_{(N)}=[A_{(N)}\mid F_{(N)}]$; ;

11. $(U,\Sigma,V)=IncSVD(U_t^{(N)},\Sigma_t^{(N)},V_t^{(N)},F_{(N)})$;

12. $U_{t+1}^{(N)}=V,\Sigma_{t+1}^{(N)}=\Sigma,V_{t+1}^{(N)}=U$;

13. End

复杂度分析:在 IncTrustTensor 算法中,主要的资源消耗是计算 SVD 的时间,$Y=[\Sigma_t\mid U_t^TF]$的大小为

$$R_n\times(R_n+I_1I_2...I_{n-1}I_{n+1}...I'_N).$$

所以,对 Y进行 SVD 计算的时间为 $O(R_n^2\times(R_n+I_1I_2...I_{n-1}I_{n+1}...I'_N))$.其中,$I'_N$ 为新增加张量的最后一个维度的大小(我们将张量要增加的维度记为张量的最后一个维度),R_n 为张量分解后第 n 维度上特征值数量.所以,算法总体的时间复杂度为 $O(N\times R_n^2\times(R_n+I_1I_2...I_{n-1}I_{n+1}...I'_N))$.可见,算法的复杂度与新增张量的大小息息相关.在实际中,新增的用户规模往往要远小于已经存在的用户规模,所以增量计算的复杂度要小于重新计算的复杂度.

4 实验及结果分析

本节首先介绍实验所用到的数据集,然后说明对比算法以及评价标准,最后给出本文提出的 TrustTensor 算法与 IncTrustTensor 算法与其他方法的对比实验结果,并对实验结果进行了相应的分析.

4.1 测试数据集

为了验证提出的 TrustTensor 以及 IncTrustTensor 在社会化网络推荐方面的性能,我们进行了一系列的对比实验.我们的实验围绕以下几个问题展开:

1) 提出的 TrustTensor 和现有的基于信任的推荐算法的性能比较;
2) 提出的 IncTrustTensor 在模型训练时间与 TrustTensor 以及现有算法的比较;
3) IncTrustTensor 在推荐效果上与 TrustTensor 以及现有算法的比较;
4) IncTrustTensor 与 TrusTensor 算法在不同稀疏程度数据上的准确性比较.

为了解决上述问题,我们选取 Epinions 数据集对算法进行测试.Epinions.com 是一家在线社交网站,用户可以对图书和视频进行评分和评价;同时,Epinions 还提供了用户对其他用户的信任关系,Epinions 数据集包含了用户明确的信任关系.所以,基于信任的推荐算法大都选用这个数据集进行对比实验.我们在本文中使用的 Epinions 数据集是 Paolo 等人在文献[20]中使用的数据集,数据集中包含用户之间的信任信息以及用户对电影的评分.Paolo 已经将数据集公开,地址为 http://www.trustlet.org/wiki/Downloaded_Epinions_dataset.在本文实验中,按照 4:1 的比例将数据集分成训练集和测试集.为了验证算法在不同稀疏程度数据上的表现,我们对 Epinions 数据集按照每个用户的信任记录个数 T 以及每个商品的评分个数 I 进行了筛选,得到不同稀疏程度的数据,我们记为 Epinions-1(T>10,I>10)和 Epinions-2(T>20,I>20).表 2 中是对 Epinions 数据集各项信息的统计.

Table 2 Statistics of the Epinions dataset

表 2 Epinions 数据集的各项统计信息

Statistics	Epinions	Epinions-1	Epinions-2
Users	49 290	8 658	5 046
Social relations	487 181	337 760	274 553
Ratings	664 824	215 393	133 057
Item	139 738	10 590	5 195
Per rating every user	13.48	24.87	26.36
Per friends every user	9.88	39.01	54.41

4.2 实验中用到的算法

为了对比提出的方法与其他现有的算法在有效性和效率方面的性能,实验中所用到的算法如下:

1) STE:STE[11]是对用户计算其对朋友的信任程度,在进行物品推荐的时候,根据用户信任得到的评估和用户自己对物品的评估之间的关系,得到用户对物品的评价.因为 STE 算法并没有基于主题,为了保

证对比的一致性,在后续的实验对比中,我们记 STE 做矩阵分解的时候特征向量的维度为主题数目;

2) TrustTensor:提出的基于主题的用户信任推荐算法,通过算法得到用户在不同的主题下对朋友们的信赖程度,在推荐阶段,根据物品所属的主题以及用户的信任度对用户进行物品推荐;

3) IncTrustTensor:提出的增量式的基于主题的用户信任推荐算法,用来对传统推荐算法大都是静态算法的解决方案.对于 IncTrustTensor 算法,目标是快速对系统中新加入的数据进行训练.实验中,我们对训练集分成 4 份,即:每份占总数据的 20%,IncTrustTensor 以 20%的数据每次进行更新.

4.3 评估方法

- 有效性评估.

实验中,对于算法的有效性的评价方法用平均绝对误差(MAE)和均方根误差(RMSE)两种评价方式:

$$MAE=\frac{\sum_{(u,i)\in R_{test}}|r_{u,i}-\widehat{r_{u,i}}|}{|R_{test}|},$$

$$RMSE=\sqrt{\frac{\sum_{(u,i)\in R_{test}}(r_{u,i}-\widehat{r_{u,i}})^2}{|R_{test}|}},$$

其中,R_{test} 为测试集中所有的(u,i)集合,$r_{u,i}$ 为用户 u 对物品 i 的真实评分,$\widehat{r_{u,i}}$ 为预测用户 u 对物品 i 的评分.

- 效率评估.

对于算法的效率评估,我们对比不同算法的运行时间,目的是验证 IncTrustTensor 方法相对于其他静态算法在效率上的优势.在这部分实验中,为了模拟数据的增量计算,训练集均分为 4 部分,每次增量计算,我们对上次计算的数据集进行扩展,然后对新数据集进行计算,即要进行 4 次增量步骤完成对整个训练集的计算,每次计算的数据大小为训练集的 1/4,2/4,3/4,4/4.对于 STE 和 TrustTensor 方法,每次增量计算时,算法都要对新数据集重新计算得到结果,但是 IncTrustTensor 算法可以根据上次迭代的结果对新数据集进行更新操作,无需重新进行张量分解计算.

4.4 实验结果与分析

实验 1. 算法有效性比较.

实验比较了各种算法在不同的主题个数下的结果.设定了模型分解时候主题个数 K= 10,20,30,50.算法的其他参数设置为使各算法最优的时候的相应值.

图 7~图 12 是算法在不同稀疏程度的数据上的对比结果,在 Epinions,Epinions-1,Epinions-2 这 3 个数据集上,算法的 MAE 与 RMSE 随着主题数量的变大而变小,说明算法的准确性在提高;同时,算法在 Epinions,Epinions-1,Epinions-2 这 3 个数据集上的准确性随着数据的稀疏度变化而变化.

总体而言,TrustTensor 与 IncTrustTensor 的准确度在 Epinions 数据集上最低,而在 Epinions-2 数据集上的准确度最高,说明算法随着数据稠密度增加,算法的准确率会提高.

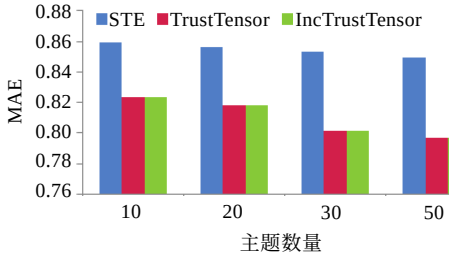


Fig.7 MAE of Epinions

图 7 Epinions 数据集的 MAE 结果

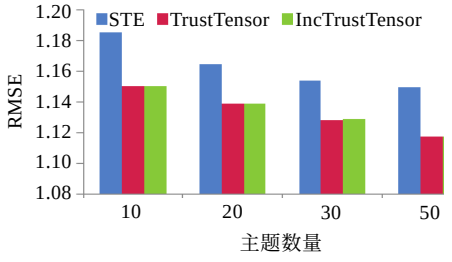


Fig.8 RMSE of Epinions

图 8 Epinions 数据集的 RMSE 结果

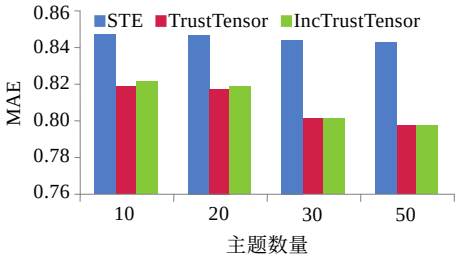


Fig.9 MAE of Epinions-1

图 9 Epinions-1 数据集 MAE 结果

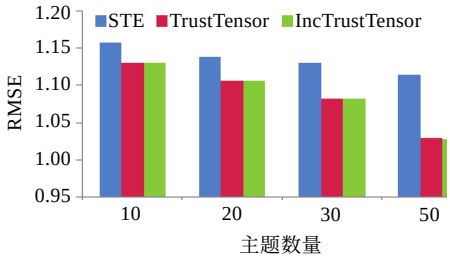


Fig.10 RMSE of Epinions-1

图 10 Epinions-1 数据集 RMSE 结果

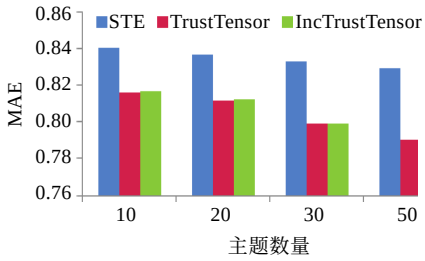


Fig.11 MAE of Epinions-2

图 11 Epinions-2 数据集 MAE 结果

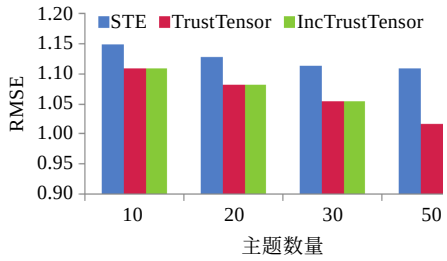


Fig.12 RMSE of Epinions-2

图 12 Epinions-2 数据集 RMSE 结果

从图 7~图 12 中可以看出:

1) 随着 K 的增加,算法精度都有一定程度的提高.但是,K 的增大会增加算法的运行时间;
2) STE 效果是最差的,这是因为 STE 方法只是将用户的信任程度和物品的评分做加权叠加,没有考虑到用户信任度在不同主题上的差异化;
3) 提出的 TrustTensor 方法和 IncTrustTensor 方法因为考虑了不同主题下用户对朋友的信任度,所以算法结果要好于其他没有分类考虑用户信任的算法.

实验证明:考虑了主题的 TrustTensor 和 IncTrustTensor 算法比 STE 有了很大的提高,充分说明了考虑不同主题下用户对朋友信任的有效性和合理性.

- 实验 2. 算法效率比较.

图 13~图 16 中为 3 个算法在不同的主题数量下不同的运行时间图(数据集为 Epinions).

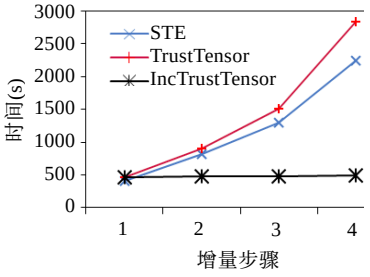


Fig.13 Run time of *Topic*=10

图 13 主题数量为 10 的运行时间

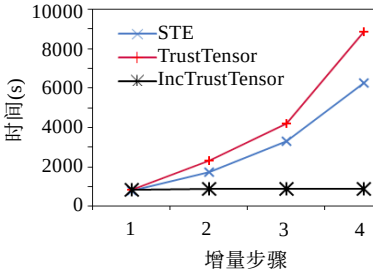


Fig.14 Run time of *Topic*=20

图 14 主题数量为 20 的运行时间

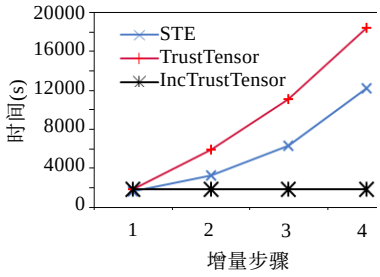


Fig.15 Run time of *Topic*=30

图 15 主题数量为 30 的运行时间

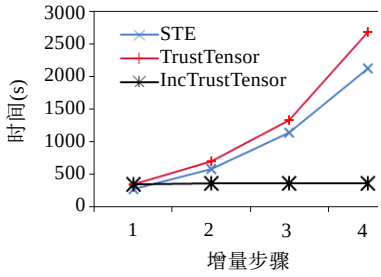


Fig.16 Run time of *Topic*=50

图 16 主题数量为 50 的运行时间

从图中可以看到:TrustTensor 的运行时间比 STE 要长,而且在 3 种算法的运行时间中是最长的.这是因为 TrustTensor 算法是基于三阶张量分解算法,而 STE 算法是二阶张量下的算法,三阶张量在分解时候的时间复杂度是高于二阶张量的.

我们提出的增量更新的张量分解算法 IncTrustTensor 在运行时间上远远小于其他两个算法,并且算法的运行时间不随数据量的变大而增长.这是因为 STE 算法和 TrustTensor 算法在每个增量步骤的时候要对原来数据和增量的数据而且合并后的新数据集重新做分解运算,所以算法时间会有较大增长;而 IncTrustTensor 在增量步骤的时候只需要对新增量的数据进行分解计算,这就保证了每个步骤的计算时间都保持在相对较低的水平,算法运行时间远远小于 STE 算法和 TrustTensor 算法.

5 总 结

本文首先对现有的基于用户信任的算法进行了总结,为了解决目前的基于信任的推荐算法都是单一信任模型的缺陷,本文提出了一种基于主题的张量分解的用户信任推荐算法,用来挖掘用户在不同的物品选取的时候对不同朋友的信任程度;由于社交网络更新速度快,鉴于目前的基于信任算法大都是静态算法,本文提出了一种增量更新的张量分解算法用于用户信任的推荐算法.最后,通过实验证明 TrustTensor 算法与 IncTrustTensor 算法比现有算法具有更好的准确性,并且 IncTrustTensor 算法可以大幅度提高推荐算法在训练数据增加后的模型训练效率,适合更新速度快的社会化网络中的推荐任务.

References:

[1] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowledge and Data Engineering, 2005,17:734−749. [doi: 10.1109/TKDE.2005.99]

[2] Wang LC, Meng XW, Zhang YJ. Context-Aware recommender systems. Ruan Jian Xue Bao/Journal of Software, 2012,23(1):1−20 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4100.htm [doi: 10.3724/SP.J.1001.2012.04100]

[3] Sherchan W, Nepal S, Paris C. A survey of trust in social networks. ACM Computing Surveys (CSUR), 2013,45(4):47. [doi: 10.1145/2501654.2501661]

[4] Ruohomaa S, Kutvonen L. Trust management survey. In: Proc. of the 3rd Int'l Conf. on Trust Management. Berlin: Springer-Verlag, 2005. 77−92. [doi: 10.1007/11429760_6]

[5] Singh S, Bawa S. A privacy, trust and policy based authorization framework for services in distributed environments. Int'l Journal of Computer Science, 2007,2(2):85−92.

[6] Granovetter M. The strength of weak ties. American Journal of Sociology, 1973,78(6):l360−1380. [doi: 10.2307/202051]

[7] Gilbert E, Karahalios K. Predicting tie strength with social media. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 2009. 211−220. [doi: 10.1145/1518701.1518736]

[8] Xiang R, Neville J, Rogati M. Modeling relationship strength in online social networks. In: Proc. of the 19th Int'l Conf. on World Wide Web. New York: ACM Press, 2010. 981−990. [doi: 10.1145/1772690.1772790]

[9] Zarghami A, Fazeli S, Dokoohaki N, Matskin M. Social trust-aware recommendation system: A *t*-index approach. In: Proc. of the 2009 IEEE/WIC/ACM Int'l Joint Conf. on Web Intelligence and Intelligent Agent Technology, Vol.3. Washington: IEEE Computer Society, 2009. 85−90. [doi: 10.1109/WI-IAT.2009.237]

[10] Jamali M, Ester M. TrustWalker: A random walk model for combining trust-based and item-based recommendation. In: Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2009. 397−406. [doi: 10.1145/1557019.1557067]

[11] Ma H, King I, Lyu MR. Learning to recommend with social trust ensemble. In: Proc. of the 32nd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2009. 203−210. [doi: 10.1145/1571941.1571978]

[12] Karatzoglou A, Amatriain X, Baltrunas L, Oliver N. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: Proc. of the 4th ACM Conf. on Recommender Systems. New York, ACM Press, 2010. 79−86. [doi: 10.1145/1864708.1864727]

[13] Kolda TG, Bader BW. Tensor decompositions and applications. SIAM Review, 2009,51(3):455−500. [doi: 10.1137/07070111X]

[14] De Lathauwer L, De Moor B, Vandewalle J. A multilinear singular value decomposition. SIAM Journal on Matrix Analysis and Applications, 2000,21(4):1253−1278. [doi: 10.1137/S0895479896305696]

[15] Tucker LR. Some mathematical notes on three-mode factor analysis. Psychometrika, 1966,31(3):279−311. [doi: 10.1007/BF02289464]

[16] Harshman RA. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. UCLA Working Papers in Phonetics, 1970,16:1−84.

[17] Carroll JD, Chang J. Analysis of individual differences in multidimensional scaling via an *N*-way generalization of "Eckart-Young" decomposition. Psychometrika, 1970,35(3):283−319. [doi: 10.1007/BF02310791]

[18] Bader BW, Kolda TG. Efficient MATLAB computations with sparse and factored tensors. SIAM Journal on Scientific Computing, 2007,30(1):205−231. [doi: 10.1137/060676489]

[19] O'Brien GW. Information management tools for updating an SVD-encoded indexing scheme [MS. Thesis]. Knoxville: University of Tennessee, 1994.

[20] Massa P, Avesani P. Trust-Aware bootstrapping of recommender systems. In: Proc. of the 2007 ACM Conf. on Recommender Systems. New York: ACM Press, 2006. 29−33. [doi: 10.1145/1297231.1297235]

附中文参考文献:

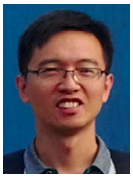
[2] 王立才,孟祥武,张玉洁.上下文感知推荐系统.软件学报,2012,23(1):1−20. http://www.jos.org.cn/1000-9825/4100.htm [doi: 10.3724/SP.J.1001.2012.04100]

邹本友(1987－),男,山东宁阳人,博士生,CCF 学生会员,主要研究领域为推荐系统,数据挖掘,大数据分析.
E-mail: zoubenyou@ruc.edu.cn

李翠平(1971－),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为推荐系统,大数据分析,数据仓库和数据挖掘,信息网络分析,流数据管理.
E-mail: licuiping@ruc.edu.cn

谭力文(1989－),男,硕士,主要研究领域为推荐系统,数据挖掘.
E-mail: tan1204@ruc.edu.cn

陈红(1965－),女,博士,教授,博士生导师,主要研究领域为高性能数据库系统,传感器网络数据管理,基于新硬件的数据管理与数据分析.
E-mail: chong@ruc.edu.cn



王绍卿(1981－),男,博士生,CCF 学生会员,主要研究领域为推荐系统,数据挖掘.
E-mail: wsq@ruc.edu.cn