基于时间的局部低秩张量分解的协同过滤推荐算法

孙艳歌[1,2] 王志海[1] 黄 丹[1]
(北京交通大学计算机与信息技术学院 北京 100044)[1]
(信阳师范学院计算机与信息技术学院 信阳 464000)[2]

摘 要 传统的推荐模型是静态的,忽略了时间因素。部分推荐算法虽然将时间因素考虑在内,但只是简单使用最近的数据或者降低过去数据的权重,这样可能会造成有用信息的丢失。针对这一问题,提出了一种考虑时间因素的局部低秩张量分解推荐算法。在传统的推荐算法的基础上,放松用户对项目的评分矩阵是低秩的这一假设,认为整个评分矩阵可能不是低秩的而是局部低秩的,即特定用户项目序偶的近邻空间是低秩的;同时又考虑时间因素,把评分矩阵看作是用户、项目和时间3个维度的张量,将传统的推荐算法延伸到张量领域。实验表明,所提算法能显著提升排名推荐性能。

关键词 推荐系统,时间因素,张量分解,局部低秩
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.07.040

Time-based Local Collaborative Filtering Recommendation Algorithm on Tensor Factorization

SUN Yan-ge[1,2] WANG Zhi-hai[1] HUANG Dan[1]
(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)[1]
(School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China)[2]

Abstract Traditional recommendation models are stationary with neglecting time factor. Some recommendation algorithms take time factor into consideration, but what they do is using the latest data or reducing the weight of past data. It may lead to the loss of some useful information. To solve the above problem, a time-based local low-rank tensor factorization algorithm was proposed. In contract to standard collaborative filtering algorithms, our method does not assume that the rating matrix is low-rank. We relaxed the assumption and assumed that the rating matrix is locally low-rank. The algorithm takes time factor into consideration and views rating matrix as 3-dimensional sensor based on the traditional recommendation algorithms which extend the traditional algorithms to tensor field. Experiments show that the algorithm could improve the efficiency of ranking recommendation.

Keywords Recommendation system, Time factor, Tensor factorization, Local low-rank

1 引言

国际数据公司IDC统计并发表报告称,2011年全球的数据总量为1.8ZB,这个数字每年将以9倍的速度递增[1]。我们已从信息匮乏时代迈入了信息过载时代。在这样的背景下,如何利用数据为用户提供个性化的推荐服务成为了一个研究热点问题。

协同过滤(Collaborative Filtering, CF)方法[2]善于发现用户的新兴趣点,且能够处理非结构化数据,是当前推荐算法研究的主要方向。基于记忆的协同过滤推荐算法是最早被提出的方法,但在数据稀疏的情况下,其推荐精度会很差;基于模型的协同过滤推荐算法具有推荐精度高、善于发现用户的新兴趣点、能够处理非结构化数据的优点,是目前推荐系统研究的热点方向。

然而,传统的推荐模型是静态的,忽略了时间因素对推荐系统的影响。虽然已有一些算法将时间因素考虑在内[3-5],但它们只是简单地使用最近的数据或者降低过去数据权重,这样可能会造成有用信息的丢失。为此,设计并实现了一种考虑时间因素的局部低秩张量分解的协同过滤推荐算法(Local Low-Rank Tensor Factorization, LLORTF)。本文主要贡献如下:

1)考虑到时间因素对用户兴趣和项目特征的影响,在建立推荐模型的过程中考虑时间因素;同时结合张量的概念,将推荐算法的研究延伸到张量领域。

2)放宽用户对项目的评分矩阵是低秩的这一假设,认为整个评分矩阵可能不是低秩的而是局部低秩的,即特定用户项目序偶的近邻空间是低秩的。

本文第2节介绍了相关研究工作;第3节介绍了基于时

到稿日期:2016-04-11 返修日期:2016-09-04 本文受国家自然科学基金(61672086),河南省科技计划项目(172102210454),信阳师范学院青年骨干教师计划(2016GGJS-08)资助。
孙艳歌(1982—),女,博士生,讲师,CCF会员,主要研究方向为数据挖掘、机器学习,E-mail:13112074@bjtu.edu.cn;王志海(1963—),男,博士,教授,博士生导师,主要研究方向为数据挖掘、推荐系统。

间的局部低秩张量分解的推荐算法;在第 4 节首先介绍了实验使用的数据集合和评价指标,然后对算法的影响因素进行分析并进行对比实验;最后对进一步的工作进行展望。

2 相关工作

Adomavicius 等人给出了推荐系统的形式化定义[6]:设 C 为所有用户的集合,s 为所有候选推荐项目的集合。设函数 u 可以计算对用户 c 而言项目 s 的推荐度,则推荐系统的目的就是为当前的用户找出那些推荐度最大的对象组成的集合,如式(1)所示:

$$r(s)=\arg\max_{s\in S} u(c,s) \tag{1}$$

协同过滤推荐算法根据用户对项目的偏好信息,研究用户或项目之间存在的相似性,再基于这些相似性进行推荐。

基于记忆的协同过滤推荐算法[7](又称基于近邻的协同过滤推荐算法)是最早被提出并使用的协同过滤推荐算法。该类型的推荐算法根据以往所有的用户与项目的相关信息进行推荐,大部分的基于记忆的协同过滤推荐算法根据用户对项目的评分进行推荐。在数据稀疏的情况下,基于记忆的协同过滤推荐算法的推荐精度会很差,这是由于在计算相似度的过程中需要基于用户拥有共同的评分项目或基于用户给相同的项目进行评分。随着用户数量和项目数量的增多,系统的可扩展性会变得越来越差,每次推荐的计算量会增大。

为了解决上述问题,人们提出了基于模型的协同过滤推荐算法[8-11]。此类方法根据已有的数据训练出推荐模型,然后在每次推荐的过程中根据已经建立好的模型进行计算,最后根据计算结果进行推荐。这样就不用频繁地调整数据库,加快了推荐的速度且增大了系统的可扩展性。Breese 等人[8]从概率的角度出发,认为推荐算法的目的是在已知用户项目评分矩阵的情况下计算特定用户对目标项目的评分期望,利用贝叶斯网络建立了协同过滤推荐算法。基于模型的协同过滤推荐算法的核心部分建立模型,除了以上提到的两种建模方式外,还有很多推荐效果不错的建模方式,例如最大熵模型[9]、统计模型[10]、线性回归模型[11]等。而在 Netflix 大赛之后,基于矩阵分解模型的协同过滤推荐算法凭借其出色的表现受到了广泛关注,成为当今推荐算法研究领域最热门的方向之一。

概率矩阵分解(Probabilistic Matrix Factorization,PMF)是矩阵分解模型的一个经典算法[12]。该算法将已知的用户对项目的评分看成是一个概率组合问题。Lee 等人[13]提出了基于矩阵分解模型的推荐算法(Local Low-Rank Matrix Approximation,LLORMA),该推荐算法放宽了评分矩阵是低秩的这一假设,认为整个评分矩阵是局部低秩的,即某个用户项目序偶的近邻空间是低秩的,评分矩阵可以由多个局部低秩的评分矩阵组合而成。Lee 等人[14]提出了一种矩阵分解和排名学习(Learn to Rank,LTR)算法[15]相结合的算法。刘海洋等人[16]将排名学习领域的知识引入推荐算法,设计了一种基于评分矩阵局部低秩假设的成列协同排名算法。

系统应该是动态变化的:项目的特性和受关注的程度会随着时间改变;与此同时,用户的兴趣也会随着时间变化。由此可见,时间因素对于推荐系统而言是至关重要的,然而以上算法都未考虑其带来的影响。Ding 等人[3]提出了一种改进的基于项目的协同过滤推荐算法。该算法在计算不同项目的权重时加入了时间权重,其中用来计算项目的时间权重的函数随着时间变化而单调递增。Gong 等人[4]从用户兴趣会随着时间变化的角度入手,改变了计算用户相似性的皮尔森系数的方法,考虑了时间因素,最近的用户项目序偶在计算用户相似度(皮尔森系数)的过程中所占权重较大,而过去的用户项目序偶在计算用户相似度的过程中所占权重较小。Lee 等人[5]提出了一种可应用于隐式反馈数据集合的有效的协同过滤推荐算法,其关键步骤是使用隐式反馈数据集合构造伪评分矩阵,在构造伪评分矩阵的过程中考虑了时间因素。

3 基于时间的局部低秩张量分解的协同过滤推荐算法

本文结合了两种协同过滤推荐算法的特点,同时在推荐算法中加入时间因素,用张量来描述具有 3 个维度的数据(用户维度、项目维度和时间维度),提出了一种基于时间的局部低秩张量分解的推荐算法;同时在算法中使用了全部数据,因此不会造成信息丢失。

具体来说,用三阶张量 R 表示 M 个用户对 N 个项目在 K 个时间片下的评分数据,其中有 $M\times N\times K$ 个数据项 R_{ijk},R_{ijk} 表示在时间片 k 下用户 i 对项目 j 的评分。为获得未知的用户对项目的评分,LLORTF 使用 CP 分解的方法[17]对 R 进行张量分解,最终得到 3 个低秩特征矩阵,即低秩用户特征矩阵 $U\in R^{M\times D}$、低秩项目特征矩阵 $V\in R^{N\times D}$、低秩时间特征矩阵 $T\in R^{K\times D}$,其中 $D\ll\min(M,N,K)$。CP 张量分解示意图如图 1 所示。

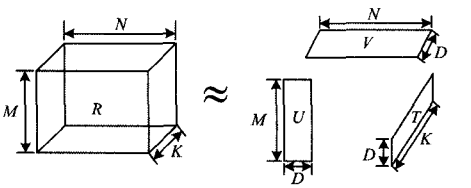

图 1 CP 张量分解示意图

对于第 t 个用户项目序偶,用户 u 对项目 i 在时间片 k 下的评分可以用分解得到的第 t 个用户项目序偶对应的 3 个低秩特征矩阵的行向量进行内积计算得到,如式(2)所示:

$$\langle U_u^t,V_i^t,T_k^t\rangle=\sum_{d=1}^{D}U_{ud}^tV_{id}^tV_{kd}^t \tag{2}$$

其中,U^t 表示被选中的第 t 个用户项目序偶近邻空间对应的低秩用户特征矩阵,V^t 表示被选中的第 t 个用户项目序偶近邻空间对应的低秩项目特征矩阵,T^t 表示被选中的第 t 个用户项目序偶近邻空间对应的低秩时间片特征矩阵,D 表示矩阵的秩。

通过 N-W 估计将选取的 q 个用户项目序偶插值点对应的 q 个低秩用户特征矩阵 U、低秩项目特征矩阵 V 和低秩时间特征矩阵 T 结合起来,得到最后的评分结果,如式(3)所示:

$$\hat{R}_{u,i,k}=\sum_{t=1}^{q}\frac{K((u_t,i_t),(u,i))}{\sum_{s=1}^{q}K((u_s,i_s),(u,i))}(\langle U_u^t,V_i^t,T_k^t\rangle) \tag{3}$$

LLORTF 采用最小化平方重构误差的方法,构建每个用户项目序偶 s 的近邻空间 N_s 的低秩用户特征矩阵 U、低秩项目特征矩阵 V 和低秩时间特征矩阵 T,其目标函数如式(4)所示:

$$F(U,V,T)=\sum_{(u,i)\in A}K((u_s,i_s),(u,i))(f_{uik}-R_{u,i,k})^2 \quad (4)$$

其中，用户项目序偶(u^*,i^*)表示插值点，A 表示所有的用户项目序偶集合，f_{uik} 表示在时间片 k 下 LLORTF 算法预测的用户 u 对项目 i 的评分，见式(3)。已知用户 u 和项目 i，就可以获得评分时间 k，因此 f_{ui} 和 f_{uik} 可以表示在时间片 k 下用户 u 对项目 i 的评分。每个被选中的用户项目序偶 s 都要进行该优化问题的求解，从而生成 q 个低秩用户特征矩阵 U、低秩项目特征矩阵 V 和低秩时间特征矩阵 T。

LLORTF 算法在计算核函数时，除了考虑用户之间的相似性和项目之间的相似性外，还考虑了时间因素，如式(5)所示：

$$K((u^*,i^*),(u,i))=K(u^*,u)K(i^*,i)T(u^*,i^*) \quad (5)$$

由式(5)可以看出，两个用户项目序偶的核函数由用户、项目和时间这 3 个平滑核函数的乘积得到，见式(6)和式(7)：

$$K_T(u_1,u_2)=(0,1-d(u_1,u_2)/h)1[d(u_1,u_2)<h] \quad (6)$$

$$K_E(u_1,u_2)=\frac{3}{4}(1-d(u_1,u_2)^2)1[d(u_1,u_2)<h] \quad (7)$$

其中，h 表示核函数的宽度，即被选中的 q 个用户项目序偶的近邻的大小。用户之间和项目之间的距离函数 $d(x_1,x_2)$ 的计算方法与 LLORMA 相同，可通过计算用户特征矩阵 U 第 i 行与第 j 行的规范化内积的反余弦函数值得到。对于项目 i 与项目 j 之间的距离，可通过计算项目特征矩阵 V 第 i 行与第 j 行的规范化内积的反余弦函数值得到。评分时间之间的距离函数的计算较为简单，将评分时间差归一化即可。

对于目标函数 $F(U,V,T)$，为了防止过拟合，要考虑正则化项，则目标函数如式(8)所示：

$$F(U,V,T)=\sum_{(u,i)\in A}K((u_s,i_s),(u,i))(f_{uik}-R_{u,i,k})^2-\frac{\lambda}{2}(\|U\|^2+\|V\|^2+\|T\|^2) \quad (8)$$

对于 q 个被选中的用户项目序偶插值点 s，为了得到其对应的低秩用户特征矩阵 U、低秩项目特征矩阵 V 和低秩时间特征矩阵 T，需要每个 s 对式(8)求最小值。首先使用随机梯度下降法求式(8)在最小值情况下的解，然后根据学习得到的 q 个用户项目序偶对应的低秩用户特征矩阵 U、低秩项目特征矩阵 V 和低秩时间特征矩阵 T 获得用户对项目的评分，最后依据所生成的评分产生推荐结果，如算法 1 所示。

算法 1

输入：训练集 R

参数：局部模型个数 q，低秩矩阵的秩 r，学习速率 γ，正则化系数 λ，最大迭代次数 itr

输出：$\hat{R}_{u^*,i^*,t^*}$

1. 对每一个 t=1,2,…,q

初始化 U^t,V^t,T^t

从 R 中随机选取用户项目序偶(u_t,i_t,k_t)

2. 对每一个用户项目序偶$(u,i)\in R$，计算

$$w_{u,i}=\sum_{t=1}^{q}K(u_t,u)K(i_t,i)T(u_t,i_t)$$

3. 重复

对每一个 t=1,2,…,q

对每一个用户 u=1,2,…,M，更新 U_u^t

$$U_u^{t(s+1)}=U_u^{t(s)}-\gamma\frac{\partial F}{\partial U_u^{t(s)}}$$

对每一个项目 i=1,2,…,N，更新 V_i^t

$$V_i^{t(s+1)}=V_i^{t(s)}-\gamma\frac{\partial F}{\partial V_i^{t(s)}}$$

对每一个时间片 k=1,2…,K，更新 T_k^t

$$T_k^{t(s+1)}=T_k^{t(s)}-\gamma\frac{\partial F}{\partial T_k^{t(s)}}$$

s=s+1

直至 s>itr 或 curErr>preErr

4. $$\hat{R}_{u^*,i^*,k^*}=\sum_{t=1}^{q}\frac{K(u_t,u^*)K(i_t,i^*)T(u_t,i_t)}{\sum_{s=1}^{q}K(u_s,u^*)K(i_s,i^*)T(u_s,i_s)}(U_u^{t*},V_i^{t*},T_k^{t*})$$

4　实验评价

4.1　数据集

实验选用两个用于电影评分的数据集合 MovieLens100K[1] 和 Netflix，这两个数据集在用户和项目上都表现出一定的时间特性。

MovieLens100K 中收集的是 1997 年 9 月 19 日到 1998 年 4 月 22 日期间在 MovieLens 网站上用户对电影的评分信息，包含了 943 个用户对 1682 部电影的 100000 个评分。此外，该数据集合还提供了用户的其他个人信息，包括用户的年龄、性别、职业和邮编等。

Netflix 来自于电影租赁网站 Netflix 的数据库，包含了 1998 年到 2005 年间 480189 个匿名用户对约 17770 部电影给出的超过 10 次亿次评分，其中用户对电影的评分数值的取值范围为 1～5 之间的整数。从 Netflix 中抽取了部分数据来进行对比实验。

4.2　评价指标

实验中采用以下 3 个被广泛应用的评价指标。

(1)倒数排名函数均值

倒数排名函数均值(Mean Reciprocal Rank，MRR)是所有用户的倒数排名函数 RR 的均值，其定义见式(11)：

$$RR_u=\sum_{i\in T_u}\frac{y_{u,i}}{L_{u,i}}\prod_{j\in T_u}(1-y_{u,j}I(L_{u,j}<L_{u,i})) \quad (9)$$

$$RR=\sum_{u\in U}RR_u \quad (10)$$

$$MRR=\frac{1}{|U|}RR \quad (11)$$

其中，RR_u 为用户的评价指标倒数排名函数，T_u 表示测试集合中用户 u 给出评分的项目的集合。

(2)平均精度均值

平均精度均值(Mean Average Precision，MAP)是反映推荐算法在全部相关推荐项目上推荐性能的单值指标。用户相关的项目在推荐序列中的排名越靠前，MAP 值就越大，如式(12)所示：

$$MAP=\frac{1}{|U|}\sum_{u\in U}\frac{1}{\sum_{i\in T_u}y_{u,i}}\sum_{i\in T_u}\frac{y_{u,i}}{L_{u,i}}\sum_{j\in T_u}y_{u,j}I(L_{u,j}<L_{u,i}) \quad (12)$$

(3)正规化折算累积收益

正规化折算累积收益(Normalized Discounted Cumulative

[1] MovieLens100K 可从 http://grouplens.org/datasets/获取。

Gain,NDCG)是依据评价指标折算累积收益(Discounted Cumulative Gain,DCG)定义的。一般使用 NDCG@k 表示,其中 k 表示每个用户给出的推荐序列中的项目个数。用户评分高的项目排名越靠前,该指标值越大。对用户 u 而言,其DCG@k 和 NDCG@k 分别见式(13)和式(14):

$$\mathrm{DCG@}k=\sum_{i=1}^{k}\frac{2^{R_{u,i}}-1}{\log_2(i+1)} \tag{13}$$

$$\mathrm{NDCG@}k=\frac{1}{DCG_{\max}}\mathrm{DCG@}k \tag{14}$$

4.3 实验设置及结果分析

本文构建了用 Java 实现的推荐系统实验平台。该平台由基本数据结构、推荐算法和工具方法 3 部分组成。此外,平台还提供了自己的用户界面。

4.3.1 参数对算法性能的影响

本文在 CPU 为 2.8GHz、内存为 8GB、操作系统为 Windows 7 的 PC 机上进行实验。将数据集合的 50%作为训练集合,另外 50%作为测试集合,实验 10 次并且取其平均作为最终的实验结果。首先固定局部模型个数为 10,设置张量的秩的变化范围为{10,20,40}。结果如图 2—图 4 所示,其中横轴表示迭代次数,纵轴为评价指标。

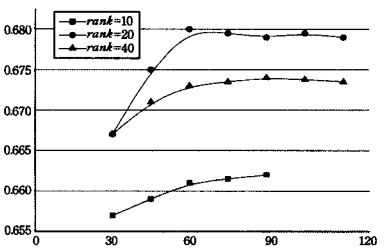

图 2 张量的秩对 MRR 的影响

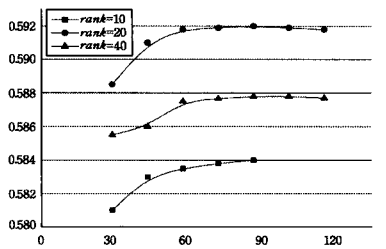

图 3 张量的秩对 MAP 的影响

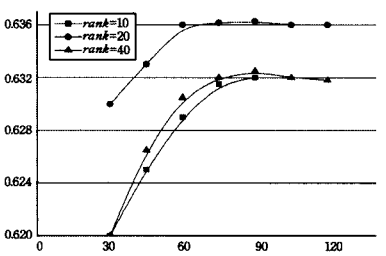

图 4 张量的秩对 NDCG@10 的影响

从图 2—图 4 可以看出,在固定局部模型个数的前提下,MRR,MAP 和 NDCG@k 的值在张量的秩等于 20 时表现最好,在张量的秩等于 10 时表现最差,而在张量的秩等于 40 时出现了过拟合的现象。随着迭代次数的增加,MRR,MAP 和 NDCG@10 的值在整体上逐渐增加,但是在个别点上却显示出了不同的趋势。如图 3 所示,MAP 在迭代次数为 120 时较迭代次数为 90 时变低,可见它们在该点出现了过拟合现象。因此在实际的运算过程中,要应用交叉验证的方法使迭代提前结束,以防止产生过拟合现象。总之,设置合适的张量的秩对提高推荐性能很重要。

接着,将张量的秩设为 20,局部模型个数的取值范围为{10,20,50}。结果如图 5—图 7 所示,可以看出,所有的评价指标均会随着局部模型个数的增加而增加。同时还发现,随着迭代次数的增加,指标的值均出现了过拟合的现象。总体来说,局部模型个数越多,推荐性能越好,但是应该在推荐过程中使用交叉验证的方法提早结束迭代,以防止出现过拟合现象。

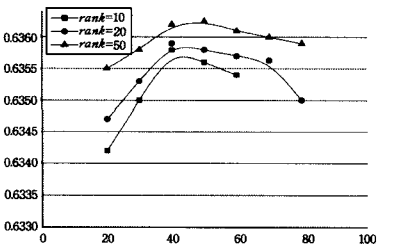

图 5 局部模型个数对 MRR 的影响

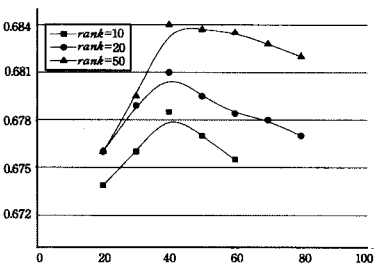

图 6 局部模型个数对 MAP 的影响

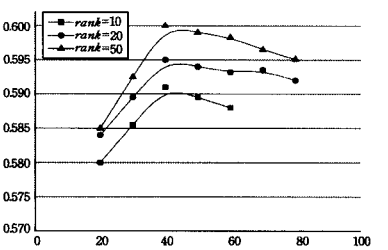

图 7 局部模型个数对 NDCG@10 的影响

4.3.2 对比实验

本文算法与以下 4 个算法进行对比:PMF[12],BPMF[18],RSVD 和 LLORMA[13]。其中,RSVD 为基于排名的协同过滤推荐算法;PMF,BPMF 和 LLORMA 为基于矩阵分解的协同过滤推荐算法。

实验中将张量的秩设为 20,局部模型的个数设为 50。采用随机梯度上升的方法来优化目标函数。在优化过程中,使用交叉验证的方式判断该次迭代是否有所提升(对训练集数据的 20%进行交叉验证),如果有提升则继续进行迭代直至达到最大的迭代次数(设为 100),反之结束优化过程。对于 LLORTF 算法,使用 N-W 估计计算最后的结果,采用 Epanechnikov核函数,宽度为 0.8。对于 LLORMA 算法,局部模型个数设为 50,采用 Epanechnikov 核函数,宽度为 0.8。所有算法的学习速率 γ 和正则化系数 λ 的值均通过交叉验证获得。其中,推荐算法 LLORTF 与 LLTFRRT 将学习速率 γ 设

置为 0.001，将正则化系数 λ 设置为 0.01。在每个数据集上运行 10 次，并且取其平均结果作为最终的结果，如表 1—表 2 所列。表中用黑色粗体表示 5 个算法在相应的数据集合的相应评价指标上表现最好的结果。

表 1　MovieLens100K 数据集的实验结果

	MRR	MAP	NDCG@10
PMF	0.6790±0.0002	0.5927±0.0010	0.6308±0.0006
BPMF	0.6733±0.0010	0.5901±0.0011	0.6310±0.0010
RSVD	0.6829±0.0003	0.5870±0.0005	0.6361±0.0012
LLORMA	0.6615±0.0015	0.5924±0.0002	0.6351±0.0010
LLORTF	**0.6834±0.0001**	**0.5936±0.0003**	**0.6370±0.0001**

表 2　Netflix 数据集的实验结果

	MRR	MAP	NDCG@10
PMF	0.7017±0.0008	0.6160±0.0002	0.6821±0.0003
BPMF	0.7006±0.0004	0.6183±0.0005	0.6827±0.0005
RSVD	0.6963±0.0003	0.6150±0.0002	0.6848±0.0006
LLORMA	0.7045±0.0006	0.6181±0.0005	**0.6872±0.0004**
LLORTF	**0.7056±0.0002**	**0.6198±0.0001**	0.6847±0.0002

从表 1 和表 2 中可以看出，LLORTF 相较于 LLORMA 在排名推荐方面有一定的提升，在 2 个数据集合、3 个评价指标的 6 种情况下，LLORTF 在 5 种情况下的表现是优于 LLORMA 的。由此可以看出，时间因素是影响推荐的一个重要的因素。同时我们也观察到，LLORTF 算法在两个数据集合 MovieLens100K 和 Netflix，3 个排名评价指标 MRR，MAP 和 NDCG@10 下，相比其他算法都具有明显的优势，这也正说明了直接优化排名这一建模方法很适合排名推荐算法。

结束语　针对传统的协同过滤推荐算法忽略了时间因素的缺点，本文提出了考虑时间因素的局部低秩张量分解推荐算法。通过分析实验验证了所提推荐算法相较于其他推荐算法在排名推荐上有一定的提升。未来的工作可以从以下两个方面进行考虑：1）全面利用用户项目信息。除了考虑时间因素外，还能获得很多其他的用户行为数据，如点击、浏览、评价等。今后的研究工作可以将自然语言处理与用户项目时间三维张量相结合，从而分析出用户对项目评价中的喜好信息来丰富用户项目时间三维张量。2）推荐系统的实时性。本文算法虽然考虑了时间因素，但是并不能做到对用户行为的实时感知。由此，如何实现推荐系统的实时性仍是亟待解决的问题。

参 考 文 献

[1] GANTZ J, REINSEL D. IDC: The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east [OL]. http://www.emc.com/leadship/digital-universe/2012view/index.htm.

[2] SU X, KHOSHGOFTAAR T M. A survey of collaborative filtering techniques [J]. Advances in Artificial Intelligence, 2009, 2009(4): 1-19.

[3] DING Y, LI X. Time weight collaborative filtering [C] // Proceedings of the 14th ACM International Conference on Information and Knowledge Management. New York, USA: ACM, 2005: 485-492.

[4] GONG S J, CHENG G H. Mining user interest change for improving collaborative filtering [C] // Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application. Washington, USA: IEEE Computer Society, 2008: 24-27.

[5] LEE T Q, PARK Y, PARK Y T. A time-based approach to effective recommender systems using implicit feedback [J]. Expert Systems with Applications, 2008, 34(4): 3055-3062.

[6] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions [J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6): 734-749.

[7] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms [C] // Proceedings of the 10th International Conference on World Wide Web. New York, USA: ACM, 2001: 285-295.

[8] BREESE J S, HECKERMAN D, KADIE C. Empirical analysis of predictive algorithms for collaborative filtering [C] // Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. San Francisco, USA: Morgan Kaufmann Publishers, 1998: 43-52.

[9] PAVLOV D, PENNOCK D. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains [C] // Proceedings of the 16th Annual Conference on Neural Information Processing Systems. MIT Press, 2002: 1441-1448.

[10] ZHANG J W, YANG Z. Collaborative filtering recommendation algorithm based on improved user clustering [J]. Computer Science, 2014, 41(12): 176-178. (in Chinese)
张峻玮，杨洲. 一种基于改进的层次聚类的协同过滤用户推荐算法研究 [J]. 计算机科学，2014，41(12)：176-178.

[11] YIN H, CUI B, SUN Y, et al. LCARS: A Spatial Item Recommender System [J]. ACM Transactions on Information Systems (TOIS), 2014, 32(3): 1-37.

[12] SALAKHUTDINOV R, MNIH A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo [C] // International Conference on Machine Learning. ACM, 2008: 880-887.

[13] LEE J, KIM S, LEBANON G, et al. Local low-rank matrix approximation [J]. Journal of Machine Learning Research, 2013, 28(2): 82-90.

[14] LEE J, BENGIO S, KIM S, et al. Local collaborative ranking [C] // Proceedings of the 23rd International Conference on World Wide Web (WWW 2014). Springer, 2014: 85-96.

[15] LIU T Y. Learning to rank for information retrieval [J]. Foundations and Trends in Information Retrieval, 2009, 3(3): 225-331.

[16] LIU H Y, WANG Z H, HUANG D, et al. Listwise Collaborative Ranking Based on the Assumption of Locally Low-Rank Rating Matrix [J]. Journal of Software, 2015, 26(11): 2981-2993. (in Chinese)
刘海洋，王志海，黄丹，等. 基于评分矩阵局部低秩假设的成列协同排名算法 [J]. 软件学报，2015，26(11)：2981-2993.

[17] KOLDA T G, BADER B W. Tensor decompositions and applications [J]. SIAM Review, 2009, 51(3): 455-500.

[18] SALAKHUTDINOV R, MNIH A. Bayesian probabilistic matrix factorization using markov chain monte carlo [C] // Proceedings of the 25th International Conference on Machine Learning. New York, USA: ACM, 2008: 880-887.