Journal of System Simulation

Volume 31 | Issue 7 Article 17

12-12-2019

# Personalized Music Recommendation Algorithm TFPMF

Xining Ye
*College of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China;*

Wang Meng
*College of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China;*

# Personalized Music Recommendation Algorithm TFPMF


## Abstract

Abstract: In recent years, the personalized context-aware recommendation is the rub and hotness in the research of recommendation system, and the data sparseness is the main problem faced by the current recommendation systems. *In the setting of music recommendation, the representing method of varieties of situational information is improved. A model of TFPMF is proposed, which combines the model of RR-PMF with the tensor decomposition. TFPMF is optimized by alternative least squares (ALS).* By the simulation experiments in the last.fm dataset, we got the TOP-N recommended list through the simulation program. The simulation results show that the proposed algorithm has great advantages in the evaluation index of Precision, Recall and NDCG, and the algorithm can effectively alleviate the data sparsity problem.



## Keywords

recommendation system, reciprocal rank, probabilistic matrix factorization, tensor factorization, TFPMF (Tensor Factorization-based Probabilistic Matrix Factorization)



## Recommended Citation

Ye Xining, Wang Meng. Personalized Music Recommendation Algorithm TFPMF[J]. Journal of System Simulation, 2019, 31(7): 1397-1407.

第 31 卷第 7 期 系统仿真学报© Vol. 31 No. 7
2019 年 7 月 Journal of System Simulation Jul., 2019

# 音乐个性化推荐算法 TFPMF 的研究

叶西宁，王猛
(华东理工大学信息科学与工程学院，上海 200237)

**摘要：**基于情境感知的个性化推荐是近年来推荐系统中的研究热点和难点问题，数据稀疏是当前推荐系统面临的主要问题。以音乐推荐为背景，*改进了多种情境信息的表示方法，将优化排名倒数(RR)的概率矩阵分解模型(RR-PMF)与张量分解相结合，提出了张量概率矩阵分解模型(TFPMF)，并使用交叉最小二乘法(ALS)优化该模型。*使用 last.fm 数据集进行仿真实验，通过仿真模型得出 TOP-N 推荐列表，结果表明该算法在准确率(Precision)、召回率(Recall)和标准化折算累加值(NDCG)评价指标上具有很大的优势，该算法能够有效缓解数据稀疏问题。

**关键词：**推荐系统；排名倒数；概率矩阵分解；张量分解；TFPMF(基于张量分解的概率矩阵分解)
中图分类号：TP391 文献标识码：A 文章编号：1004-731X (2019) 07-1397-11
DOI: 10.16182/j.issn1004731x.joss.17-0256

## Personalized Music Recommendation Algorithm TFPMF

*Ye Xining, Wang Meng*
(College of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)

**Abstract:** In recent years, the personalized context-aware recommendation is the rub and hotness in the research of recommendation system, and the data sparseness is the main problem faced by the current recommendation systems. *In the setting of music recommendation, the representing method of varieties of situational information is improved. A model of TFPMF is proposed, which combines the model of RR-PMF with the tensor decomposition. TFPMF is optimized by alternative least squares (ALS).* By the simulation experiments in the last.fm dataset, we got the TOP-N recommended list through the simulation program. The simulation results show that the proposed algorithm has great advantages in the evaluation index of Precision, Recall and NDCG, and the algorithm can effectively alleviate the data sparsity problem.

**Keywords:** recommendation system; reciprocal rank; probabilistic matrix factorization; tensor factorization; TFPMF (Tensor Factorization-based Probabilistic Matrix Factorization)

## 引言

作为解决信息超载(Information Overload)问题的重要工具，推荐系统(Recommendation System)近年来得到了快速发展。由于人们的兴趣偏好会随着时间、位置等情境的变化而发生改变，因此如何根据情境信息为用户提供个性化的推荐服务成为了近年来研究的热点问题。

情境感知推荐可以被划分为情境预过滤(Pre-Filtering)、情境后过滤(Post-Filtering)和情境建模(Contextual Modeling)[1]。早期情境感知推荐的研究主要是在情境预过滤方面[2]，根据情境信

收稿日期：2017-05-26 修回日期：2017-08-23;
基金项目：国家自然科学基金(61304071);
作者简介：叶西宁(1968-)，女，陕西蒲城，博士，副教授，研究方向为模式识别、数据挖掘、嵌入式系统应用；王猛(1991-)，男，河南沈丘，硕士，研究方向为数据挖掘、模式识别。



息将数据集划分为不同的数据块，然后运用传统的推荐方法进行推荐。最近的研究工作主要集中在情境建模方面，通过模型提取用户的偏好特征并进行推荐，如比较流行的张量分解(Tensor Factorization，TF)[3]和拓展传统矩阵分解(Matrix factorization，MF)的情境感知模型[4-6]；但是大部分的研究都是基于显式反馈的情境感知推荐[6-7]，利用隐式反馈的研究相对较少。隐式反馈具有收集成本低、不易引起用户反感等优点，在实际应用中具有很大优势，但也具有高噪声、缺少负反馈等缺点。在利用隐式反馈数据的研究中，Shi Y 等[8]提出一种张量分解(TF)模型 TFMAP 用于训练隐式反馈数据，该模型通过直接最大化平均准确率 MAP(Mean Average Precision)生成 TOP-N 推荐列表，作者使用随机选的数据作为负反馈，不可避免地会引入干扰，对实验会产生一定的影响，且算法时间复杂度较大，在实际应用中具有一定的局限性，实际应用中需要充分重视算法的运算速度。Nguyen 等[9]提出了一种基于高斯过程的非线性协同过滤算法 GPFM (Gaussian Processes Factorization Machines)，使用相关和非相关数据构建基于 GPFM 的成对偏好模型 GPPW(GPFM-based Pairwise Preference Model)进行推荐；但是为了解决隐式反馈数据缺少负反馈的问题，作者同样采用了随机选择数据作为负反馈，且计算需要较大的内存空间，在大数据量的工业环境下需要优化。

基于隐式反馈的情景感知推荐算法都是随机选择数据作为负反馈，导致算法计算速度慢，而且推荐效果较差。本文针对该问题从以下几个方面提出解决方法并设计如下仿真实验：

1) 针对情境感知推荐和隐式反馈数据的特点，提出了一种新的情境信息的表示方法，使多维情境信息转化为一维情境信息，使信息表示易于计算，减小计算量；

2) 针对情境感知推荐中数据稀疏、维度较高等问题，使用排名倒数平滑表示隐式反馈数据，提出一种与张量分解相结合的概率矩阵分解模型 TFPMF(Tensor Factorization-based Probabilistic Matrix Factorization)；

3) 根据建立的模型设计仿真程序，并在不同用户数量的 Last.fm 数据集上进行仿真实验，以验证本文提出算法的效果。

# 1 相关算法

## 1.1 基于排名倒数的概率矩阵分解算法

基于排名倒数(Reciprocal Rank)的概率矩阵分解是利用隐式反馈信息，在排名倒数的基础上结合概率矩阵分解处理排名倒数矩阵，其公式如下[10]：

$$P(\boldsymbol{RR}\,|\,\boldsymbol{U},\boldsymbol{V},\sigma^2)=\prod_{u=1}^{M}\prod_{i=1}^{N}[N(\boldsymbol{RR}_{ui}\,|\,\boldsymbol{U}_u^T\boldsymbol{V}_i,\sigma^2)]^{I_{ui}} \quad (1)$$

式中：$\boldsymbol{RR}$ 为用户-项目排名倒数矩阵；$\boldsymbol{U}$ 为 $M\times K$ 的用户特征矩阵；$\boldsymbol{V}$ 为 $N\times K$ 的项目特征矩阵；$M$ 为用户的个数；$N$ 为项目个数；$I_{ui}$ 为指示函数，表示用户 $u$ 播放过项目 $i$。

对于用户、项目的先验分布，我们假设为均值为 0 的高斯分布，如下：

$$P(\boldsymbol{U}\,|\,\sigma_U^2)=\prod_{u=1}^{M}[N(\boldsymbol{U}_u\,|\,0,\sigma_u^2\boldsymbol{I})] \quad (2)$$

$$P(\boldsymbol{V}\,|\,\sigma_V^2)=\prod_{i=1}^{N}[N(\boldsymbol{V}_i\,|\,0,\sigma_i^2\boldsymbol{I})] \quad (3)$$

式中：$\sigma_U^2$ 和 $\sigma_V^2$ 分别表示 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 的分布方差；$\boldsymbol{I}$ 表示单位矩阵。

## 1.2 张量分解

设 $N$ 阶张量 $\boldsymbol{X}$ 是一个秩一张量，则它可被写成 $N$ 个向量的外积，如一个三阶秩一张量可以写为 $\boldsymbol{X}=\boldsymbol{a}\circ\boldsymbol{b}\circ\boldsymbol{c}$，其模型如图 1 所示[11]，对于 $\boldsymbol{X}$ 的元素$(i, j, k)$的值可以表示为 $\boldsymbol{X}_{ijk}=\boldsymbol{a}_i*\boldsymbol{b}_j*\boldsymbol{c}_k$。

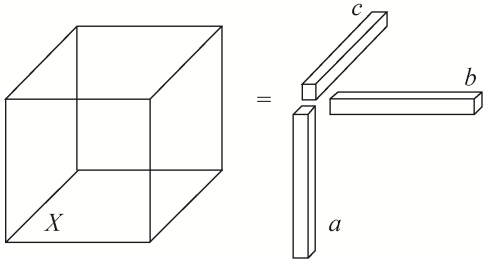



图 1 三阶秩一张量模型

Fig. 1 Third-order rank-one tensor model

张量分解将传统的二维矩阵拓展到高维，在情境感知的推荐系统中应用非常广泛。

## 2 基于排名倒数的 TFPMF 算法

在本节中，首先改进情境信息的表示方法，然后给出模型所需的定义，接着根据情境信息的表示方法将张量分解与概率矩阵分解相结合提出张量概率矩阵分解模型，最后描述算法的推导及实现。

### 2.1 情境信息的表示

以往情境信息的常用表示方法为 < userId, itemId, context_1, ⋯, context_i, ⋯, context_N > ，其中 userId 表示不同用户的 Id 编号，itemId 表示不同项目的 Id 编号，context_i (i = 1⋯$N$) 分别表示不同的情境类型，因此使用张量分解去进行计算时需要使用 $N$+2 维的模型去处理，当维数过高时计算起来会比较困难；比如多维时间情境信息使用如下方法表示 < userId, itemId, context_day, context_week > ，其中 context_day = (morrning, afternoon, night) 表示每天的时间情境，context_week = (workday, weekend) 表示工作日情况的时间情境，这时使用张量分解需要处理四阶张量。

不同于以上方法，本文将所有的情境信息都在一维向量中进行表示，从而使多维情境向量转化为一维情境向量，降低需要处理的张量维度。例如使用 < userId, itemId, Context > 的形式表示情境信息，其中 Context = ( (morrning, workday), (afternoon, workday), ⋯, (night, weekend)) ，因此只需使用三阶张量进行处理，这样更加方便、有效。

### 2.2 基本定义

为了方便后面的推导计算，本文首先定义一些公式，若向量 $\boldsymbol{a}$、$\boldsymbol{b}$、$\boldsymbol{c}$ 都是维度为 $K$ 的向量，则定义三向量内积为如下公式：

$$<\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}>=\sum_{i=1}^{K} a_i b_i c_i \tag{4}$$

若特征矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 分别为 $M\times K$ 和 $N\times K$ 的矩阵，则定义矩阵之间的 $\odot$ 运算如公式(5)所示：

$$\boldsymbol{U}\odot\boldsymbol{V}=\boldsymbol{Z}_{MN\times K}=\begin{bmatrix}\boldsymbol{U}_1*\boldsymbol{V}\\ \vdots\\ \boldsymbol{U}_i*\boldsymbol{V}\\ \vdots\\ \boldsymbol{U}_M*\boldsymbol{V}\end{bmatrix} \tag{5}$$

式中：$\boldsymbol{U}_i$ 表示矩阵 $\boldsymbol{U}$ 的第 $i$ 行的行向量；$\boldsymbol{U}_i*\boldsymbol{V}$ 表示矩阵 $\boldsymbol{U}$ 的第 $i$ 行的行向量与矩阵 $\boldsymbol{V}$ 的行向量叉乘，得到一个 $N\times K$ 的矩阵，即若 $\boldsymbol{U}_i=[U_{i1},U_{i2},\cdots,U_{iK}]$，则

$$\boldsymbol{U}_i*\boldsymbol{V}=\begin{bmatrix}U_{i1}*V_{11} & U_{i2}*V_{12} & \cdots & U_{iK}*V_{1K}\\ U_{i1}*V_{21} & U_{i2}*V_{22} & \cdots & U_{iK}*V_{2K}\\ \vdots & \vdots & \ddots & \vdots\\ U_{i1}*V_{N1} & U_{i2}*V_{N2} & \cdots & U_{iK}*V_{NK}\end{bmatrix}$$

若特征矩阵 $\boldsymbol{U}$、$\boldsymbol{V}$ 和 $\boldsymbol{C}$ 分别为 $M\times K$、$N\times K$ 和 $C\times K$ 的矩阵，为了方便计算，则定义如下 3 个特征矩阵之间的内积运算如公式(6)所示：

$$<\boldsymbol{U},\boldsymbol{V},\boldsymbol{C}>=\boldsymbol{U}(\boldsymbol{V}\odot\boldsymbol{C})^T,<\boldsymbol{V},\boldsymbol{U},\boldsymbol{C}>=\boldsymbol{V}(\boldsymbol{U}\odot\boldsymbol{C})^T \tag{6}$$

### 2.3 基于排名倒数的张量概率矩阵分解模型

根据特定情境为用户提供个性化推荐是推荐系统的发展趋势，能够给用户带来更好的体验。而在基于情境感知的个性化推荐系统中，由于引入了情境信息，原来的数据会变得更加稀疏，因此进行 TOP-N 推荐会变得更加困难，情境感知预过滤推荐效果相对较差。为了解决数据稀疏的问题，本文将概率矩阵分解模型与张量分解相结合，提出一种基于排名倒数的张量概率矩阵分解模型(TFPMF)，具体如图 2 所示。

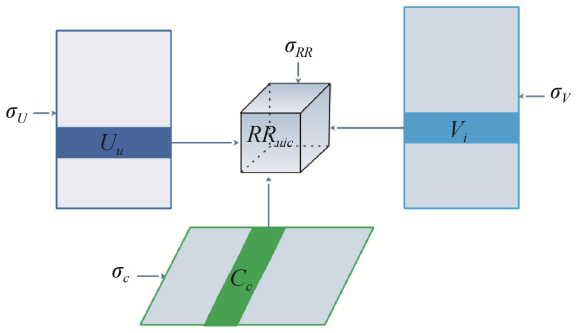



图 2 张量概率矩阵分解模型
Fig. 2 TFPMF Model

在该模型中，$\boldsymbol{U}_u$ 表示用户 $u$ 的特征向量，$\boldsymbol{V}_i$ 表示项目 $i$ 的特征向量，$\boldsymbol{C}_c$ 表示情境 $c$ 的特征向量，$R_{uic}$ 表示项目 $i$ 在情境 $c$ 下在用户 $u$ 播放列表中的预测排名倒数，在该模型中其计算值为上述 3 个向量的内积，即 $R_{uic}=<\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{C}_c>$。

由于隐式反馈并不像显式反馈那样由用户提供对项目的主观量化评价，存在有大量的噪声，因此根据隐式反馈数据的特点，本文使用公式(7)平滑表示排名倒数(RR)，定义项目 $i$ 在用户 $u$ 的列表中的排名倒数为：

$$R_{uic}\approx f(A_{uic}) \tag{7}$$

式中：$f(x)=x/(1+x)$，该函数的曲线图如图 3 所示。

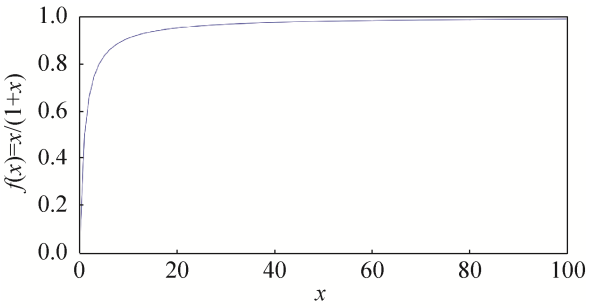



图 3 函数 $f(x)$的曲线图
Fig. 3 Curve of function $f(x)$

当 $x$ 大于 0 时，该函数的取值范围为[0，1]，随着 $x$ 值的增加而增加，即 $x$ 的值越大，排名倒数的值越大，排名越靠前。

由于用户的活跃度不同，有的用户比较活跃，其收听的音乐的频次大部分都比较高，而活跃度低的用户的收听频次比较低，因此考虑到用户之间的差异，为了抵消这些因素带来的影响，本文采用一种相对选择倾向程度计算排名倒数：

$$A_{uic}=\frac{S_{uic}}{\overline{S_{uc}}} \tag{8}$$

式中：$S_{uic}$ 表示用户 $u$ 在情境 $c$ 下对项目 $i$ 的播放次数；$\overline{S_{uc}}$ 表示用户 $u$ 在情境 $c$ 下对所有项目的平均播放次数。

在模型的推荐计算中，用户排名倒数矩阵的计算公式为：

$$\boldsymbol{RR}_{UVC}=<\boldsymbol{U},\boldsymbol{V},\boldsymbol{C}>=\boldsymbol{U}(\boldsymbol{V}\odot\boldsymbol{C})^T \tag{9}$$

式中：$\boldsymbol{RR}_{UVC}$ 为 $M\times NS$ 维的用户-项目情境排名倒数矩阵；$\boldsymbol{U}$ 为 $M\times K$ 的用户特征矩阵；$\boldsymbol{V}$ 为 $N\times K$ 的项目特征矩阵；$\boldsymbol{C}$ 为 $S\times K$ 的情境特征矩阵；$M$ 为用户的个数；$N$ 为项目个数；$S$ 为选取的情境个数；$K$ 表示选取的隐含特征个数。

对于排名倒数 $\boldsymbol{RR}$ 中元素$(u,i,c)$的值表示为：

$$RR_{uic}=<\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{C}_c> \tag{10}$$

式中：$<\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{C}_c>$ 表示三向量的内积，即 $RR_{uic}=\sum_{k=1}^{K}U_{uk}V_{ik}C_{ck}$。

定义与张量分解相结合的条件概率公式如下：

$$P(\boldsymbol{RR}\,|\,\boldsymbol{U},\boldsymbol{V},\boldsymbol{C},\sigma^2)=\prod_{u=1}^{M}\prod_{i=1}^{N}\prod_{c=1}^{S}[N(RR_{uic}\,|<\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{C}_c>,\sigma^2)]^{I_{uic}} \tag{11}$$

式中：$I_{uic}$ 为指示函数，表示用户 $u$ 在情境 $c$ 下播放过项目 $i$。

通过最大化用户和项目特征矩阵相对评分与固定参数(观察误差方差和先验方差)的后验概率的对数来学习特征矩阵参数，对后验概率公式(11)取对数可近似获得：

$$\begin{aligned}&P(\boldsymbol{U},\boldsymbol{V},\boldsymbol{C}\,|\,\boldsymbol{RR},\sigma^2,\sigma_U^2,\sigma_V^2,\sigma_C^2)=-\frac{1}{2\sigma^2}\\&\sum_{u=1}^{M}\sum_{i=1}^{N}\sum_{c=1}^{S}I_{uic}(RR_{uic}-<\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{C}_c>)^2-\frac{1}{2\sigma_U^2}\\&\sum_{u=1}^{M}\boldsymbol{U}_u^T\boldsymbol{U}_u-\frac{1}{2\sigma_V^2}\sum_{i=1}^{N}\boldsymbol{V}_i^T\boldsymbol{V}_i-\frac{1}{2\sigma_C^2}\sum_{c=1}^{S}\boldsymbol{C}_c^T\boldsymbol{C}_c-\\&\frac{1}{2}((\sum_{u=1}^{M}\sum_{i=1}^{N}\sum_{c=1}^{S}I_{uic})\ln\sigma^2+MK\ln\sigma_U^2+\\&NK\ln\sigma_V^2+SK\sigma_C^2)+\mathrm{CO}\end{aligned} \tag{12}$$

式中：$((\sum_{u=1}^{M}\sum_{i=1}^{N}\sum_{c=1}^{S}I_{uic})\ln\sigma^2+MK\ln\sigma_U^2+NK\ln\sigma_V^2+SK\sigma_C^2)$ 为固定值；CO 是与参数不相关的常量，可以最大化公式(12)来优化学习特征矩阵 $\boldsymbol{U}$、$\boldsymbol{V}$ 和 $\boldsymbol{C}$，在实际计算时一般选取公式中的主要分量进行计算，由于常量对优化无影响，为了方便计算，去掉无关常量并乘以 $\sigma^2$，最大化上述对数后验概率可以得到需要最小化的目标函数(13)：

$$\begin{aligned}&F(\boldsymbol{U},\boldsymbol{V},\boldsymbol{C}\,|\,\boldsymbol{RR},\sigma^2,\sigma_U^2,\sigma_V^2,\sigma_C^2)=\\&\frac{1}{2}\sum_{u=1}^{M}\sum_{i=1}^{N}\sum_{c=1}^{S}I_{uic}(RR_{uic}-<\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{C}_c>)^2+\\&\frac{\lambda_U}{2}\sum_{u=1}^{M}\boldsymbol{U}_u^T\boldsymbol{U}_u+\frac{\lambda_V}{2}\sum_{i=1}^{N}\boldsymbol{V}_i^T\boldsymbol{V}_i+\frac{\lambda_C}{2}\sum_{c=1}^{S}\boldsymbol{C}_c^T\boldsymbol{C}_c\end{aligned} \tag{13}$$

式中：$\lambda_U$ ，$\lambda_V$ 和 $\lambda_C$ 为正则化参数，分别为 $\lambda_U=\sigma^2/\sigma_U^2$ ，$\lambda_V=\sigma^2/\sigma_V^2$ ，$\lambda_C=\sigma^2/\sigma_C^2$ ，由于目标函数 $F$ 对 $\boldsymbol{U}_u$ 求导时不含 $\boldsymbol{U}$ 的项为零，对求导没有影响，因此为了方便后面的优化计算，使用(14)~(16) 3 种不同的表示方式：

$$\boldsymbol{F}_u=\frac{1}{2}\sum_{u=1}^{M}\sum_{i=1}^{N}\sum_{c=1}^{S}I_{uic}(RR_{uic}-<\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{C}_c>)^2+\frac{\lambda_U}{2}\sum_{u=1}^{M}\boldsymbol{U}_u^T\boldsymbol{U}_u \tag{14}$$

$$\boldsymbol{F}_v=\frac{1}{2}\sum_{i=1}^{N}\sum_{u=1}^{M}\sum_{c=1}^{S}I_{uic}(RR_{iuc}-<\boldsymbol{V}_i,\boldsymbol{U}_u,\boldsymbol{C}_c>)^2+\frac{\lambda_V}{2}\sum_{i=1}^{N}\boldsymbol{V}_i^T\boldsymbol{V}_i \tag{15}$$

$$\boldsymbol{F}_c=\frac{1}{2}\sum_{c=1}^{S}\sum_{u=1}^{M}\sum_{i=1}^{N}I_{uic}(RR_{uic}-<\boldsymbol{C}_c,\boldsymbol{U}_u,\boldsymbol{V}_i>)^2+\frac{\lambda_C}{2}\sum_{c=1}^{S}\boldsymbol{C}_c^T\boldsymbol{C}_c \tag{16}$$

这种通过概率矩阵分解的方法直接优化项目的排名倒数，不仅可以缓解数据稀疏的问题，而且能够有效提取用户的隐含特征，实现比较好的推荐效果。

### 2.4 模型优化

交叉最小二乘法(Alternative Least Squares，ALS)通过交替地固定其中一个潜在特征矩阵 $\boldsymbol{U}$ 优化另一个潜在特征矩阵 $\boldsymbol{V}$ 以保证该算法最终收敛[12]；而且它非常适合并行化，因此在大规模数据的运算中具有优势。

本文选择交叉最小二乘法优化目标函数，分别固定 $\boldsymbol{U}$、$\boldsymbol{V}$ 和 $\boldsymbol{C}$，通过求解 $\frac{\partial \boldsymbol{F}_u}{\partial \boldsymbol{U}_u}=0$ ，$\frac{\partial \boldsymbol{F}_v}{\partial \boldsymbol{V}_i}=0$ 和 $\frac{\partial \boldsymbol{F}_c}{\partial \boldsymbol{C}_c}=0$，可以求得如下公式：

$$\boldsymbol{U}_u=\boldsymbol{RR}_u(\boldsymbol{V}\odot\boldsymbol{C})(\boldsymbol{V}^T\boldsymbol{V}*\boldsymbol{C}^T\boldsymbol{C}+\lambda\boldsymbol{I})^{-1} \tag{17}$$

$$\boldsymbol{V}_i=\boldsymbol{RR}_i(\boldsymbol{U}\odot\boldsymbol{C})(\boldsymbol{U}^T\boldsymbol{U}*\boldsymbol{C}^T\boldsymbol{C}+\lambda\boldsymbol{I})^{-1} \tag{18}$$

$$\boldsymbol{C}_c=\boldsymbol{RR}_c(\boldsymbol{U}\odot\boldsymbol{V})(\boldsymbol{U}^T\boldsymbol{U}*\boldsymbol{V}^T\boldsymbol{V}+\lambda\boldsymbol{I})^{-1} \tag{19}$$

式中：$\boldsymbol{U}_u$ 表示用户 $u$ 的特征向量；$\boldsymbol{V}_i$ 表示项目 $i$ 的特征向量；$\boldsymbol{RR}_u$ 为 $1\times NS$ 的向量，表示用户 $u$ 在不同情境下播放过的音乐的排名倒数；$\boldsymbol{RR}_i$ 为 $1\times MS$ 的向量，表示音乐 $i$ 在不同情境下所有用户播放的排名倒数；$\boldsymbol{RR}_c$ 为 $1\times MN$ 的向量，表示在情境 $c$ 下所有用户播放过的音乐排名倒数。使用交叉最小二乘法优化 TFPMF 模型的步骤如下所示：

算法：ALS 优化的 TFPMF 算法

输入：训练数据集.

输出：特征矩阵 $\boldsymbol{U}$, $\boldsymbol{V}$ 和 $\boldsymbol{C}$。

① 计算排名倒数矩阵；
② 初始化特征矩阵；
③ for each step containing s
④ for u = 1,2,…,M do
⑤ calculate $\boldsymbol{U}_u$ by formula 17
⑥ end for
⑦ for i = 1,2,…,N do
⑧ calculate $\boldsymbol{V}_i$ by formula 18
end for
⑨ for c = 1,2,…,C do
⑩ calculate $\boldsymbol{C}_c$ by formula 19
⑪ end for
⑫ end for

## 3 实验结果

本文采用的实验平台为 PC(Intel(R)，CPU i7-4510，RAM(4GB)，Windows10 操作系统，开发工具使用 PyCharm Edu，算法使用 Python 语言编写。

本节首先介绍实验数据集、实验设置，接着说明评价标准及与其他算法的对比结果，最后对 TFPMF 算法与其他对比算法的实验结果进行分析。

### 3.1 度量标准

在推荐系统中，TOP-N 推荐更加适用于实际应用场景，尤其是音乐推荐领域，TOP-N 推荐更加符合用户的习惯。在 TOP-N 推荐系统中，排名越靠前的项目对用户越重要，这与用户的浏览行为相一致，因此推荐项目的排名对推荐系统至关重要。

本文采用准确率(Precision Rate)、召回率(Recall)和标准化折算累加值(Normalized Discounted Cumulative Gain，NDCG)作为评价指

标。其中准确率表示推荐计算结果的准确度，Precision@$N$则表示为用户生成的TOP-N推荐列表中符合用户需求的项目数与 $N$ 的比值，其定义为：

$$\text{Precision}@N=\frac{1}{\sum_{u\in U}N_u}\sum_{u\in U}|N_u\cap T(u)| \quad (20)$$

式中：$T(u)$表示测试集中用户 $u$ 的列表；$N_u$ 表示对用户 $u$ 生成的 TOP-N 推荐列表。

召回率是指推荐的正确项目和测试集中实际项目的比率，衡量的是查全率，Recall@$N$ 定义为公式(21)。

$$\text{Recall}@N=\frac{1}{\sum_{u\in U}|T(u)|}\sum_{u\in U}|N_u\cap T(u)| \quad (21)$$

式中：$T(u)$表示测试集中用户 $u$ 的项目列表；$|T(u)|$表示测试集中用户 $u$ 的项目列表中的项目个数；$\sum_{u\in U}|T(u)|$ 表示测试集中所有用户项目列表中的项目总和。

NDCG 度量一个排序列表的评价指标，推荐效果越好 NDCG 值越大，NDCG@N 定义为：

$$NDCG@N=\frac{DCG}{IDCG}=\frac{1}{|U|}\sum_{u\in U}\frac{1}{IDCG}\sum_{i=1}^{N}\frac{2^{r_{ui}}-1}{\log_2(i+1)} \quad (22)$$

当推荐的第 $i$ 个物品属于测试集 $T(u)$中物品时，$r_{ui}$ 为 1，否则为 0，上式中 IDCG 是完美匹配时的 DCG 值。

## 3.2 数据集

本实验建立在 last.fm 数据集的基础上，last.fm 音乐网站有来自世界各地的活跃用户群体，它提供了 API 供研究者使用。本文通过编写程序采集所选取用户近几年的播放日志记录，采集的数据形式为<user, artist, song, timestamp>；通过对数据进行清洗和处理，剔除不符合训练集和测试集划分要求的数据，筛选出有效数据用于实验。

本文主要利用 context_day 和 context_week 两维情境信息进行推荐，预测用户在不同的情境下偏好的音乐，并生成 TOP-N 推荐列表。本文采用 < userId,itemId,Context > 的方式划分情境数据集，其中 Context = ( (morrning, workday), (morrning, weekend),…,(evening,weekend))，含有 6 个元素。

为了验证本文所提方法的有效性和可行性，文章分别选取 306 个用户和 844 个用户来验证算法，数据集信息如表 1 所示。

表 1 数据集信息
Tab. 1 Information of datasets

| 用户数 | 音乐数 | 稀疏度/% | 隐式反馈数 | |
|---|---|---|---|---|
| | | | 训练集 | 测试集 |
| 306 | 81 299 | 99.342 | 1 654 667 | 87 499 |
| 844 | 185 634 | 99.678 | 5 451 058 | 279 470 |

采用的数据越稀疏，推荐的难度就会越大，因此解决数据的稀疏问题非常重要，使用合适的算法来缓解数据稀疏问题就非常具有实用意义。从表 1 可以看出，306 users 和 844 users 数据集的稀疏度都在 99.3%以上，数据非常稀疏。

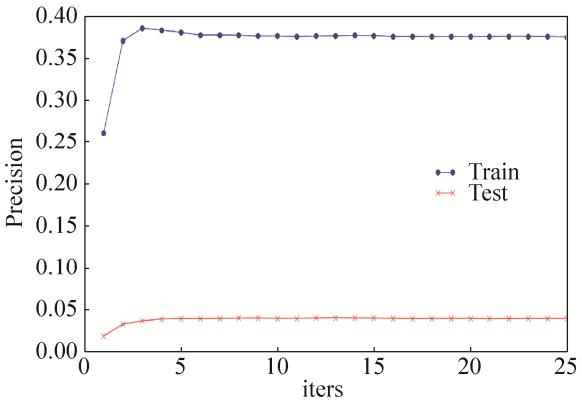
结合音乐推荐的实际应用，本文按时间来划分训练集和测试集，用于预测将来一段时间用户在特定情境下的播放倾向，实现个性化推荐；且用户在训练集中播放过的音乐不包含在测试集中，即只向用户推荐在该情境下没有播放过的音乐。

## 3.3 实验结果

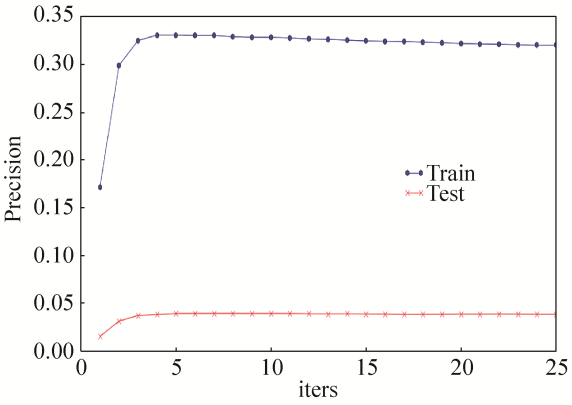
分别使用 306 users 和 844 users 数据集输入模型进行迭代计算，当训练集的准确率趋于稳定时模型取得最优解，停止迭代计算并输出推荐计算所需的用户特征矩阵 $\boldsymbol{U}$，项目特征矩阵 $\boldsymbol{V}$ 和情境特征矩阵 $\boldsymbol{C}$。通过计算结果可知，当迭代次数超过 10 次之后，训练集的准确率趋于稳定。

为了说明模型的可靠性及稳定性，本文选取前 25 次迭代计算的特征矩阵进行推荐计算，分别计算出推荐结果在训练集和测试集的准确率，并画出折线图如图 4 所示。

从图 4 可以看出，在 2 个数据集上，随着迭代步数的增加，训练集的准确率在达到峰值之后略有下降，并趋于稳定，测试集的准确率逐渐升高并最终趋于稳定，模型取得最优解，不会出现过拟合问题。



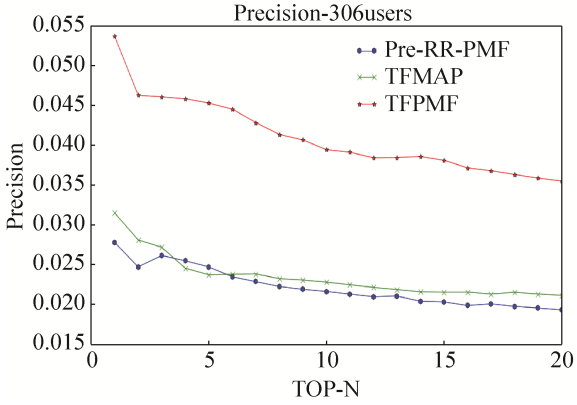

(a) 306users



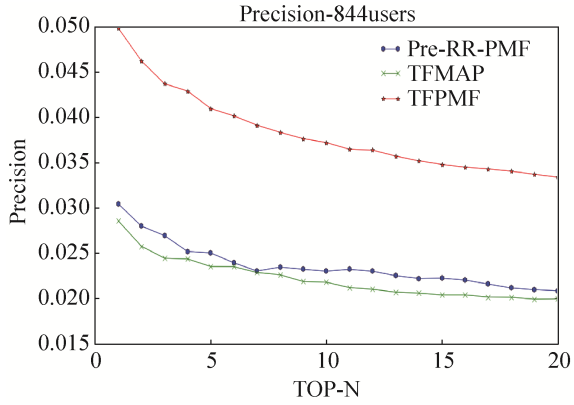

(b) 844users

图 4　不同迭代步数的准确率比较

Fig. 4　Precision comparison of different iteration steps




(a) 306users




(b) 844users

图 5　不同算法的准确率比较

Fig. 5　Precision comparison of different algorithms

本文选取基于情境感知预过滤的 RR-PMF 算法[10]和 TFMAP[11]的推荐算法作为对比算法，各种算法的隐含特征数均选为 30，通过实验仿真计算各种算法推荐列表。

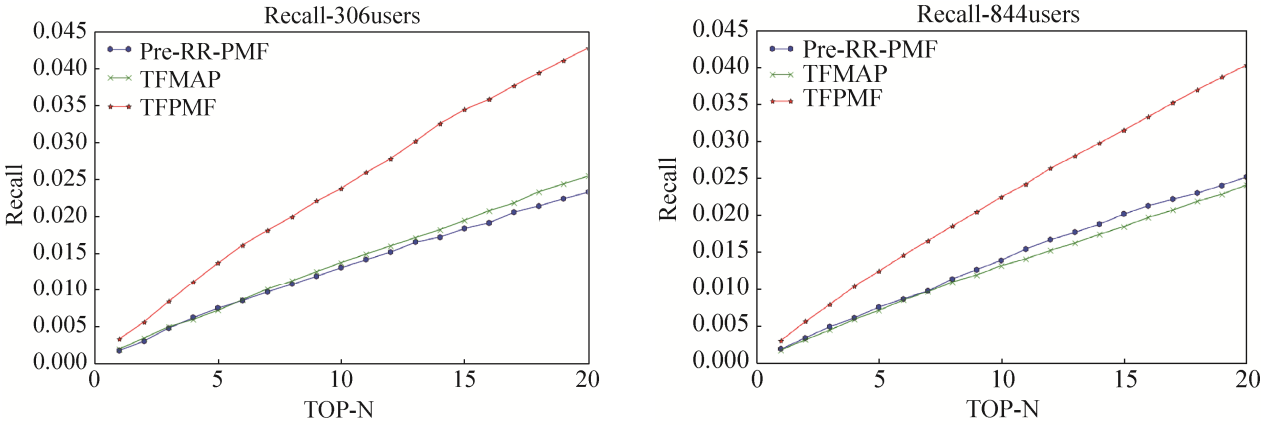
根据推荐计算结果与测试集进行比较计算出各种算法的 Precision、Recall 和 NDCG 指标，画出不同 TOP-N 推荐下的 Precision@N 折线图、Recall@N 折线图和 NDCG@N 折线图分别如图 5~7 所示。在图 5~7 中，N 的取值范围均为 1 到 20，图 5~7 中的(a)和(b)分别表示在 306 个用户和 844 个用户数据集上的仿真结果。

由图 5 可以看出，相比较其他的算法，TFPMF 在 2 个数据集上均表现出优秀的推荐效果，尤其在 TOP-1 和 TOP-2 推荐中命中率较高，随着推荐列表的增长，准确率逐渐趋于平稳，说明 TFPMF 推荐算法更加适用于 TOP-N 推荐。

从图 6 可以看出，推荐算法的召回率成斜线增长趋势，且 TFPMF 算法的增长斜率明显大于对比算法，说明 TFPMF 算法的查全率较高，推荐项目能够以更高的比率覆盖用户播放列表，更加符合用户兴趣，在推荐效果上表现更加突出。

通过图 7 的 NDCG 指标曲线图可以看出，NDCG 指标随着推荐列表的增长达到稳定值，TFPMF 算法在不同推荐列表长度上均具有明显优势，说明该算法在排名推荐中能够更好地挖掘用户的偏好，推荐出符合用户兴趣的音乐列表。

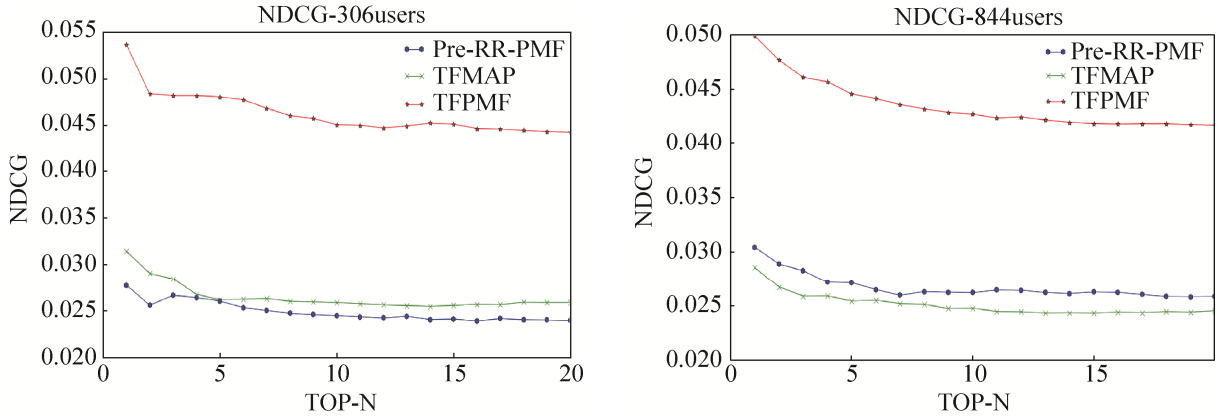
分别选取各种算法在 TOP-5、TOP-10 和 TOP-15 下的推荐结果，作 Precision 柱状图、Recall 柱状图和 NDCG 柱状图分别如图 8~10 所示。图 8~10 中的(a)和(b)分别表示在 306 个用户和 844 个用户数据集上的仿真结果。




(a) 306users (b) 844users

图 6 不同算法的召回率比较

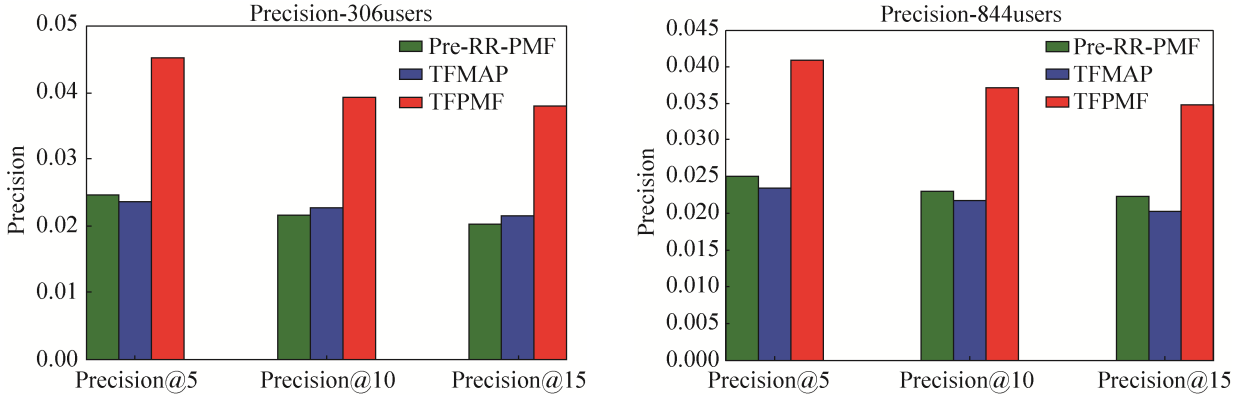
Fig. 6 Recall comparison of different algorithms




(a) 306users (b) 844users

图 7 不同算法的 NDCG 比较

Fig. 7 NDCG comparison of different algorithms



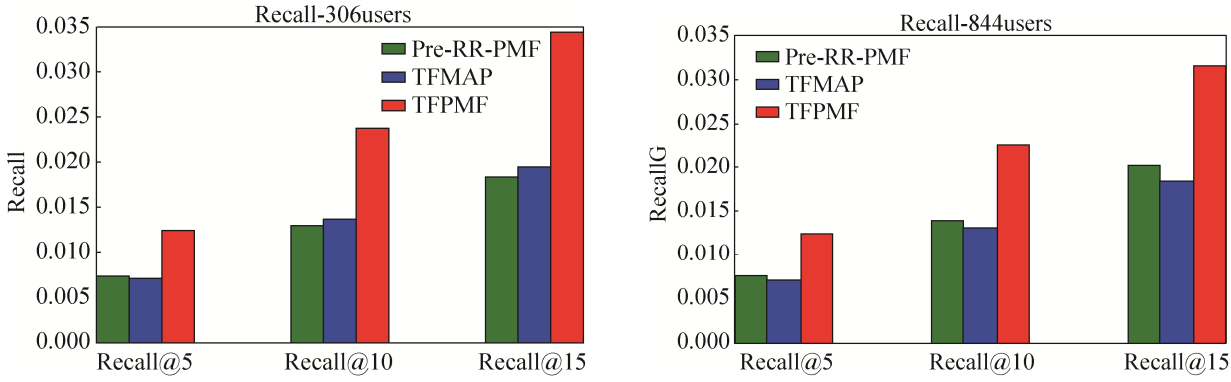

(a) 306users (b) 844users

图 8 准确率在 TOP-5、10、15 上的比较

Fig. 8 Precision comparison in TOP-5,10 and 15

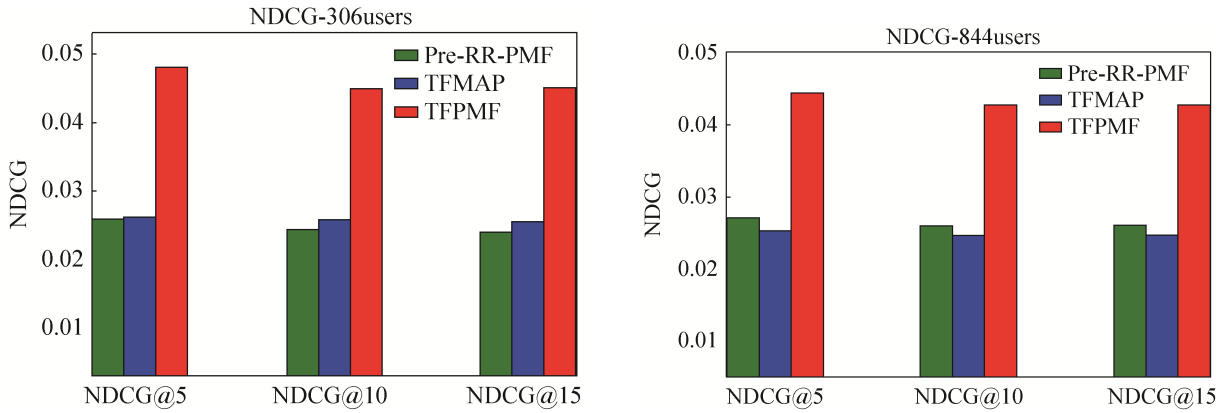


https://dc-china-simulation.researchcommons.org/journal/vol31/iss7/17
DOI: 10.16182/j.issn1004731x.joss.17-0256




(a) 306users (b) 844users

图 9 召回率在 TOP-5、10、15 上的比较
Fig. 9 Recall comparison in TOP-5, 10 and 15




(a) 306users (b) 844users

图 10 NDCG 在 TOP-5、10、15 上的比较
Fig. 10 NDCG comparison in TOP-5, 10 and 15

由图 8~10 可以明显看出，在 2 个数据集上 TOP-5，TOP-10 和 TOP-15 的推荐结果中，TFPMF 算法与其他算法相比在 Precision，Recall 和 NDCG 指标上均有明显提升，且在 Recall 指标上增长较快，在 NDCG 指标上有着良好的稳定性，因此 TFPMF 算法具有很大的应用价值。

本文以 TOP-10 仿真结果为例，分别对不同算法在 Precsion@10，Recall@10 和 NDCG@10 上的推荐效果进行量化比较，具体结果如表 2 所示，表中的提升效果表示 TFPMF 算法与对应算法在 Precsion@10，Recall@10 和 NDCG@10 上提升的百分比，其值越大说明效果越好。

表 2 算法效果对比
Tab. 2 Comparison of results of different algorithms

| 用户数 | 算法 | Precision@10 | 提升率/% | Recall@10 | 提升率/% | NDCG@10 | 提升率/% |
|---|---|---|---|---|---|---|---|
| 306 | Pre-RR-PMF | 0.021 58 | 82.53 | 0.013 01 | 82.55 | 0.024 46 | 84.10 |
| | TFMAP | 0.022 75 | 73.14 | 0.013 72 | 73.10 | 0.025 86 | 74.13 |
| | TFPMF | 0.039 39 | --- | 0.023 75 | --- | 0.045 03 | --- |
| 844 | Pre-RR-PMF | 0.022 75 | 63.43 | 0.013 89 | 61.84 | 0.026 17 | 63.13 |
| | TFMAP | 0.021 76 | 70.86 | 0.013 16 | 70.82 | 0.024 74 | 72.55 |
| | TFPMF | 0.037 18 | --- | 0.022 48 | --- | 0.042 69 | --- |

对比表中数据，从 844 users 数据集实验结果可得，TFPMF 算法在 Precision@10 分别比基于情境感知预过滤的 RR-PMF 算法和 TFMAP 算法提高了 63.43%和 70.86%，在准确性方面具有明显优势；在 Recall@10 的提升效果方面和准确率相似，具有较大的提高；在 NDCG@10 评价指标上分别提升了 63.13%和 72.55%，说明 TFPMF 算法在 TOP-N 推荐系统中表现出了良好效果。总之，实验结果表明：在准确性和 NDCG 评价指标上，TFPMF 算法比其他算法均有明显提高。

此外，使用交叉最小二乘(ALS)优化算法能够实现并行化计算，在实际应用中可以部署到分布式系统中进行离线计算，从而可以提高计算效率，适用于工业中大规模数据的计算，因此本文提出的 TFPMF 算法具有较强的实用性和可移植性。

### 3.4 时间复杂度

算法的时间花费在实际应用中具有重要意义，较少的时间花费能够节省更多的资源。TFPFM 算法只需考虑式(17)~(19)的时间消耗。$\boldsymbol{U}$ 为 $M\times K$ 的特征矩阵，$\boldsymbol{K}$ 表示选取特征个数，$\boldsymbol{U}^T\boldsymbol{U}$ 的时间复杂度为 $O(K^2M)$，$\boldsymbol{V}^T\boldsymbol{V}$ 的时间复杂度为 $O(K^2N)$，$\boldsymbol{C}^T\boldsymbol{C}$ 的时间复杂度为 $O(K^2S)$。假设 $\boldsymbol{V}^T\boldsymbol{V}*\boldsymbol{C}^T\boldsymbol{C}+\lambda\boldsymbol{I}$ 求逆的时间复杂度为 $O(K^3)$，$\boldsymbol{V}^T\boldsymbol{V}*\boldsymbol{C}^T\boldsymbol{C}$ 提前计算，由于 $\boldsymbol{RR}_u$ 中只包含 $n_u$ 个非零项，式(17)的时间复杂度可写为 $O(K^2n_R+Kn_R+K^3n_u)$，其中 $n_R=\sum_u n_u$，即有效交互数据的个数。由于矩阵 $\boldsymbol{RR}$ 高度稀疏，故有 $n_R<<MNS$。同理，由于 $\boldsymbol{RR}_i$ 和 $\boldsymbol{RR}_c$ 中只包含 $n_i$ 和 $n_c$ 个非零项，式(18)和式(19)的时间复杂度分别为 $O(K^2n_R+Kn_R+K^3n_i)$ 和 $O(K^2n_R+Kn_R+K^3n_c)$。因此在用户个数 $n_u$、推荐对象的个数 $n_i$ 和情境信息的维数 $n_c$ 确定的情况下，该算法的复杂度与 $n_R$ 成正比，本文提出的算法具有较强的可拓展性。

本文分别在 306 users 和 844 users 数据集上对比单步迭代的时间花费，不同算法迭代运行时间如表 3 所示。

表 3 算法单步迭代时间

Tab. 3 Iteration time of single step in different algorithms

| 算法 | | Pre-RR-PMF | TFMAP | TFPMF |
|---|---|---|---|---|
| 单步迭代时间/s | 306 users | 90±2 | 43 110±1 000 | 27±5 |
| | 844 users | 201±10 | 201 555±2 000 | 70±15 |

从表 3 可以看出，TFPMF 算法的单步迭代时间相对较小，如果使用并行化计算，时间花费还会大大降低，在大规模数据的处理中具有明显优势。

实验结果表明，本文提出的算法能够有效缓解数据稀疏问题，对推荐结果有明显提升；由于使用交叉二乘法优化模型，可以部署到分布式系统中如 HADOOP 平台进行离线计算，通过多节点并行化计算从而极大地节省运算时间，因此在基于情境感知的大规模隐式反馈推荐系统中具有很大的优势，完全适用于大规模数据的处理。

## 4 结论

本文根据排名倒数(RR)平滑表示的理论知识，在前面基于排名倒数(RR)的概率矩阵分解模型(RR-PMF)的基础上，与张量分解相结合提出了 TFPMF 算法进行 TOP-N 推荐。该算法能够有效地提取不同情境下的用户特征和缓解数据稀疏问题，相比其他算法在 Precision，Recall 和 NDCG 评价指标上有明显提升；由于该算法统一了情境信息的表示方法，其数据处理方法适用于不同推荐背景的隐式反馈数据的处理，而且该算法在不同用户数量的数据集上均表现出良好效果，其时间复杂度较低，与交互数据的数据量成正比，并使用交叉最小二乘法进行优化，在大数据量的实际应用中可以进行并行化计算，进一步大大缩短算法计算时间，因此具有良好的可移植性和拓展性，在基于情境感知的大规模隐式反馈系统中有着较大的应用价值。

## 参考文献：

[1] Adomavicius G, Tuzhilin A. Context-Aware Recommender Systems[J]. Ai Magazine (S0738-4602), 2010, 16(3): 2175-2178.

[2] Baltrunas L, Ricci F. Context-based splitting of item ratings in collaborative filtering[C]// Proceedings of the

third ACM conference on Recommender systems. New York: ACM, 2009: 245-248.

[3] Karatzoglou A, Amatriain X, Baltrunas L, et al. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering[C]// Proceedings of the fourth ACM conference on Recommender systems. Barcelona: ACM, 2010: 79-86.

[4] C Tianqi, Z Weinan, L Qiuxia, et al. SVDFeature: a toolkit for feature-based collaborative filtering[J]. Journal of Machine Learning Research (S1532-4435), 2012, 13(1): 3619-3622.

[5] Hidasi B. Factorization models for context-aware recommendations[J]. Infocommun J (S0219-1377), 2014, VI(4): 27-34.

[6] Baltrunas L, Ludwig B, Ricci F. Matrix factorization techniques for context aware recommendation[C]// ACM Conference on Recommender Systems, Recsys 2011. USA: Chicago, 2011: 301-304.

[7] Baltrunas L, Kaminskas M, Ludwig B, et al. InCarMusic: Context-Aware Music Recommendations in a Car[J]. Lecture Notes in Business Information Processing (S1865-1356), 2011, 85: 89-100.

[8] Yue S, Karatzoglou A, Baltrunas L, et al. TFMAP: optimizing MAP for top-n context-aware recommendation[C]// Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. Geneva: ACM, 2012: 155-164.

[9] Nguyen T V, Karatzoglou A, Baltrunas L. Gaussian process factorization machines for context-aware recommendations[C]// Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. Portland: ACM, 2014: 63-72.

[10] 王猛, 叶西宁. 音乐个性化推荐算法 RR-UBPMF 的研究[J]. 华东理工大学学报, 2017, 43(1): 113-118.
Wang Meng, Ye Xining. RR-UBPMF, A personalized music recommendation algorithm[J]. Journal of East China University of Science and Technology(Natural Science Edition), 2017, 43(1): 113-118.

[11] Kolda T G, Bader B W. Tensor decompositions and applications[J]. SIAM Review (S0036-1445), 2009, 51(3): 455-500.

[12] Pilászy I, Zibriczky D, Tikk D. Fast als-based matrix factorization for explicit and implicit feedback datasets[C]// Proceedings of the fourth ACM conference on Recommender systems. Barcelona: ACM, 2010: 71-78.