

# 基于受约束偏置的概率矩阵分解算法

梅 忠<sup>1,2</sup>, 肖如良<sup>1,2</sup>, 张桂刚<sup>3</sup>

<sup>1</sup>(福建师范大学 软件学院, 福州 350117)

<sup>2</sup>(大数据分析与应用福建省高校工程研究中心, 福州 350117)

<sup>3</sup>(中国科学院自动化研究所, 北京 100190)

**摘 要:** 在概率矩阵分解(PMF)模型拟合之后, 评分较少用户的特征趋近于先验分布的平均值, 导致对其评分预测接近物品的平均评分. 受约束概率矩阵分解(CPMF)未考虑到不同评分系统的整体差异以及数据集内部用户与物品存在的固有属性. 针对以上问题, 提出将传统矩阵分解中的用户和物品偏置项以及全局平均分结合受约束概率矩阵分解来建立新的矩阵分解算法. 算法利用整体平均分衡量不同评分系统, 在采用偏置来表示用户以及物品之间相互独立的属性的同时, 引入约束使行为相近用户拥有相近的用户偏置, 从而提高预测精度. 在两个真实数据集上的实验结果表明, 该算法相对于 PMF 和 CPMF 算法预测精度得到了提高.

**关键词:** 推荐系统; 协同过滤; 概率矩阵分解; 约束; 偏置

## Probabilistic Matrix Factorization Based on Constrained Bias

MEI Zhong<sup>1,2</sup>, XIAO Ru-Liang<sup>1,2</sup>, ZHANG Gui-Gang<sup>3</sup>

<sup>1</sup>(Faculty of Software, Fujian Normal University, Fuzhou 350117, China)

<sup>2</sup>(Fujian Provincial University Engineering Research Center of Big Data Analysis and Application, Fuzhou 350117, China)

<sup>3</sup>(Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China)

**Abstract:** Users who rate few will have features that are close to the prior mean once the probabilistic matrix factorization(PMF) model has been fitted, which leads to the predictions close to the movie average ratings. Constrained probabilistic matrix factorization(CPMF) algorithm has not fully considered about the holistic diversity among different rating systems and the inherent attributes that different users and products hold within the datasets. To solve the problems above, the user and product bias and global average are combined with constrained probabilistic matrix factorization to build a new matrix factorization algorithm. The algorithm brings in the constraints to restrain the user bias among users of similar action while evaluating different rating systems with global average and representing the attributes of different users and products with biases to increase the prediction accuracy. The results of experiments on two real datasets indicate that the prediction accuracy of the algorithm has been improved compared to PMF and CPMF.

**Key words:** recommendation system; collaborative filtering; probabilistic matrix factorization; constrain; bias

协同过滤是应用最早、范围最广,也是效果最好的推荐方法,其中的矩阵分解(Matrix Factorization,MF)模型常用于评分预测.文献[1]提出了概率矩阵分解(Probabilistic MF, PMF),对矩阵分解模型进行了概率解释.即使是受约束概率矩阵(Constrained PMF, CPMF)也只是利用用户行为得到约束向量来影响用户特征向量,通过用户和物品特征向量内积进行评分预测,使其评

分预测较为单一,影响预测精度.

如何提高预测精度有多种方法,文献[2]提出将用户行为时序化之后建立用户之间非对等的影响关系,但冷启动问题严重;文献[3]通过不信任关系约束的信任关系的传播来解决冷启动问题进而提高精度,但算法复杂度高;文献[4]通过隐式标签信息来计算用户或者物品的相似度得到邻域关系,但获取足够的社会关

基金项目:河南省重点科技攻关项目(142102210225)

收稿时间:2015-08-26;收到修改稿时间:2015-10-26

通讯作者:肖如良, Email:xiaoruliang@163.com

系和标签信息在实际中都是比较困难的. 在传统的矩阵分解中加入偏置已在 Netflix Prize 中证实可行<sup>[5]</sup>.

受此启发, 本文将偏置以及系统平均评分融合到受约束概率矩阵分解(CPMF)中, 同时引入用户偏置约束, 提出受约束偏置的概率矩阵分解算法(Constrained Biased PMF, CBPMF), 可有效地提高推荐性能.

## 1 概率矩阵分解模型

### 1.1 PMF 模型<sup>[1]</sup>

假设有  $N$  个用户,  $M$  部电影,  $R=[R_{ij}]_{N \times M}$  为评分系统的评分矩阵,  $R_{ij}$  表示用户  $i$  对电影  $j$  的评分, 取值范围为 1 到 5.  $U \in R^{D \times N}$ 、 $V \in R^{D \times M}$  分别为用户  $i$  和电影  $j$  的特征向量. PMF 基本思想是假设整体评分服从均值为用户和物品特征向量内积的高斯分布, 那么对评分矩阵的条件分布可定义为公式(1):

$$p(R|U, V, \sigma_R^2) = \prod_{i=1}^N \prod_{j=1}^M \left( N(R_{ij} | U_i^T V_j, \sigma_R^2) \right)^{I_{ij}} \quad (1)$$

其中  $I_{ij}$  为示性函数, 若用户  $i$  评价过电影  $j$  则其为 1, 否则为 0,  $\sigma_R$  为整体评分方差,  $N(x|\mu, \sigma^2)$  表示高斯分布. 假设用户和物品的特征向量的元素是服从均值为 0, 方差分别为  $\sigma_U$ 、 $\sigma_V$  的高斯分布, 对他们作如下定义:

$$\begin{aligned} p(U|\sigma_U^2) &= \prod_{i=1}^N N(U_i | 0, \sigma_U^2 I) \\ p(V|\sigma_V^2) &= \prod_{j=1}^M N(V_j | 0, \sigma_V^2 I) \end{aligned} \quad (2)$$

为方便计算, 对用户和电影的后验分布取自然对数得到式(3):

$$\begin{aligned} \ln p(U, V | R, \sigma_R^2, \sigma_U^2, \sigma_V^2) &= -\frac{1}{2\sigma_R^2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} (R_{ij} - U_i^T V_j)^2 - \frac{1}{2\sigma_U^2} \sum_{i=1}^N U_i^T U_i - \frac{1}{2\sigma_V^2} \sum_{j=1}^M V_j^T V_j \\ &\quad - \frac{1}{2} \left( \sum_{i=1}^N \sum_{j=1}^M I_{ij} \right) \ln \sigma_R^2 + ND \ln \sigma_U^2 + MD \ln \sigma_V^2 + C \end{aligned} \quad (3)$$

若保持整体评分  $R$ 、评分先验方差  $\sigma_R$ 、用户及物品先验方差  $\sigma_U$ 、 $\sigma_V$  固定, 最大化(3)式等同于最小化式(4)的目标函数:

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} (R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2} \sum_{i=1}^N U_i^T U_i + \frac{\lambda_V}{2} \sum_{j=1}^M V_j^T V_j \quad (4)$$

对于式(4), 我们可用随机梯度下降法进行迭代训练来建立模型.

PMF 模型中, 在算法拟合之后, 评分较少用户的特征接近先验分布的平均值, 导致对于他们的评分预

测接近电影的平均评分, 影响推荐精度. 一种改进方法是通过用户评分行为来约束用户特征向量.

### 1.2 CPMF 模型

文献[1]中提出利用约束矩阵  $W \in R^{M \times D}$  来约束用户特征, 即赋予每部电影除特征向量之外的一个约束向量, 让每个用户评分过的所有电影约束向量的均值来影响其特征向量, 因此评分行为相近的用户会有相似的特征向量. 定义新的用户特征如式(5):

$$Y_i = U_i + \frac{\sum_{k=1}^M I_{ik} W_k}{\sum_{k=1}^M I_{ik}} \quad (5)$$

因此对于整个评分矩阵新的条件分布定义如式(6):

$$p(R|U, V, W, \sigma_R^2) = \prod_{i=1}^N \prod_{j=1}^M \left( N(R_{ij} | (U_i + \frac{\sum_{k=1}^M I_{ik} W_k}{\sum_{k=1}^M I_{ik}})^T V_j, \sigma_R^2) \right)^{I_{ij}} \quad (6)$$

其中  $W$  定义为服从均值为 0, 方差为  $\sigma_W$  的高斯先验分布:

$$p(W|\sigma_W^2) = \prod_{k=1}^M N(W_k | 0, \sigma_W^2 I) \quad (7)$$

同 PMF 一样, 对于  $U$ 、 $V$  和  $W$  的后验分布取对数之后的最大化等同于最小化式(8):

$$\begin{aligned} E &= \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} \left( R_{ij} - (U_i + \frac{\sum_{k=1}^M I_{ik} W_k}{\sum_{k=1}^M I_{ik}})^T V_j \right)^2 \\ &\quad + \frac{\lambda_U}{2} \sum_{i=1}^N U_i^T U_i + \frac{\lambda_V}{2} \sum_{j=1}^M V_j^T V_j + \frac{\lambda_W}{2} \sum_{k=1}^M W_k^T W_k \end{aligned} \quad (8)$$

同样, 建立 CPMF 模型的过程也可以对  $U$ 、 $V$  和  $W$  分别使用随机梯度下降法来进行训练.

CPMF 相对于 PMF 模型一定程度上提高了推荐精度, 但其仍然忽略了不同评分系统以及用户及物品存在的相互独立的属性, 这些属性会很大程度影响评分的预测, 针对这一问题, 本文提出了一种新的算法: CBPMF 算法.

## 2 CBPMF模型: 融合受约束偏置与全局平均的方法

有的用户倾向于只对自己非常满意的物品给出评分, 因此他的评分均值相对于不管对什么物品都会给出评分的用户的肯定要高. 有些物品本身质量较好, 因此用户给出的评分相对其他物品整体来说也较高. 同时,

不同的评分系统也肯定有不同的用户群体和不同的物品种类. 在 Netflix Prize 比赛中已证实将系统全局评分均值添加到传统矩阵分解中可有效提高推荐精度<sup>[5]</sup>, 因此本文将偏置与全局平均等因素加入 CPMF 模型中. 不仅如此, 为了避免用户偏置在训练后也出现像用户特征一样接近先验平均的现象, 影响推荐精度, 并且让评分行为相似的用户也拥有相似的偏置, 我们引入  $Z \in R^M$  来约束用户偏置. 假设用户偏置、物品偏置以及整体评分相互独立, 用户和物品偏置以及偏置约束数组服从均值为 0, 方差分别为  $\sigma_p$ 、 $\sigma_Q$ 、 $\sigma_Z$  的高斯分布.

$$\begin{aligned} p(P|\sigma_p^2) &= \prod_{n=1}^N N(P_n | 0, \sigma_p^2) \\ p(Q|\sigma_Q^2) &= \prod_{m=1}^M N(Q_m | 0, \sigma_Q^2) \\ p(Z|\sigma_Z^2) &= \prod_{l=1}^M N(Z_l | 0, \sigma_Z^2) \end{aligned} \quad (9)$$

整个评分矩阵的条件分布重新定义为式(10):

$$p(R|U, V, Z, P, Q, W, \sigma_R^2) = \prod_{i=1}^N \prod_{j=1}^M \left( N(R_{ij} | \mu + (P_i + \frac{\sum_{l=1}^M I_{il} Z_l}{\sum_{l=1}^M I_{il}}) + Q_j + (U_i + \frac{\sum_{k=1}^M I_{ik} W_k}{\sum_{k=1}^M I_{ik}})^T V_j, \sigma_R^2) \right)^{I_{ij}} \quad (10)$$

根据贝叶斯推断, 对用户、物品、偏置和约束矩阵的后验分布取自然对数得到式(11):

$$\begin{aligned} \ln p(U, V, Z, P, Q, W | R, \sigma_R^2, \sigma_U^2, \sigma_V^2, \sigma_Z^2, \sigma_P^2, \sigma_Q^2, \sigma_W^2) \\ = -\frac{1}{2\sigma_R^2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} \left( R_{ij} - \mu - (P_i + \frac{\sum_{l=1}^M I_{il} Z_l}{\sum_{l=1}^M I_{il}}) - Q_j - (U_i + \frac{\sum_{k=1}^M I_{ik} W_k}{\sum_{k=1}^M I_{ik}})^T V_j \right)^2 \\ -\frac{1}{2\sigma_U^2} \sum_{i=1}^N U_i^T U_i - \frac{1}{2\sigma_V^2} \sum_{j=1}^M V_j^T V_j - \frac{1}{2\sigma_P^2} \sum_{n=1}^N P_n^T P_n - \frac{1}{2\sigma_Q^2} \sum_{m=1}^M Q_m^T Q_m \\ -\frac{1}{2\sigma_Z^2} \sum_{l=1}^M Z_l^T Z_l - \frac{1}{2\sigma_W^2} \sum_{k=1}^M W_k^T W_k - C \end{aligned} \quad (11)$$

对于求(11)式的最大值同样可以转化成为对(12)式作为目标函数求最小值.

$$\begin{aligned} E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} \left( R_{ij} - \mu - (P_i + \frac{\sum_{l=1}^M I_{il} Z_l}{\sum_{l=1}^M I_{il}}) - Q_j - (U_i + \frac{\sum_{k=1}^M I_{ik} W_k}{\sum_{k=1}^M I_{ik}})^T V_j \right)^2 \\ + \frac{\lambda_U}{2} \sum_{i=1}^N U_i^T U_i + \frac{\lambda_V}{2} \sum_{j=1}^M V_j^T V_j + \frac{\lambda_P}{2} \sum_{n=1}^N P_n^T P_n \\ + \frac{\lambda_Q}{2} \sum_{m=1}^M Q_m^T Q_m + \frac{\lambda_Z}{2} \sum_{l=1}^M Z_l^T Z_l + \frac{\lambda_W}{2} \sum_{k=1}^M W_k^T W_k \end{aligned} \quad (12)$$

我们仍用随机梯度下降法对式(12)进行训练来建立模型. 为后面表述梯度公式方便, 对公式在表示上做简化处理:

$$X_i = P_i + \frac{\sum_{l=1}^M I_{il} Z_l}{\sum_{l=1}^M I_{il}}, \quad Y_i = U_i + \frac{\sum_{k=1}^M I_{ik} W_k}{\sum_{k=1}^M I_{ik}}$$

对式(12)分别对  $U$ 、 $V$ 、 $P$ 、 $Q$ 、 $W$ 、 $Z$  求梯度可以得到下面的梯度公式(13)~(18):

$$\frac{\partial E}{\partial U} = -\sum_{j=1}^M I_{ij} (R_{ij} - \mu - X_i - Q_j - Y_i^T V_j) V_j + \lambda_U U_i \quad (13)$$

$$\frac{\partial E}{\partial V} = -\sum_{i=1}^N I_{ij} (R_{ij} - \mu - X_i - Q_j - Y_i^T V_j) Y_i + \lambda_V V_j \quad (14)$$

$$\frac{\partial E}{\partial P} = -\sum_{j=1}^M I_{ij} (R_{ij} - \mu - X_i - Q_j - Y_i^T V_j) + \lambda_P P_i \quad (15)$$

$$\frac{\partial E}{\partial Q} = -\sum_{i=1}^N I_{ij} (R_{ij} - \mu - X_i - Q_j - Y_i^T V_j) + \lambda_Q Q_j \quad (16)$$

$$\frac{\partial E}{\partial Z} = -\sum_{i=1}^N I_{ij} (R_{ij} - \mu - X_i - Q_j - Y_i^T V_j) \left/ \sum_{l=1}^M I_{il} \right. + \lambda_Z Z_l \quad (17)$$

$$\frac{\partial E}{\partial W} = -\sum_{i=1}^N I_{ij} (R_{ij} - \mu - X_i - Q_j - Y_i^T V_j) \left/ \sum_{k=1}^M I_{ik} \right. + \lambda_W W_k \quad (18)$$

CBPMF 算法的计算时间开销主要来自于目标函数  $E$  和梯度下降公式的计算, 从式(12)可以看出, 目标函数  $E$  的时间复杂度为  $O(NM^3)$ , (13)~(18)式的时间复杂度分别为  $O(M^2)$ 、 $O(NM^2)$ 、 $O(M^2)$ 、 $O(MN)$ 、 $O(NM^2)$ 、 $O(NM^2)$ , 因此每次迭代计算, 算法的时间复杂度为  $O(NM^3)$ .

梯度计算模型的图模型如图 1 所示.

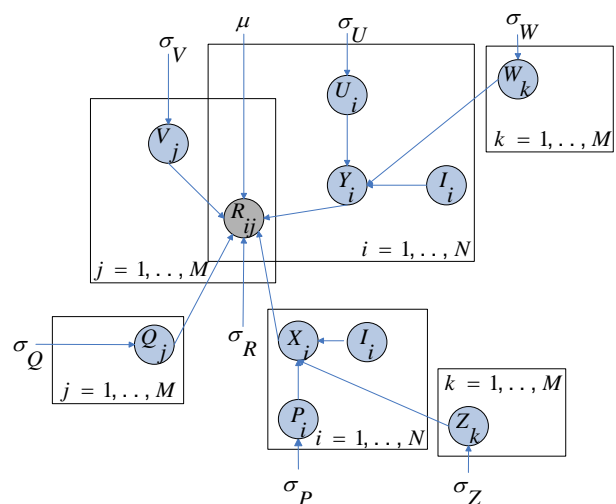


图 1 CBPMF 图模型

### 3 实验结果及分析

本文在两组数据集上给出 PMF、CPMF、CBPMF

三种算法的实验对比与分析。

### 3.1 数据集描述以及评价指标

我们选用两组数据集来做对比实验, 一组是 MovieLens 的 1M 数据集, 其包含 6040 个用户对 3952 部电影的 1000209 条评分数据. 另一组是从 Netflix Prize 数据集中随机抽取的, 其包含 6000 个用户对 4000 部电影的 987015 条评分, 称为 toyNetflix 数据集. 实验中, 我们对两组数据集都进行 7:1 的比例进行分割, 其中的 7/8 作为训练集, 剩下的 1/8 作为测试集.

我们采用 RMSE 和 MAE 作为实验的评价指标, 两个指标都是值越低, 代表预测精确度越高. 假设测试集中有  $n$  个评分, 其中已存在的评分  $r_i$  的预测评分为  $\hat{r}_i$ , RMSE, MAE 的定义如下所示.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (r_i - \hat{r}_i)^2}{n}}, MAE = \frac{\sum_{i=1}^n |r_i - \hat{r}_i|}{n}$$

### 3.2 实验参数的调试

在两组数据集上的实验均取三种对比算法的最佳结果. CBPMF 中涉及的  $\lambda_U$ ,  $\lambda_V$ ,  $\lambda_W$ ,  $\alpha$ ,  $momentum$  这五个参数在提出 PMF 和 CPMF 算法的文章<sup>[1]</sup>中均有实验取值作为参考, 调整范围不大, 所占时间较少. 参数  $D$  为用户及电影特征维度, 三种算法均取值 10. 主要针对 CBPMF 算法中  $\lambda_p$ ,  $\lambda_Q$ ,  $\lambda_Z$  参数进行调整, 由于这三个参数与  $\lambda_U$ ,  $\lambda_V$ ,  $\lambda_W$  属同类型参数, 因此先确定调整范围为 0.001 至 0.005 之间, 调整过程以 0.001 递增进行. 为得到最佳参数组合, 可轮流将参数固定, 调试剩余参数, 最终综合各参数最佳值进行取舍, 得到最后组合取值.

目前情况下, 可考虑使用 EM 算法在每一次迭代过程中自动调整参数, 这样只需给出参数粗略初始值即可得到比手动调试更精确的参数取值, 且可得到更低的 RMSE 及 MAE 值.

### 3.3 实验过程及结果分析

本文选取 PMF, CPMF 和 CBPMF 三个算法的 RMSE 和 MAE 结果来进行对比分析. CBPMF 的参数设置如下:

(1) movieLens 1M 数据集:  $D=10$ 、 $\lambda_U=0.001$ 、 $\lambda_V=0.001$ 、 $\lambda_Z=0.002$ 、 $\lambda_W=0.005$ 、 $\lambda_p=0.005$ 、 $\lambda_Q=0.001$ 、 $\alpha=0.0005$ 、 $momentum=0.9$ . 最终实验结果如图 2.

图 2 上部分, PMF 和 CPMF 的 RMSE 最小值分别

为 0.9273、0.8898, CBPMF 算法的 RMSE 最小值为 0.8786, 相对于 PMF 和 CPMF 分别提高 5.25%、1.26%. 图 2 下部分 PMF 和 CPMF 的 MAE 最小值分别为 0.7257、0.6986, CBPMF 算法的 MAE 最小值为 0.6816, 相对与 PMF 和 CPMF 分别提高 6.08%、2.43%.

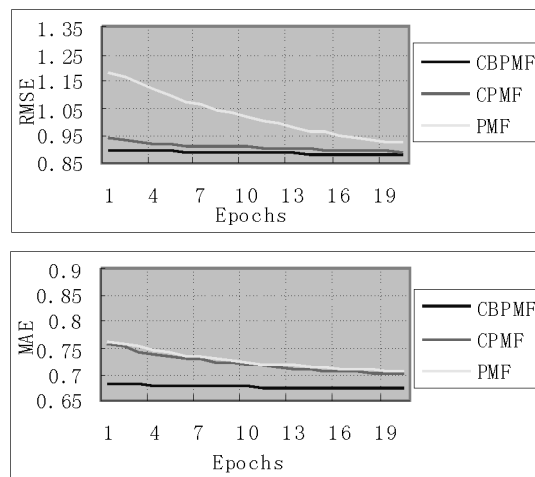


图 2 movieLens 1M 数据集上结果

(2) toyNetflix 16M 数据集:  $D=10$ 、 $\lambda_U=0.001$ 、 $\lambda_V=0.001$ 、 $\lambda_Z=0.005$ 、 $\lambda_W=0.005$ 、 $\lambda_p=0.005$ 、 $\lambda_Q=0.005$ 、 $\alpha=0.0006$ 、 $momentum=0.8$ . 最终实验结果如图 3.

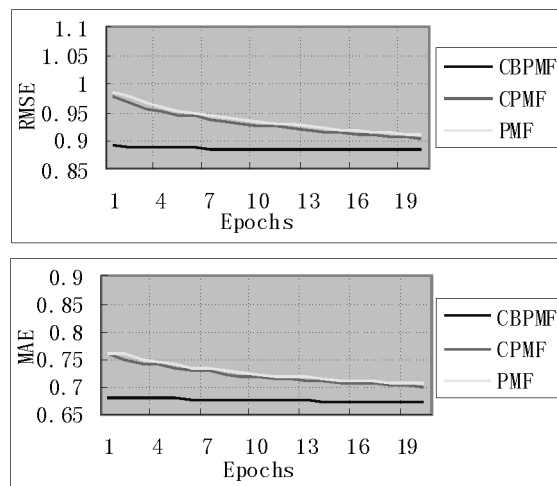


图 3 toyNetflix 16M 数据集上结果

图 3 上部分, PMF 和 CPMF 的 RMSE 最小值分别为 0.9107、0.9051. CBPMF 的 RMSE 最小值为 0.8844, 相对于 PMF 和 CPMF 分别提高了 2.65%、2.29%. 图 3 下部分, PMF 和 CPMF 的 MAE 最小值分别为 0.7065、

0.7015, CBPMF 的 MAE 最小值为 0.6756, 相对于 PMF 和 CPMF 分别提高 4.37%、3.69%。

通过在两组真实数据集上的实验, 与 PMF 和 CPMF 相比, CBPMF 的预测精度都有一定的提高。说明将全局平均、用户和物品偏置融合到概率矩阵分解中, 能提供更好的推荐质量。

#### 4 结语

本文提出了基于受约束用户偏置和物品偏置及全局评分均值的新的概率矩阵分解算法, 综合考虑不同评分系统以及用户和物品之间的固有差异。经过在两组数据集上的实验表明, 本文提出的新算法相比传统的矩阵分解算法在推荐精确度上有一定的提高, 但算法需要消耗部分时间对涉及到的参数进行手动调试。在未来的研究中, 如何让参数在迭代过程中自动适应从而进一步提高推荐精度是我们接下来的工作重点。

#### 参考文献

- 1 Mnih A, Salakhutdinov R. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*, 2007.
- 2 孙光福, 吴乐, 刘淇, 朱琛, 陈恩红. 基于时序行为的协同过滤推荐算法. *软件学报*, 2013, 24(11): 2721-2733.
- 3 印桂生, 张亚楠, 董宇欣, 等. 基于受限信任关系和概率分解矩阵的推荐. *电子学报*, 2014, (5): 904-911.
- 4 Wu L, Chen E, Liu Q, et al. Leveraging tagging for neighborhood-aware probabilistic matrix factorization. *Proc. of the 21st ACM International Conference on Information and Knowledge Management*. ACM. 2012. 1257-1264.
- 5 Koren Y. The BellKor solution to the Netflix grand prize. *Netflix Prize Documentation*, 2009.
- 6 Salakhutdinov R, Mnih A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. *Proc. of the Twenty Fifth International Conference Machine Learning*. Helsinki Finland. June 5-9, 2008. 880-887.
- 7 Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. *Computer*, 2009, 42(8): 30-37.
- 8 Ma H, Yang H, Lyu M. SoRec: Social recommendation using probabilistic matrix factorization. *Proc. of the 17th Acm Conference on Information and Knowledge Management*. California Usa. October 26-30, 2008. 931-940.
- 9 Shan H, Banerjee A. Generalized probabilistic matrix factorizations for collaborative filtering. *2010 IEEE 10th International Conference on Data Mining (ICDM)*. IEEE. 2010. 1025-1030.
- 10 Hernandezlobato J, Houlsby N, Ghahramani Z. Probabilistic Matrix factorization with non-random missing data. *Proc. of the 31st International Conference on Machine Learning*. 2014.
- 11 Lawrence N D, Urtasun R. Non-linear matrix factorization with Gaussian processes. *International Conference on Machine Learning*. 2009.