

文章编号:1007-130X(2018)10-1731-06

## 基于标签自适应选择的矩阵分解推荐算法<sup>\*</sup>

宋 威<sup>1,2</sup>, 李雪松<sup>1</sup>

(1. 北方工业大学计算机学院, 北京 100144; 2. 大规模流数据集成与分析技术北京市重点实验室, 北京 100144)

**摘 要:**将标签融入矩阵分解方法是当前推荐系统研究的热点。提出了一种基于标签自适应选择的矩阵分解推荐算法。首先,提出了标签-评分稀疏系数,较好地平衡了推荐过程中潜在特征与标签的使用问题。其次,利用标签的次数来计算标签向量,体现了标签的不同频率对不同物品的影响。最后,给出了算法的总体描述。实验结果表明,算法具有较高的推荐精度和较快的收敛速度。

**关键词:**推荐系统;矩阵分解;隐语义模型;标签自适应选择;标签-评分稀疏系数

**中图分类号:**TP181

**文献标志码:**A

**doi:**10.3969/j.issn.1007-130X.2018.10.003

## A matrix factorization recommendation algorithm based on adaptive tag selection

SONG Wei<sup>1,2</sup>, LI Xue-song<sup>1</sup>

(1. College of Computer Science and Technology, North China University of Technology, Beijing 100144;

2. Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data, Beijing 100144, China)

**Abstract:** Incorporating tags into matrix factorization is a hot topic in the field of recommender system. Based on adaptive tag selection, we propose a new matrix factorization recommendation algorithm. Firstly, we put forward a tag-rating sparsity factor, which balances the usage of latent factors and tags in recommendation. Secondly, tag vectors are computed by the number of tags, which reflects the influence of different frequencies of tags on different items. Finally, the overall description of the proposed algorithm is illustrated. Experimental results show that the proposed algorithm has high recommendation accuracy and high convergence speed.

**Key words:** recommendation system; matrix factorization; latent factor model; adaptive tag selection; tag-rating sparsity coefficient

### 1 引言

作为获取互联网有效信息的主要工具,推荐系统<sup>[1,2]</sup>是多学科交叉的研究热点,协同过滤<sup>[3]</sup>、基于内容的推荐<sup>[4]</sup>、混合推荐<sup>[5]</sup>是常见的几类推荐算法,这些算法在诸多领域<sup>[6,7]</sup>有着广泛的应用。

矩阵分解是将规模较大的复杂问题转换为多个规模较小的简单子问题的矩阵分析方法。基于

矩阵分解的推荐算法<sup>[8,9]</sup>正得到越来越多学者的关注。奇异值分解 SVD(Singular Value Decomposition)<sup>[10]</sup>是最早用于推荐的矩阵分解方法。此后,用隐语义模型 LFM(Latent Factor Model)<sup>[11]</sup>改进 SVD 并用于推荐,成为了这类方法中最为主流的一类。基于矩阵分解的推荐系统通常使用物品的类别来作为潜在特征,但类别对物品的描述不够精确,使得这类算法的推荐精度很难得到进一步的提升。

<sup>\*</sup> 收稿日期:2017-09-11;修回日期:2018-01-04

基金项目:北京市自然科学基金(4162022);北京市科技计划项目(D161100005216002);北京市优秀人才青年拔尖个人项目(2015000026833ZK04)

通信地址:100144 北京市石景山区晋元庄路 5 号北方工业大学计算机学院

Address: College of Computer Science and Technology, North China University of Technology, 5 Jinyuanzhuang Rd, Shijingshan District, Beijing 100144, P. R. China

近年来,标签的应用越来越广泛,标签不但可以描述物品的特征,而且还可以体现用户对物品的喜好,因此将标签引入推荐系统可以使得矩阵分解的潜在特征更加具体,从而改进推荐效果<sup>[12,13]</sup>。最近,方冰和牛晓婷使用标签来扩充隐语义模型,得到了较好的推荐效果<sup>[14]</sup>。

在提高推荐精度的同时,使用标签来扩充隐语义模型也增加了矩阵的规模,特别是在标签较为稀疏时会带来较高的计算开销。为解决这一问题,本文提出了一种基于标签自适应选择的矩阵分解推荐算法 RATMF (Recommendation based on Adaptive Tag of Matrix Factorization)。RATMF 算法在隐语义模型的基础上,融入了标签的自适应选择策略。一方面,提出了标签-评分稀疏系数,通过比较标签与评分的相对数量,来决定对某一物品预测时是否引入标签因子;另一方面,使用了标签的相对次数来组成物品的标签向量,体现了不同标签数量对不同物品的影响。实验结果表明,本文提出的算法具有较好的推荐效果。

## 2 推荐算法 RATMF

### 2.1 隐语义模型

定义用户集合为  $U = \{u_1, u_2, \dots, u_m\}$ , 物品集合为  $O = \{o_1, o_2, \dots, o_n\}$ , 推荐系统一般由一个  $(m \times n)$  的用户-物品评分矩阵  $R$  来描述, 其中元素  $R_{ij}$  代表用户  $u_i$  对物品  $o_j$  的评分。通过计算, 推荐系统向目标用户推荐其尚未评分且计算得分较高的若干个物品。

隐语义模型假设在用户与物品之间存在  $f$  个潜在特征, 根据这些特征将  $R$  分解为一个  $(m \times f)$  的矩阵  $P$  和一个  $(n \times f)$  的矩阵  $Q$ , 使得:

$$R \approx P \times Q^T = \hat{R} \quad (1)$$

其中, 矩阵  $P$  称作用户-特征矩阵, 用来描述用户与潜在特征的关系; 矩阵  $Q$  称作物品-特征矩阵, 用来描述物品与潜在特征的关系,  $Q^T$  表示矩阵  $Q$  的转置矩阵。这样, 通过 LFM 可以将用户-物品评分矩阵  $R$  分解为两个低维矩阵的乘积, 并通过反复迭代, 使得这两个矩阵的乘积  $\hat{R}$  尽量逼近  $R$ 。这样, 用户  $u_i$  对物品  $o_j$  的预测评分可由式(2)计算得到:

$$\hat{r}_{ij} = p_i \cdot q_j = \sum_{k=1}^f p_{ik} \times q_{jk} \quad (2)$$

其中,  $p_i$  为矩阵  $P$  中与用户  $u_i$  对应的行向量,  $q_j$  为矩阵  $Q$  中与物品  $o_j$  对应的行向量。

### 2.2 标签的自适应选择

由 2.1 节可知, 矩阵  $P$  和  $Q$  的  $f$  个潜在特征对 LFM 算法的推荐效果至关重要。但是, 现有算法一般仅以物品的类别来表示潜在特征, 有时对物品的描述不够精准。由于标签一般是建立在用户对物品较为深入了解的基础之上, 所以学者们研究了将标签因子扩充到用户-特征矩阵和物品-特征矩阵中的推荐算法<sup>[14]</sup>, 将矩阵  $P$  扩充得到  $(m \times (f+t))$  的增益矩阵  $\bar{P}$ , 将矩阵  $Q$  扩充得到  $(n \times (f+t))$  的增益矩阵  $\bar{Q}$ , 其中  $t$  是标签因子的数量。

但是, 标签的增加也会使矩阵的计算开销加大, 特别是当物品标签过于稀疏时, 更是会导致训练的时间的大幅增加。为解决这一问题, 本文提出了物品的标签-评分稀疏系数 TRSF (Tag-Rating Sparsity Factor), 由式(3)计算得到:

$$TRSF(o_j) = TS(o_j) / RS(o_j) \quad (3)$$

其中,  $TS(o_j)$  与  $RS(o_j)$  分别表示物品  $o_j$  的标签稀疏度和评分稀疏度, 分别由式(4)和式(5)计算得到:

$$TS(o_j) = |T(o_j)| / t \quad (4)$$

$$RS(o_j) = |R(o_j)| / |U| \quad (5)$$

其中,  $T(o_j)$  代表物品  $o_j$  的标签集合,  $R(o_j)$  代表对物品  $o_j$  有评分的用户集合,  $t$  代表标签因子的数量,  $U$  代表全体用户集合,  $|U|$  表示用户的总数。

由式(3)~式(5)可知, 物品的标签-评分稀疏系数能够体现标签稀疏程度与评分稀疏程度的对比关系, 即: 当  $TRSF(o_j) > 1$  时, 物品的标签比例更高; 当  $TRSF(o_j) = 1$  时, 物品的标签与评分的比例一致; 当  $TRSF(o_j) < 1$  时, 物品的评分比例更高。基于这种分析, 我们仅在  $TRSF(o_j) > 1$  时, 使用标签信息预测物品的评分; 否则, 使用传统的 LFM 算法来预测用户对物品的评分。

此外, 文献[14]在利用标签扩充 LFM 时, 物品标签仅为二值化取值, 即当商品  $o_j$  具有标签  $t_k$  时, 则  $q_{jk} = 1$ ; 否则,  $q_{jk} = 0$ 。尽管与传统的 LFM 算法相比, 能够提高推荐精度, 但同样是具有标签的情况, 标签出现 1 次和标签出现 10 次, 对推荐物品的影响肯定是不一样的。因此, RATMF 算法中, 对物品  $o_j$  的标签向量  $q_j = (q_{j1}, q_{j2}, \dots, q_{jt})$ , 使用式(6)来计算每个元素:

$$q_{jk} = \begin{cases} w_{jk} / \sum_{i=1}^t w_{ji}, & o_j \text{ 具有标签 } t_k \\ 0, & o_j \text{ 不具有标签 } t_k \end{cases} \quad (6)$$

其中,  $w_{jk}$  代表物品  $o_j$  被标记具有标签  $t_k$  的次数。

这样,RATMF 算法不但利用了标签是否出现的信息,而且能够体现不同出现次数对物品的不同影响。

2.3 基于自适应选择的梯度下降

在第一次由随机初始化的方式得到矩阵  $R$  的两个分解矩阵  $P$  和  $Q$ ,并分别填充标签因子得到  $\bar{P}$  与  $\bar{Q}$  之后,可以计算  $\bar{P}$  和  $\bar{Q}$  的乘积  $\hat{R}$ 。 $\hat{R}$  的计算方法由  $P$  和  $Q$  的乘积改为  $\bar{P}$  和  $\bar{Q}$  的乘积,可以将标签的信息也加入到  $\hat{R}$  之中,从而更精准地描述物品的属性。传统方法通过比较  $R$  与  $\hat{R}$  的差异,用梯度下降法反复迭代,找到最优的  $P$  和  $Q$ 。而基于自适应选择的梯度下降每次迭代只优化矩阵  $\bar{P}$  和  $Q$ (由  $\bar{Q}$  矩阵的前  $f$  个列向量组成的矩阵,不包括标签部分)。这样做是因为  $\bar{Q}$  矩阵中标签因子是用户赋予物品的,既有较为明确的实际含义,也有具体的数值,因此,在 RATMF 算法中,并没有像其它潜在特征方法一样,通过梯度下降法来更新物品的标签。

假设用户  $u_i$  对物品  $o_j$  的实际评分为  $r_{ij}$ ,而通过式(2)计算得到的预测评分为  $\hat{r}_{ij}$ ,则实际评分与预测评分的误差为:

$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2 = (r_{ij} - \sum_{k=1}^f p_{ik} q_{jk})^2 \quad (7)$$

使用梯度下降法,由式(8)和式(9)分别对  $p_{ik}$  与  $q_{jk}$  求偏导:

$$\frac{\partial}{\partial p_{ik}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij}) q_{jk} = -2e_{ij} q_{jk} \quad (8)$$

$$\frac{\partial}{\partial q_{jk}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij}) p_{ik} = -2e_{ij} p_{ik} \quad (9)$$

在得到梯度之后, $p_{ik}$  与  $q_{jk}$  按式(10)和式(11)迭代得到新的取值:

$$p'_{ik} = p_{ik} - \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 \quad (10)$$

$$q'_{jk} = q_{jk} - \alpha \frac{\partial}{\partial q_{jk}} e_{ij}^2 \quad (11)$$

其中, $\alpha$  是一个常数,称作步长。

以上描述了一次迭代的过程,这一过程反复执行,直到误差低于某个设定的阈值,或者迭代次数超过最大次数阈值为止。

在 RATMF 算法中,为了避免过拟合,使用了如下的误差计算方法来代替式(7)。

$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2 + \lambda (\|p_i\|^2 + \|q_j\|^2) \quad (12)$$

其中, $\lambda$  是一个常数,称作正则项。

2.4 算法流程

算法 1 描述了推荐算法 RATMF 的流程。

算法 1 RATMF 推荐算法

输入:用户-物品评分矩阵  $R$ ,大小为  $(n \times t)$  的物品标签矩阵  $T$ ,最大迭代次数  $\delta$ 。

输出:推荐物品集合。

```
1 初始化  $(m \times f)$  的矩阵  $P$  和  $(n \times f)$  的矩阵  $Q$ ;  
2 由矩阵  $T$  扩充得到  $(m \times (f+t))$  的矩阵  $\bar{P}$  和  $(n \times (f+t))$  的矩阵  $\bar{Q}$ ;  
3 times=0;  
4 while times< $\delta$  do  
5   for  $\bar{P}$  的每个行向量  $p_i$  do  
6     for  $\bar{Q}$  的每个行向量  $q_j$  do  
7       if  $R$  中没有  $p_i$  对  $q_j$  的评分 then  
8         if  $TRSF(q_j)>1$  then  
9            $\hat{r}_{ij} = \sum_{k=1}^{f+t} p_{ik} \times q_{jk}$ ;  
10          else  
11             $\hat{r}_{ij} = \sum_{k=1}^f p_{ik} \times q_{jk}$ ;  
12          end if  
13        end if  
14      end for  
15    end for  
16    for  $\bar{P}$  的每个行向量  $p_i$  do  
17      for  $k=1$  to  $(f+t)$  do  
18        用式(10)更新每个元素  $p_{ik}$ ;  
19      end for  
20    end for  
21    for  $\bar{Q}$  的每个行向量  $q_j$  do  
22      for  $k=1$  to  $f$  do  
23        用式(11)更新每个元素  $q_{jk}$ ;  
24      end for  
25    end for  
26    times++;  
27 end while  
28 由最终得到的矩阵  $\bar{P}$  与  $\bar{Q}$  向目标用户作推荐。
```

3 实验验证

3.1 数据集

实验使用 MovieLens 数据集 (<http://grouplens.org/datasets/movielens>) 和自行爬取的豆瓣电影数据集。MovieLens 数据集由 2 620 个用户和 7 128 部电影组成,共有 40 838 条评分记录,其中:标签记录 33 101 条,含 2 239 个标签。豆瓣电影数据集由 8 782 个用户和 54 213 部电影组成,共有 60 293 条记录,其中:标签记录 34 451 条,含标签 1 103 个。数据集分为用户对电影的评分数据和电影的标签数据。评分记录的数据格式为“用户-电影-评分”,评分为 0~5 的实数。标签记录的

数据格式为“电影-标签-标签权重”,标签权重是一个正整数。我们用 80% 的数据作训练集,用 20% 的数据作测试集。表 1 与表 2 给出了部分示例数据。

Table 1 Example data of user-movie rating  
表 1 用户-电影评分示例数据

| userID | movieID | rating |
|--------|---------|--------|
| 75     | 32      | 4.5    |
| 75     | 51      | 3.0    |
| 76     | 24      | 5.0    |

Table 2 Example data of movie-tag  
表 2 电影-标签示例数据

| movieID | tagID | tagWeight |
|---------|-------|-----------|
| 1       | 7     | 1         |
| 1       | 326   | 10        |
| 2       | 13    | 2         |

为提高推荐精度,降低计算开销,我们在实验中删去了那些被用户标记次数过低的电影标签。凡是标记次数小于 5 的标签,都在预处理环节中被删除。

3.2 评测方法

使用平均绝对误差 MAE(Mean Absolute Error)与均方根误差 RMSE(Root Mean Squared Error)来评价算法性能。平均绝对偏差与均方根误差是最为常见的推荐效果评价参数,它们均是通过用户预测评分与物品实际评分间的差异来评价推荐结果的准确性,分别由式(13)和式(14)计算得到:

$$MAE = \frac{\sum_{i=1}^N |p_i - q_i|}{N} \tag{13}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (p_i - q_i)^2}{N}} \tag{14}$$

其中, $\{p_1, p_2, \dots, p_N\}$ 是计算得到的用户评分集合, $\{q_1, q_2, \dots, q_N\}$ 是用户实际的评分集合, $N$ 表示评分物品的数量。由式(13)和式(14)可知,当计算得到的评分与实际评分越接近,即 MAE 和 RMSE 越小时,推荐精度越高。

3.3 实验参数

RATMF 算法中,要设定潜在特征的数量  $f$ 、正则项  $\lambda$ 、步长  $\alpha$  的具体取值。一般而言, $f$  取正整数, $\lambda$  和  $\alpha$  取  $(0,1)$  内比较小的常数。对此,我们均通过实验找到参数大致的最佳取值范围,然后再通

过不断细化的方式确定参数的最终取值。表 3 和表 4 给出了 MovieLens 数据集上, $f$  和  $\lambda$  取不同值时,MAE 和 RMSE 的不同结果,这些均是在大致最佳取值范围(如  $f$  大致最佳取值在 50~100)内的实验结果。

Table 3 Change of MAE and RMSE with different values of  $f$  on MovieLens dataset

表 3 MovieLens 数据集上  $f$  取不同值时 MAE 与 RMSE 的结果变化

|      | $f$   |       |       |       |       |       |
|------|-------|-------|-------|-------|-------|-------|
|      | 50    | 60    | 70    | 80    | 90    | 100   |
| MAE  | 0.187 | 0.186 | 0.186 | 0.184 | 0.186 | 0.186 |
| RMSE | 0.274 | 0.273 | 0.273 | 0.272 | 0.273 | 0.273 |
| 均值   | 0.231 | 0.230 | 0.230 | 0.228 | 0.230 | 0.230 |

由表 3 可知,当潜在特征数量为 80 时,MAE 与 RMSE 的均值最小,故实验中  $f$  取 80。

Table 4 Change of MAE and RMSE with different values of  $\lambda$  on MovieLens dataset

表 4 MovieLens 数据集上  $\lambda$  取不同值时 MAE 与 RMSE 的结果变化

|      | $\lambda$ |       |       |       |       |       |       |
|------|-----------|-------|-------|-------|-------|-------|-------|
|      | 0.04      | 0.06  | 0.08  | 0.1   | 0.12  | 0.14  | 0.16  |
| MAE  | 0.194     | 0.191 | 0.189 | 0.186 | 0.187 | 0.191 | 0.195 |
| RMSE | 0.280     | 0.277 | 0.275 | 0.273 | 0.274 | 0.276 | 0.279 |
| 均值   | 0.237     | 0.234 | 0.232 | 0.230 | 0.231 | 0.234 | 0.237 |

由表 4 可知,当正则项  $\lambda$  为 0.1 时,MAE 与 RMSE 的均值最小,故实验中  $\lambda$  取 0.1。

在  $f=80, \lambda=0.1$  的前提下,表 5 给出了步长  $\alpha$  取不同值时,RATMF 算法收敛时的迭代次数。

Table 5 Different iteration times under different values of  $\alpha$  on MovieLens dataset

表 5 MovieLens 数据集上  $\alpha$  取不同值时梯度下降的不同迭代次数

| $\alpha$ | 0.04 | 0.06 | 0.08 | 0.10 | 0.12 | 0.14 | 0.16 |
|----------|------|------|------|------|------|------|------|
| 迭代次数     | 43   | 39   | 36   | 35   | 37   | 38   | 38   |

由表 5 可知,当步长为 0.1 时,梯度下降的迭代次数最少,故实验中  $\alpha$  取 0.1。

采用同样的方法,验证了豆瓣电影数据集上, $f=40, \lambda=0.01, \alpha=0.1$  时算法的性能最佳。

3.4 实验结果

本节将提出的 RATMF 算法与隐语义模型 LFM 算法<sup>[15]</sup>、只含标签因子的 LFM 算法(记作 LFMT)<sup>[14]</sup>、基于标签的矩阵分解推荐算法(记作 RTMF)<sup>[14]</sup>进行了实验比较。经过多次实验,4 种

算法在迭代 80 次之内都可以收敛,所以最大迭代次数  $\delta$  设定为 80。在两个数据集上各算法的 MAE 和 RMSE 分别如表 6 和表 7 所示。

Table 6 Recommendation result comparison of different methods on MovieLens dataset

表 6 MovieLens 上不同方法推荐效果比较

|      | LFM   | LFMT  | RTMF  | RATMF |
|------|-------|-------|-------|-------|
| MAE  | 0.271 | 0.328 | 0.215 | 0.186 |
| RMSE | 0.461 | 0.523 | 0.331 | 0.273 |

Table 7 Recommendation result comparison of different methods on Douban movie dataset

表 7 豆瓣电影上不同方法推荐效果比较

|      | LFM   | LFMT  | RTMF  | RATMF |
|------|-------|-------|-------|-------|
| MAE  | 0.566 | 0.678 | 0.120 | 0.076 |
| RMSE | 0.890 | 0.982 | 0.199 | 0.100 |

由表 6 和表 7 可知,RATMF 算法的 MAE 和 RMSE 结果都优于其它方法。在表 6 中:与算法 LFM 相比,RATMF 算法的 MAE 和 RMSE 分别提高了 31.37% 和 40.35%;与 LFMT 算法相比,RATMF 算法的 MAE 和 RMSE 分别提高了 43.29% 和 47.42%;与 RTMF 算法相比,RATMF 算法的 MAE 和 RMSE 分别提高了 13.49% 和 17.52%。在表 7 中:与 LFM 算法相比,RATMF 算法的 MAE 和 RMSE 分别提高了 86.57% 和 88.76%;与 LFMT 算法相比,RATMF 算法的 MAE 和 RMSE 分别提高了 88.79% 和 89.82%;与 RTMF 算法相比,RATMF 算法的 MAE 和 RMSE 分别提高了 36.67% 和 49.75%。

此外,由以上实验结果还可以看出,只使用标签而不考虑其它潜在特征的 LFMT 算法的性能最差,这说明尽管在潜在特征的基础上使用标签可以提高推荐精度,但只使用标签的效果却未必好。

与推荐精度最为接近的 RTMF 算法相比<sup>[14]</sup>,本文提出的 RATMF 算法引入了自适应标签选择策略,来确定是否使用标签及标签向量元素的数值。为了进一步说明这一策略的优势,我们还比较了这两种算法使用梯度下降法优化的收敛迭代次数,两个数据集上的对比结果分别如图 1 和图 2 所示。

由图 1 可知:在 MovieLens 数据集上,RTMF 算法关于 MAE 和 RMSE 的收敛迭代次数分别为 70 次和 60 次,而 RATMF 算法关于这两个参数的收敛迭代次数均为 35 次。由图 2 可知:在豆瓣电影数据集上,RTMF 算法关于 MAE 和 RMSE 的收敛迭代次数分别为 35 次和 30 次,而 RATMF 算法关于这两个参数的收敛迭代次数分别为 15 次

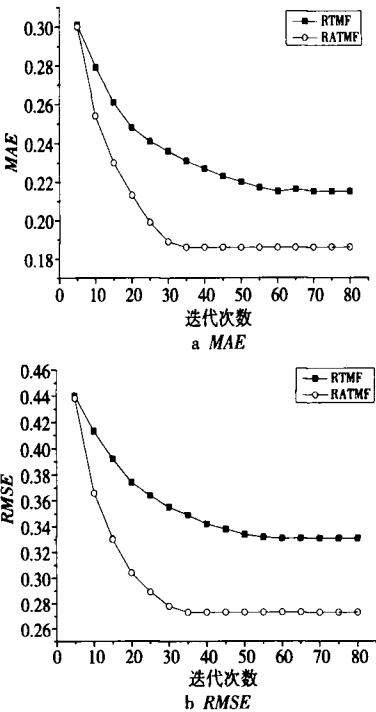


Figure 1 Comparison of iteration times for convergence between the two methods on MovieLens dataset  
图 1 MovieLens 上两种算法的收敛迭代次数对比

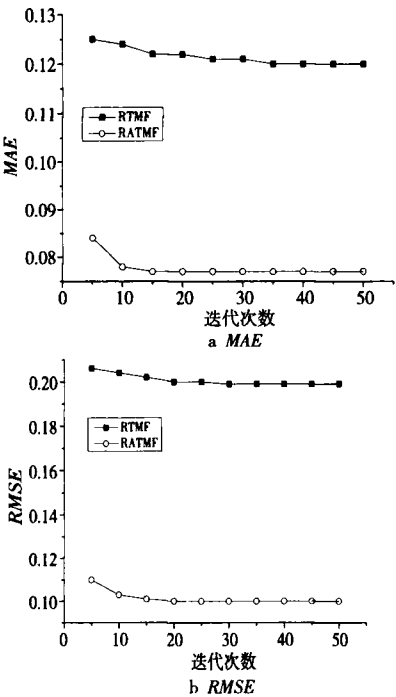


Figure 2 Comparison of iteration times for convergence between the two methods on Douban movie dataset  
图 2 豆瓣电影上两种算法的收敛迭代次数对比

和 20 次。可以看出,自适应标签选择策略不但可以提高推荐精度,而且可以降低计算开销。

4 结束语

本文将标签融入 LFM 算法,提出了一种基于

标签自适应选择的矩阵分解推荐算法。通过标签-评分稀疏系数来确定计算预测评分时是否使用标签,较好地平衡了潜在特征与标签的使用。通过标签的出现频率来计算标签向量,使得标签向量具有更高的个性化程度。实验结果表明,提出的算法具有较高的推荐精度和较快的收敛速度。

### 参考文献:

- [1] Aggarwal C C. Recommender systems [M]. Cham, Switzerland: Springer, 2016.
- [2] Zhu Yang-yong, Sun Jing. Recommender system: Up to now [J]. Journal of Frontiers of Computer Science and Technology, 2015, 9(5): 513-525. (in Chinese)
- [3] Li Jian-feng, Qin Zheng. A Slope One collaborative filtering recommendation algorithm based on local nearest neighbors [J]. Computer Engineering & Science, 2017, 39(7): 1346-1351. (in Chinese)
- [4] Soares M, Viana P. Tuning metadata for better movie content-based recommendation systems [J]. Multimedia Tools and Applications, 2015, 74(17): 7015-7036.
- [5] Song Wei, Liu Peng. Hybrid recommendation method based on frequent itemset and collaborative filtering [J]. Journal of Shanxi University (Natural Science Edition), 2017, 40(1): 35-43. (in Chinese)
- [6] Mahboob V A, Jalali M, Jahan M V, et al. Swallow: Resource and tag recommender system based on heat diffusion algorithm in social annotation systems [J]. Computational Intelligence, 2017, 33(1): 99-118.
- [7] Wang Sen. A location recommendation algorithm based on location-based social networks [J]. Computer Engineering & Science, 2016, 38(4): 667-672. (in Chinese)
- [8] Yu Y, Wang C, Wang H, et al. Attributes coupling based matrix factorization for item recommendation [J]. Applied Intelligence, 2017, 46(3): 521-533.
- [9] Aleksandrova M, Brun A, Boyer A, et al. Identifying representative users in matrix factorization-based recommender systems: Application to solving the content-less new item cold-start problem [J]. Journal of Intelligent Information Systems, 2017, 48(2): 365-397.
- [10] Sarwar B M, Karypis G, Konstan J A, et al. Application of dimensionality reduction in recommender system—a case study [C]//Proc of Web Mining for E-Commerce-Challenges and Opportunities, 2000: 1.
- [11] Luo X, Ouyang Y, Xiong Z. Improving latent factor model based collaborative filtering via integrated folksonomy factors [J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2011, 19(2): 307-327.
- [12] Kim B S, Kim H, Lee J, et al. Improving a recommender system by collective matrix factorization with tag information [C]//Proc of International Symposium on Soft Computing & Intelligent Systems, 2015: 980-984.
- [13] Lu C, Shen B, Zhang L, et al. Tag recommendation via robust probabilistic discriminative matrix factorization [C]//Proc of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, 2016: 1170-1174.
- [14] Fang Bing, Niu Xiao-ting. Tag-based matrix factorization recommendation algorithm [J]. Application Research of Computers, 2017, 34(4): 1022-1025. (in Chinese)
- [15] Hofmann T. Latent semantic models for collaborative filtering [J]. ACM Transactions on Information Systems, 2004, 22(1): 89-115.

### 附中文参考文献:

- [2] 朱扬勇, 孙婧. 推荐系统研究进展 [J]. 计算机科学与探索, 2015, 9(5): 513-525.
- [3] 李剑锋, 秦拯. 一种基于局部近邻 Slope One 协同过滤推荐算法 [J]. 计算机工程与科学, 2017, 39(7): 1346-1351.
- [5] 宋威, 刘朋. 基于频繁项集与协同过滤的混合推荐方法 [J]. 山西大学学报(自然科学版), 2017, 40(1): 35-43.
- [7] 王森. 基于位置社交网络的地点推荐算法 [J]. 计算机工程与科学, 2016, 38(4): 667-672.
- [14] 方冰, 牛晓婷. 基于标签的矩阵分解推荐算法 [J]. 计算机应用研究, 2017, 34(4): 1022-1025.

### 作者简介:



宋威(1980-),男,辽宁抚顺人,博士,教授,CCF 会员(19725S),研究方向为数据挖掘和推荐系统。E-mail: songwei@ncut.edu.cn

SONG Wei, born in 1980, PhD, professor, CCF member (19725S), his research interests include data mining, and recommender system.



李雪松(1994-),男,北京人,硕士生,研究方向为推荐系统。E-mail: 1187220556@qq.com

LI Xue-song, born in 1994, MS candidate, his research interest includes recommender system.