

SVD 推荐算法在传统服饰商城的应用

钟志攀,袁仕芳,梁荣锋,朱翡虹

(五邑大学 数学与计算科学学院,广东 江门 529020)

摘要:随着互联网的普及和信息技术的发展,人们日常生活可以接触到的信息越来越丰富,产生信息数据的速度也越来越快。在拥有海量信息数据的同时,过量的信息不可避免地导致了“信息过载”问题,而推荐系统正是解决这个问题的有效方法之一。首先介绍了奇异值分解(SVD)和 SVD 推荐算法之间的联系,然后叙述了 SVD 推荐算法的思路并给出 SVD 算法模型的最优损失函数。在此基础上,建立一个传统服饰商城的推荐系统,利用 158 组样本数据测试该系统,得出推荐系统的均方根误差为 0.999 4,平方绝对值误差为 0.736 6。此外还计算了样本数据的误差百分比,结果表明绝大多数测试样本的误差百分比在 30% 以内。基于 SVD 推荐模型的特点和实验结果得出,SVD 推荐模型适用于小型电商平台的商城用户的个性化推荐。

关键词:SVD 推荐算法;推荐系统;服饰商城;最速下降算法;奇异值分解

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2020)02-0211-05

doi:10.3969/j.issn.1673-629X.2020.02.041

Application of SVD Recommendation Algorithm in Traditional Apparel Mall

ZHONG Zhi-pan, YUAN Shi-fang, LIANG Rong-feng, ZHU Fei-hong

(School of Mathematics and Computational Science, Wuyi University, Jiangmen 529020, China)

Abstract: With the popularity of the Internet and the development of information technology, the information that people can access in daily life is becoming more and more abundant, and the speed of generating information data is getting faster and faster. While we have a huge amount of information data, excessive information inevitably leads to the problem of “information overload”, and the recommendation system is one of the effective methods to solve this problem. In this study, the relationship between singular value decomposition (SVD) and SVD recommendation algorithm is introduced firstly, and then the idea of SVD recommendation algorithm is described and the optimal loss function of the SVD algorithm model is given. Based on that, a recommendation system of traditional clothing store is established. The system is tested with 158 sets of sample data, of which the root mean square error is 0.999 4 and the squared absolute error is 0.736 6. In addition, we also calculate the error percentage of the sample data. The results show that the error percentage of most test samples is within 30%. Based on the characteristics of the SVD recommendation model and the experimental results, it is concluded that the SVD recommendation model is suitable for personalized recommendation of mall users on small e-commerce platforms.

Key words: SVD recommendation algorithm; recommendation system; clothing mall; steepest descent algorithm; singular value decomposition

0 引言

随着互联网的普及和信息技术的发展,人们日常生活可以接触到的信息越来越丰富,产生信息数据的速度也越来越快。但是“信息过载”问题也越来越严重。想要从这海量的信息中获取目的信息需要耗费大

量的时间和精力。

人们解决“信息过载”问题的方法主要有建立搜索引擎、建立推荐系统(recommended system)^[1-3]。推荐系统是比搜索引擎处理问题更为人性化的工具,它可以根据用户的个人爱好、历史浏览记录等为用户推

收稿日期:2019-03-16

修回日期:2019-07-22

网络出版时间:2019-11-07

基金项目:2018 年全国大学生创新创业训练计划创新训练项目(201811349010);广东省大学生创新创业训练计划创业实践项目(201811349062S)

作者简介:钟志攀(1997-),男,研究方向为数据处理;袁仕芳,博士,教授,硕导,通讯作者,研究方向为数据处理和数值分析。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20191107.0918.064.html>

荐所感兴趣的内容^[4-6]。建立一个好的推荐系统可以节省用户查找感兴趣内容的时间,增强用户体验,还可以提高商品销售量。推荐系统在很多行业都有应用,例如新闻行业中的《今日头条》,个性音乐软件“网易云音乐”,尤其是电商行业,最著名的例子就有亚马逊的“尿不湿和啤酒”^[7]。推荐系统在工业上最经典的应用莫过于协同过滤算法(CF)^[8-9],该算法通过找到与用户有相同品味的用户,然后将相似用户过去喜欢的物品推荐给用户。

文中首先介绍了奇异值分解(SVD)和 SVD 推荐算法之间的联系,以及 SVD 推荐算法原理,在此基础上,建立了一个传统服饰商城的推荐系统。然后利用传统服饰商城的用户样本数据测试该推荐系统的性能。

1 算法原理

协同过滤算法是推荐系统中的经典算法。协同过滤算法主要分为两类,一类是基于邻域(neighborhood methods)的协同过滤算法,另一类是隐语义模型(latent factor models)的协同过滤算法^[10],后者是利用矩阵分解(matrix factorization)算法实现的。

为了评价推荐算法的性能,一般使用平方绝对值误差(MAE)和均方根误差(RMSE)测试推荐系统的准确性,公式如下:

MAE(f) = \frac{1}{|R_{\text{test}}|} \sum_{r_{(u_i, c_j)} \in R_{\text{test}}} |f(u_i, c_j) - r_{(u_i, c_j)}| \tag{1}

RMSE(f) = \sqrt{\frac{1}{|R_{\text{test}}|} \sum_{r_{(u_i, c_j)} \in R_{\text{test}}} |f(u_i, c_j) - r_{(u_i, c_j)}|^2} \tag{2}

其中, R_{test} 表示测试数据集, u_i 表示用户 i , c_j 表示商品 j , $r_{(u_i, c_j)}$ 表示用户 u_i 对商品 c_j 的评分。

1.1 奇异值分解

SVD 算法是在奇异值分解(singular value decomposition)^[11]的基础上建立起来的矩阵分解算法。对于矩阵 $A \in R^{n \times m}$, 它的奇异值分解为:

A = U \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} V^T \tag{3}

其中, $\Sigma_r = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_r)$, $r = \text{rank}(A)$,

$U \in R^{n \times n}$ 和 $V \in R^{m \times m}$ 是正交矩阵。 U 的列向量称为左奇异值向量; Σ_r 对角线上的值是 A 非零奇异值; V^T 是 V 的转置, V 的列向量称为右奇异向量。

通常定义奇异值为 σ_i , 在矩阵 Σ 中沿对角线从大到小排列为:

\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r, r \leq \min\{n, m\}

σ_i 的下降速度很快, 大多数情况下, 前 10% 甚至 1% 的奇异值的和就可占有所有奇异值之和的 99% 以上, 所以这里 r 往往远小于 n 和 m 。根据奇异值的这个性质, 可以用一部分奇异值近似地描述矩阵所包含的信息, 即有:

A_{n \times m} \approx U_{n \times k} \Sigma_{k \times k} V_{k \times m}^T \tag{4}

其中 $k \leq r$ 。

1.2 SVD 算法

传统的推荐系统一般会通过用户对商品进行评价, 得出评分矩阵 R (rating matrix), 再通过评分矩阵进行分类、预测用户喜好, 从而进行商品推荐。当用户数量比较大, 商品种类和数量比较多时, 很难做到让所有的客户对所有的商品进行评分, 此时评分矩阵 R 就会出现很多的缺失值。

这些缺失的评分可以看作是用户 u_i 对于商品 c_j 的未知的潜在评分。SVD 算法的思路就是在评分矩阵 R 缺失的情况下, 对矩阵进行分解, 从而得到用户因子矩阵 U (users features matrix) 和商品因子矩阵 C (item features matrix), 最后通过矩阵 U 和 C 来预测用户对商品的潜在评分。

由式(6)推广到分解评分矩阵 R , 有以下等式:

U_{n \times r} = U_{n \times n} \Sigma_{r \times r} \tag{5}

R_{n \times m} = U_{n \times r} C_{r \times m}^T \tag{6}

对于评分矩阵 R 的分解, 用户因子矩阵 U 等价于对矩阵 R 的左奇异矩阵 $U_{n \times n}$ 乘上奇异值矩阵 $\Sigma_{r \times r}$, 而商品因子矩阵 C 等价于对矩阵 R 的右奇异值矩阵 $V_{r \times m}^T$ 。

用隐语义模型来进行协同过滤的目标是揭示隐藏的特征, 这些隐藏的特征能够解释观测到的评分。SVD 是为了识别隐语义因子而发展起来的, 然而由于大部分评分值缺失, 把 SVD 应用到 CF 领域的显式评分变得相对困难。例如表 1。

表 1 评分矩阵 R

	item_id = 1	item_id = 2	...	item_id = m
user_id = 1	$r_{(1,1)}$	$r_{(1,2)}$...	$r_{(1,m)}$
user_id = 2	$r_{(2,1)}$	$r_{(2,2)}$...	$r_{(2,m)}$
...
user_id = n	$r_{(n,1)}$	$r_{(n,2)}$...	$r_{(n,m)}$

实际情况中往往只能获得形如表 1 所示的评分矩阵, 所有用户无法对所有商品进行评价, 造成评分矩阵

存在缺失值。当 $r_{(i,j)}=0$,表示用户 i 对商品 j 的评分缺失,如表1所示。存在缺失的现实意义就是用户尚未购买或者与某商品有一些交互动作,因此,通过预测缺失的评分,在用户尚未评价的商品中,通过用户原有的评分记录和其他用户的相似性这一先验知识,对用户

没有评分的商品进行评分预测。

SVD 算法在奇异值分解的基础上,将存在缺失值的评分矩阵 \boldsymbol{R} 分解成形如表2和表3的用户因子矩阵 \boldsymbol{U} 和商品因子矩阵 \boldsymbol{C} 。

表2 用户因子矩阵 \boldsymbol{U}

	user_id = 1	user_id = 2	...	user_id = m
factor_1	$\text{ur}_{(1,1)}$	$\text{ur}_{(1,2)}$...	$\text{ur}_{(1,m)}$
factor_2	$\text{ur}_{(2,1)}$	$\text{ur}_{(2,2)}$...	$\text{ur}_{(2,m)}$
...
factor_n	$\text{ur}_{(n,1)}$	$\text{ur}_{(n,2)}$...	$\text{ur}_{(n,m)}$

表3 商品因子矩阵 \boldsymbol{C}

	item_id = 1	item_id = 2	...	item_id = m
factor_1	$\text{cr}_{(1,1)}$	$\text{cr}_{(1,2)}$...	$\text{cr}_{(1,m)}$
factor_2	$\text{cr}_{(2,1)}$	$\text{cr}_{(2,2)}$...	$\text{cr}_{(2,m)}$
...
factor_n	$\text{cr}_{(n,1)}$	$\text{cr}_{(n,2)}$...	$\text{cr}_{(n,m)}$

可以不用知道这两个矩阵中的因子 (factors) 具体是什么,第一个因子可以是商品的价格,也可以是商品的外观的美观程度。关于“因子”,可以把它认为是商品的子属性。对同一个因子进行评分,在用户因子矩阵中表示用户偏爱于这种商品的属性的程度,而在商品因子矩阵中则表示此属性在该商品反映的程度。

对于要推荐的商品,如衣服,价格是衣服的属性之一,假设这件衣服的价格很低廉,那么相对应评分就会高些,此时用户在购买衣服时十分注重衣服的价格,那么对于衣服的价格的偏爱程度就会高一点。

法),得到了上面的用户因子矩阵 \boldsymbol{U} 和商品因子矩阵 \boldsymbol{C} ,通过分解出来的因子评分可以预测出用户对新商品的评分,在这些新商品的评分中选取前 n (Top - N)个预测评分最高的物品,构造出推荐列表进行推荐。

预测用户 u_i 对商品 c_j 的评分,利用用户因子矩阵 \boldsymbol{U} 和商品因子矩阵 \boldsymbol{C} 计算可得预测评分:

$$\hat{r}_{(i,j)} = \sum_{k=1}^n \text{ur}_{(k,i)} \cdot \text{cr}_{(k,j)}$$

(7)

利用式(7)可以得到表4。

通过矩阵分解 (可以是任何合理的矩阵分解算

表4 评分矩阵 $\hat{\boldsymbol{R}}$

	item_id = 1	item_id = 2	...	item_id = m
user_id = 1	$\hat{r}_{(1,1)}$	$\hat{r}_{(1,2)}$...	$\hat{r}_{(1,m)}$
user_id = 2	$\hat{r}_{(2,1)}$	$\hat{r}_{(2,2)}$...	$\hat{r}_{(2,m)}$
...
user_id = n	$\hat{r}_{(n,1)}$	$\hat{r}_{(n,2)}$...	$\hat{r}_{(n,m)}$

1.3 SVD 算法的“学习”预测

通过上述内容,SVD 算法实现推荐的主要思路是:通过矩阵分解技术将评分矩阵 \boldsymbol{R} 分解成用户因子矩阵 \boldsymbol{U} 和商品因子矩阵 \boldsymbol{C} ,再通过这两个矩阵的合成从而预测用户对商品的评分。但是矩阵 \boldsymbol{U} 和 \boldsymbol{C} 并不是直接通过矩阵分解得出,而是通过学习的方法得到。

最小,于是定义一个损失函数:

$$E = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^m I_{(i,j)} (r_{(i,j)} - \hat{r}_{(i,j)})^2 + \frac{k_u}{2} \sum_{n=1}^n \| \boldsymbol{p}_i \|^2 + \frac{k_c}{2} \sum_{j=1}^m \| \boldsymbol{q}_j \|^2$$

(9)

其中 $I \in \{0,1\}_{n \times m}$,作为一个指示器,指示相应位置是否有评分,有评分为“1”,没有评分为“0”,最后两项是正则化项; k_u 和 k_c 是常数,防止过拟合^[12]。

考虑由于个体差异可能导致评分过低或过高,使用 baseline predictors 基准预测。将平均评分设为 μ ,设用户做出的评分相对于平均评分的偏差为 b_{u_i} ,设商

预测用户 u_i 对商品 c_j 的评分可以表示为:

$$\hat{r}_{(i,j)} = \boldsymbol{q}_j^T \boldsymbol{p}_i$$

(8)

其中, \boldsymbol{q} 表示矩阵 \boldsymbol{C} 的第 j 列分量, \boldsymbol{p} 表示矩阵 \boldsymbol{U} 的第 i 列分量。期望实际评分 $r_{(i,j)}$ 和 $\hat{r}_{(i,j)}$ 之间的误差

品 c_j 相对于平均分的偏差为 b_{c_j} , 得到:

$$\hat{r}_{(i,j)} = \mu + b_{c_j} + b_{u_i}$$

(10)

结合式(8)和式(10), 有:

$$\hat{r}_{(i,j)} = \mu + b_{c_j} + b_{u_i} + \mathbf{q}_j^T \mathbf{p}_i$$

(11)

于是损失函数可以重新表述为:

$$E = \sum_{i=1}^n \sum_{j=1}^m I_{(i,j)} (r_{(i,j)} - \mu - b_{u_i} - b_{c_j} - \mathbf{q}_j^T \mathbf{p}_i)^2 + \lambda (\|\mathbf{p}_i\|^2 + \|\mathbf{q}_j\|^2 + b_{u_i}^2 + b_{c_j}^2)$$

(12)

利用 Funk 等人^[13]推广的随机梯度最速下降法来求解 $\min(E)$ ^[11], 即最优化损失函数的参数值: $\mathbf{q}_j^T \mathbf{p}_i$ 、 μ 、 b_{c_j} 、 b_{u_i} 、 λ , 满足:

$$\min E = \sum_{i=1}^n \sum_{j=1}^m I_{(i,j)} (r_{(i,j)} - \mu - b_{u_i} - b_{c_j} - \mathbf{q}_j^T \mathbf{p}_i)^2 + \lambda (\|\mathbf{p}_i\|^2 + \|\mathbf{q}_j\|^2 + b_{u_i}^2 + b_{c_j}^2)$$

(13)

对式(13)求导到求解函数:

$$\begin{cases} e_{(i,j)} = r_{(i,j)} - \hat{r}_{(i,j)} \\ b_{u_i} \leftarrow b_{u_i} + \gamma(e_{(i,j)} - \lambda \cdot b_{u_i}) \\ b_{c_j} \leftarrow b_{c_j} + \gamma(e_{(i,j)} - \lambda \cdot b_{c_j}) \\ \mathbf{p}_i \leftarrow \mathbf{p}_i + \gamma(e_{(i,j)} - \lambda \cdot \mathbf{p}_i) \\ \mathbf{q}_j \leftarrow \mathbf{q}_j + \gamma(e_{(i,j)} - \lambda \cdot \mathbf{q}_j) \end{cases}$$

(14)

其中, γ 为学习率。

利用评分矩阵中已知的评分求解上述函数(14), 训练得到损失函数值最优时的参数, 从而确定损失函数, 完成对模型的训练。通过损失函数可以预测用户对商品的评分, 完成推荐任务。

2 算法实现

实验是在 Intel-Core i7 双核处理器, CPU1 2.60 GHz, CPU2 2.0 GHz, 内存 4 G 的计算机上进行, 采用 Window8.1 操作系统, 编程语言是 Python 3.6。

2.1 数据集

实验数据来自传统汉服网站“衣缘巷”的用户统计数据。随机选取了 158 名用户对 20 件服装进行评分, 评分范围是 1 ~ 7。

2.2 算法实现与结果分析

算法流程如图 1 所示。

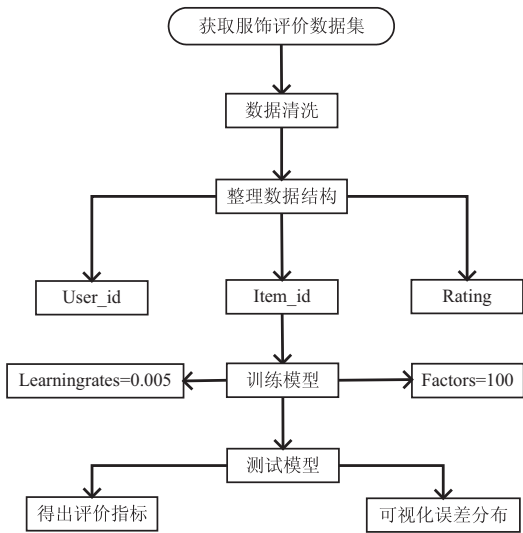


图 1 算法流程

将获取到的数据进行数据清理, 剔除 user_id、item_id 数据内容缺失的数据项。再将数据集重新整合成 (user_id, item_id, rating) 三元组结构。使用交叉验证^[14]来验证数据, 将数据随机分成三份进行三组测试, 第一组实验选定其中两份数据作为训练数据, 剩下的一份作为测试数据; 其他两组数据也是执行同样的操作。

初始化式(13)的学习率 γ 为 0.002, 因子 factors 的维数初始化为 100, 表示将要分解的评分矩阵的因子最多有 100 个。模型测试指标见表 5。

表 5 模型测试指标

指标	Fold_1	Fold_2	Fold_3	Mean	Std
RMSE	0.991 6	1.008 5	0.997 9	0.999 4	0.007 0
MAE	0.722 3	0.742 9	0.744 7	0.736 6	0.010 2
训练时间	0.30	0.32	0.83	0.48	0.25
测试时间	0.02	0.02	0.01	0.01	0.00

通过机器学习不断训练之后, 得到较为满意的模型。其中将测试集合随机分为 3 部分, 交叉验证模型性能, 得到模型 $\overline{\text{RMSE}} = 0.999\ 4$, $\text{Std} = 0.007\ 0$; 对于指标 RMSE, 模型除了较为良好, 还是较为稳定的。指标 $\overline{\text{MAE}} = 0.736\ 6$, $\text{Std} = 0.010\ 2$, 也是较为良好和稳定的。

图 2 反映了实际评分和预测评分之间的绝对误差

主要集中在“1”分左右, 意味着该模型预测用户的评分出现的误差大部分不超过“1”。

通过计算客户对商品的实际评分和模型的预测评分之间的绝对误差占总分“7”的百分比, 观察模型的总体预测性能, 如图 3 所示。可以发现误差百分比超过 50% 的误差基本没有, 误差百分比大部分集中在 30%, 其中误差百分比为 10% 的误差的样本数量

居多。

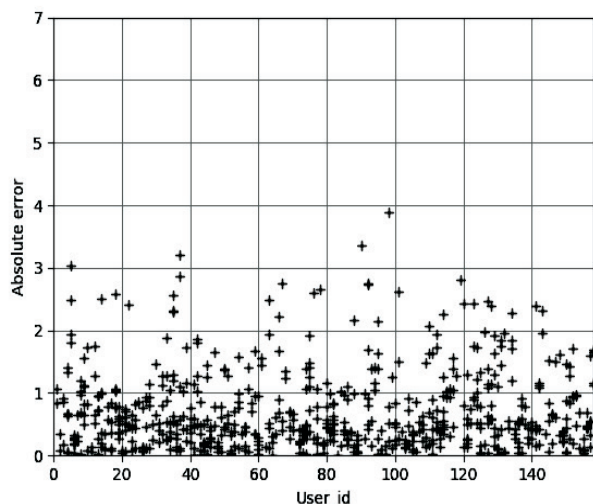


图2 绝对评分误差散点分布

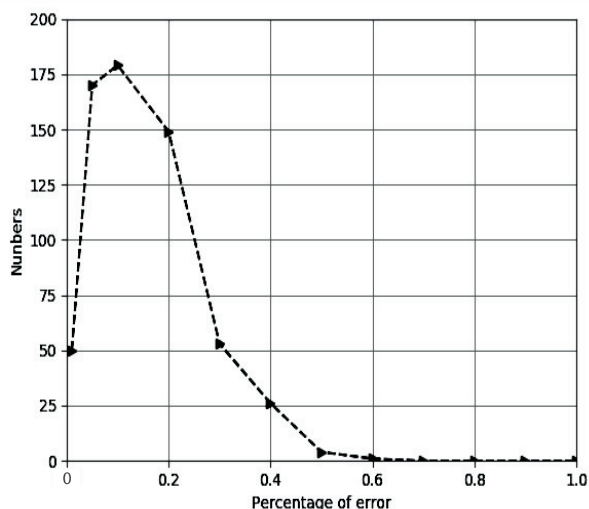


图3 模型预测误差分布

3 结束语

使用传统服饰商城的用户评分数据建立了基于SVD的推荐系统,通过预测用户对服饰的评分,实现了对传统服饰的推荐任务。使用评价推荐系统优劣的两个指标RMSE和MAE,在该模型中 $\overline{\text{RMSE}}=0.999\ 4$, $\overline{\text{MAE}}=0.736\ 6$,总体方差较小,反映了模型总体的准确性和鲁棒性。而模型中测试样本的数量在误差百分比为30%时占有大多数。

总体来说,该模型相对于现行的其他推荐算法是可靠的。SVD算法的优势在于可以较为准确地预测用户评分,实现精准推荐。但在用户和商品基数特别

大的情况下,SVD算法会因矩阵分解的时间效率低而优势不在,表明该算法非常适合用户数量不够巨大,商品种类不太繁多的小型在线商城。

参考文献:

- [1] BURKE R. Hybrid web recommender systems[M]//The adaptive web. Berlin: Springer, 2007: 377-408.
- [2] 王嘉菲,朱志锋. 基于协同过滤算法的视频智能推荐系统[J]. 湖北大学学报:自然科学版, 2019, 41(2): 202-207.
- [3] 刘欣雨. 基于SVD++在线图书推荐系统的研究[J]. 电子测试, 2019(Z1): 50-52.
- [4] AMATRIAIN X. Mining large streams of user data for personalized recommendations[J]. ACM SIGKDD Explorations Newsletter, 2013, 14(2): 37-48.
- [5] 邓磊,古发辉,李海平. 用户多兴趣下的个性化推荐算法分析[J]. 信息与电脑:理论版, 2019(3): 69-71.
- [6] 庞海龙,赵辉,李万龙,等. 融合协同过滤的线性回归推荐算法[J]. 计算机应用研究, 2019, 36(5): 1302-1304.
- [7] LINDEN G, SMITH B, YORK J. Amazon. com recommendations; item-to-item collaborative filtering[J]. IEEE Internet Computing, 2003, 7(1): 76-80.
- [8] SCHAFER J B, KONSTAN J A, RIEDL J. E-commerce recommendation applications[J]. Data mining and Knowledge Discovery, 2001, 5(1-2): 115-153.
- [9] 陆俊尧,李玲娟. 基于Spark的协同过滤算法并行化研究[J]. 计算机技术与发展, 2019, 29(1): 85-89.
- [10] DEERWESTER S, DUMAIS S T, FURNAS G W, et al. Indexing by latent semantic analysis[J]. Journal of the American Society for Information Science, 1990, 41(6): 391-407.
- [11] KOREN Y. Factorization meets the neighborhood; a multifaceted collaborative filtering model[C]//Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining. Las Vegas, Nevada, USA: ACM, 2008: 426-434.
- [12] BELL R, KOREN Y, VOLINSKY C. Modeling relationships at multiple scales to improve accuracy of large recommender systems[C]//Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining. San Jose, USA: ACM, 2007: 95-104.
- [13] PATEREK A. Improving regularized singular value decomposition for collaborative filtering[C]//Proceedings of KDD cup and workshop. California, USA: ACM, 2007: 5-8.
- [14] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.