基于 SVD 填充的混合推荐算法

刘晴晴 罗永龙 汪逸飞 郑孝遥 陈 文
(安徽师范大学计算机与信息学院 安徽 芜湖 241002)[1]
(安徽师范大学网络与信息安全安徽省重点实验室 安徽 芜湖 241002)[2]

摘 要 随着互联网技术的发展,信息过载问题日益严重,推荐系统是缓解该问题的有效手段。针对协同过滤中因数据稀疏和冷启动导致的推荐效率低下问题,提出基于 SVD 填充的混合推荐算法。首先,采用奇异值分解技术分解项目评分矩阵,通过随机梯度下降法填充稀疏矩阵;然后,在矩阵中加入时间权重,优化用户相似度,同时在项目矩阵中加入 Jaccard 系数优化项目相似度;接着,综合基于项目和基于用户的协同过滤计算预测评分,从而选择最优项目;最后,在 MovieLens 和 Jester 数据集中将所提算法与传统算法进行实验对比,证明了所提算法的有效性。

关键词 推荐系统,协同过滤,奇异值分解,填充矩阵,时间权重
中图法分类号 TP391 **文献标识码** A

Hybrid Recommendation Algorithm Based on SVD Filling

LIU Qing-qing LUO Yong-long WANG Yi-fei ZHENG Xiao-yao CHEN Wen
(School of Computer and Information, Anhui Normal University, Wuhu, Anhui 241002, China)
(Anhui Provincial Key Laboratory of Network and Information Security, Anhui Normal University, Wuhu, Anhui 241002, China)

Abstract With the development of Internet technology, the issue of information overload is becoming increasingly serious. The recommendation system is an effective means to alleviate this problem. Focusing on the problem of low recommendation efficiency caused by sparse data and cold start in collaborative filtering, this paper proposed a hybrid recommendation algorithm based on SVD filling. Firstly, Singular Value Decomposition technique is used to decompose the user-item score matrix, and sparse matrix is filled by stochastic gradient descent method. Secondly, time weights are added to optimize the user similarity in the user matrix. At the same time, Jaccard coefficients are added to optimize the item similarity in the item matrix. Then, item-based and user-based collaborative filtering are combined to calculate prediction scores and select the optimal project. Finally, the proposed algorithm is compared with other existing algorithms on Movielens and Jester data set, and the result of experiments verifies that the effectiveness of the proposed algorithm.

Keywords Recommendation system, Collaborative filtering, Singular value decomposition, Fill matrix, Time weight

1 引言

随着互联网的快速发展以及大数据时代的来临,用户从海量信息中寻找自己感兴趣的信息成为一个棘手的问题[1]。虽然常用的搜索引擎可以通过检索关键字在一定程度上缓解用户信息负担,但仍不能满足用户个性化信息需求。为用户提供推荐功能的网站和电子商务平台越来越受欢迎,亚马逊表示 35%的销售来自推荐系统;谷歌新闻称推荐系统使其文章的阅读量提高了 38%;Netflix 公司也表示 60%的影片租赁业务来自推荐系统[2]。由此可知,推荐系统可以有效缓解用户信息过载问题,提高服务质量。因此,如何为人们提供更准确、高效的推荐服务已经成为研究者们关注的热点问题。

协同过滤推荐(Collaborative Filtering, CF)是最常见的一种推荐方法。Grouplens[3] 和 Video 推荐系统[4] 是早期的协同过滤推荐,基于协同过滤思想,通过其他人的意见分别为用户提供新闻和视频等推荐服务。协同过滤推荐一般分为基于用户的 CF[5] 和基于项目的 CF [6-7]。基于用户的 CF 方法(User-based CF, UCF)是通过计算用户与用户的相似度得到邻居集合,根据邻居集合的评分预测用户对某产品的喜欢程度;基于项目的 CF 方法(Item-based CF, ICF)通过计算项目与项目之间的相似度,预测用户对该项目的喜欢程度。另外,研究者们开始关注通过加权融合用户和项目的 CF 方法形成混合推荐方法[8-11],从而提高推荐系统精度。然而,CF 方法也面临很多挑战,实际应用中项目评分矩阵一般比较稀疏,基于这些稀疏数据提供的推荐面临精确度不高,且无法向新用户或现有用户推荐新项目等问题。大数据时代,用户和项目的数量迅速增长,导致用户评分数据集更加稀疏。传统的相似度计算方法在这种情况下效率不佳,使得推荐系统的质量明显下降。用户评分矩阵稀疏是造成质量差的主要原因。

SVD 算法在文献[12]中最先用于协同过滤推荐,在 Net-

本文受国家自然科学基金项目(61672039,61772034),安徽省自然科学基金项目(1808085MF172)资助。
刘晴晴(1994—),女,硕士生,主要研究领域为推荐隐私保护;**罗永龙**(1972—),男,教授,博士生导师,主要研究领域为空间数据处理、信息安全、隐私保护,E-mail:qliuahnu@163.com(通信作者);**汪逸飞**(1993—),男,硕士生,主要研究领域为信息安全、隐私保护研究;**郑孝遥**(1981—),男,博士生,副教授,主要研究领域为信息安全、个性化推荐;**陈 文**(1979—),男,博士生,教授,主要研究领域为空间数据处理、信息安全。

flix Prize 中表现出了优秀的预测准确性和稳定性，迅速成为最流行的推荐算法之一[13]。

为了解决上述问题，本文基于矩阵分解思想，提出了一种基于奇异矩阵分解填充的混合推荐算法（Hybrid Collaborative Filtering based Singular Value Decomposition Filling，SVD-HCF）。该推荐算法融合了 UCF 和 ICF 推荐方法，首先采用 SVD 技术分解项目评分矩阵，通过随机梯度下降法填充稀疏矩阵。然后加入时间权重，优化用户相似度计算，同时加入 Jaccard 系数优化项目相似度计算。最后综合计算预测评分，选择最优项目。实验表明，本文提出的混合推荐方法相比传统的 CF 方法可以有效缓解冷启动问题，提高推荐的准确度。

2　相关工作

2.1　协同过滤推荐算法

许多商业网站使用 CF 推荐算法为用户提供建议[14]。CF 算法是根据项目评分矩阵计算用户或物品的相似度来预测目标用户对项目的喜好程度，具有良好的可扩展性，易于实施。

2.1.1　基于用户的协同过滤算法

基于用户的协同过滤推荐算法根据项目评分矩阵计算出用户的最相似邻居，根据最相似邻居的评分预测该用户的偏好。

其中相似性度量方法是协同过滤推荐的关键因素[9]。最常用的是皮尔森相关系数计算方法，具体计算方法如下：

$$sim(a,b)=\frac{\sum\limits_{v\in V_{ab}}(r_{av}-\overline{r_a})(r_{bv}-\overline{r_b})}{\sqrt{\sum\limits_{v\in V_{ab}}(r_{av}-\overline{r_a})^2(r_{bv}-\overline{r_b})^2}} \tag{1}$$

其中，$V_{ab}=V_a\cap V_b$ 表示用户 a 和用户 b 评分项目的交集；r_{av} 表示用户 a 对项目 v 的评分；$\overline{r_a}$ 表示用户 a 的平均评分。

因此传统的 UCF 推荐可以通过相似度计算出用户 u 对项目 v 的预测评分，如式(2)所示：

$$P_{uv}=\overline{r_u}+\frac{\sum\limits_{a\in U_x}sim(a,u)(r_{av}-\overline{r_a})}{\sum\limits_{a\in U_x}sim(a,u)} \tag{2}$$

其中，U_x 表示用户 u 的邻居用户集合。

2.1.2　基于项目的协同过滤算法

基于项目的协同过滤推荐算法根据项目评分矩阵计算出项目的最相似邻接项目，并通过邻接项目与目标项目之间的相似度计算得到评分排序最高的前 k 个项目推荐给用户。

项目之间的相似度计算公式如下：

$$sim(i,j)=\frac{\sum\limits_{u\in U_{ij}}(r_{ui}-\overline{r_i})(r_{uj}-\overline{r_j})}{\sqrt{\sum\limits_{u\in U_{ij}}(r_{ui}-\overline{r_i})^2(r_{uj}-\overline{r_j})^2}} \tag{3}$$

其中，$U_{ij}=U_i\cap U_j$ 表示对项目 i 和项目 j 都评分过的用户集合；r_{uj} 表示用户 u 对项目 i 的评分；$\overline{r_j}$ 表示项目 j 的平均评分。

因此，传统的 ICF 推荐可以通过相似度计算出用户 u 对项目 v 的预测评分，如式(4)所示：

$$P_{uv}=\overline{r_v}+\frac{\sum\limits_{i\in V_x}sim(i,v)(r_{ui}-\overline{r_i})}{\sum\limits_{a\in U_x}sim(i,v)} \tag{4}$$

其中，V_x 表示项目 v 的邻居用户集合

2.2　填充技术

在协同过滤中推荐算法中，用户对项目的评分可以表示成一个项目评分矩阵 $\boldsymbol{R}$，其中 $\boldsymbol{R}[u][i]$ 就是用户 u 对物品 i 的评分。由于用户不会对所有物品进行评分，因此这个评分矩阵中有很多空元素，即缺失值，也就是说，用户评分矩阵是非常稀疏的。目前，低质矩阵补全方法成为了预测稀疏评分矩阵缺失项的重要手段。研究者提出了多种求解低质矩阵的有效算法来解决稀疏矩阵填充问题。

2.2.1　NMF 填充技术

NMF（Non-negative Matrix Factorization）是通过计算从原矩阵提取权重和特征两个不同的矩阵出来，如式(5)所示：

$$V_{m\times \boldsymbol{n}}\approx W_{m\times k}\cdot H_{k\times n} \tag{5}$$

其中，$W_{m\times k}\geqslant 0$，$H_{k\times \boldsymbol{n}}\geqslant 0$，$m$ 为特征，$\boldsymbol{n}$ 可以为特征向量。

NMF 将问题描述为：$V\in R_{+}^{m\times n}$，求解 $w\in \boldsymbol{R}_{+}^{m\times k}$，$H\in R_{+}^{k\times n}$，则 NMF 算法求解的目标是 $\min\limits_{W,H}\|X-WH\|$。这里 $+$ 表示矩阵元素需要满足非负约束，W 的列向量被称为基向量，而 H 的列向量被称为表示系数。

如果假设噪声服从高斯分布，则最大化 X 的 log 似然所得到的损失函数如式(6)所示：

$$L(W,H)=\sum_{ij}\frac{1}{2\sigma_{ij}^2}(X_{ij}-(WH)_{ij})^2+\sum_{ij}\log(\sqrt{2\pi}\sigma_{ij}^2) \tag{6}$$

通过梯度下降求出 NMF 的乘性迭代公式为：

$$W_{ij}^{t+1}=W_{ij}^{t}\frac{(XH^{\mathrm{T}})_{ij}}{(XHH^{\mathrm{T}})_{ij}},H_{ij}^{t+1}=H_{ij}^{t}\frac{(W^{\mathrm{T}}X)_{ij}}{(W^{\mathrm{T}}WH)_{ij}} \tag{7}$$

NFM 更符合人类直觉，已经成功应用于文本聚类、人脸识别、基因分析和特征选择中。

传统 NFM 算法通常将一个数据完整矩阵分解为两个非负矩阵乘积，其迭代过程（见式(7)）不适用于矩阵存在大量缺失值的情况。而协同过滤推荐环境下用户项目评分矩阵通常存在大量缺失值，因此不适用于本文应用环境。

2.2.2　SVD 填充技术

SVD 是一种矩阵分解技术，它可以将一个 $m\times n$ 的项目评分矩阵 $\boldsymbol{R}$ 分解为 3 个矩阵的乘积，如式(8)所示：

$$\boldsymbol{R}_{m\times n}=\boldsymbol{U}_{m\times r}\cdot\boldsymbol{S}_{r\times r}\cdot\boldsymbol{V}_{r\times n} \tag{8}$$

其中，$\boldsymbol{U}$ 和 $\boldsymbol{V}$ 是两个大小分别为 $m\times r$，$r\times n$ 的正交矩阵，r 是矩阵 $\boldsymbol{R}$ 的秩，$\boldsymbol{R}$ 为 $r\times r$ 的对角矩阵，$\boldsymbol{R}$ 的所有奇异值作为其对角项。

矩阵 $\boldsymbol{R}_{norm}$ 被定义为：

$$\boldsymbol{R}_{norm}=\boldsymbol{U}_{m\times k}\cdot\boldsymbol{S}_{k\times k}\cdot\boldsymbol{V}_{k\times n} \tag{9}$$

其中，$\boldsymbol{U}_{m\times k}$ 表示用户属性，$\boldsymbol{V}_{k\times n}$ 表示项目属性。通过保留前 k 个奇异值可以降低计算数据的维度，捕捉原始矩阵 $\boldsymbol{R}$ 中存在的不明显且重要的“潜在”关系。

SVD 技术就是通过填充缺失值，解决 CF 算法中因数据稀疏性导致的推荐性能不佳的问题[14-17]。

利用传统的 SVD 进行填充，得到的矩阵，计算任何用户 u 对产品 v 的推荐分数。

$$P_{uv}=\bar{u}+\boldsymbol{U}_k\cdot\sqrt{\boldsymbol{S}_k}'(c)\cdot\sqrt{\boldsymbol{S}_k}\boldsymbol{V}_k'(p) \tag{10}$$

其中，$\boldsymbol{U}_k\cdot\sqrt{\boldsymbol{S}_k}'(c)$ 是 $\boldsymbol{U}_k\ \sqrt{\boldsymbol{S}_k}'$ 的第 c 行，$\sqrt{\boldsymbol{S}_k}\boldsymbol{V}_k'(p)$ 是 $\sqrt{\boldsymbol{S}_k}\boldsymbol{V}_k'$ 的第 p 列的点积，$\bar{u}$ 是用户评分平均值。

传统的 SVD 进行预测的评分准确度不高，假设已知的评分为 R_{uv}，则真实值与预测值的误差为：$e_{uv}=R_{uv}-P_{uv}$，继而可以计算出总的误差平方和：$SSE=\sum\limits_{u,v}e_{uv}^2$。同样地，只要通过局部优化算法把 SSE 降到最小，预测值就能更接近真实值。这里本文采用随机梯度下降法，也就是说如果要最小化

SSE,必须往其负梯度方向搜索。计算完一个 e_{uv} 后,就对预测评分进行更新。

3 基于 SVD 填充矩阵的混合推荐算法

3.1 改进相似度计算

协同过滤算法的核心是根据用户或项目的相似程度找到相应的邻居集合。因此,用户或项目的相似度计算是整个算法的关键所在。

3.1.1 改进用户相似度计算

现有的基于用户的协同过滤推荐技术在查找邻居集合计算相似度时大多没有考虑时间因素。而用户的兴趣爱好会随时间发生变化。比如,小时候喜欢看动画片的用户随着时间的推移会喜欢看爱情片。在计算用户集合时如果没有考虑时间因素,将严重影响推荐的准确性。推荐系统应该给用户推荐与他/她兴趣相似的用户最近喜欢的物品,或是推荐与他/她最近喜欢的物品最相似的物品。因此在本文中提出加入时间权重改进相似度的计算。用户对最近评价或购买过的物品应该有较大的兴趣权重,而对过去评价或购买的物品有较小的时间权重。

结合上述问题,本文用艾宾浩斯遗忘曲线的走势来设计时间权重[18]。权重定义如下:

$$WT_{uv}=\begin{cases}1, & t_{\max}=t_{\min}\\ e^{\frac{t_{uv}-t_{\min}}{t_{\max}-t_{\min}}-1}, & t_{\max}\neq t_{\min}\end{cases} \tag{11}$$

其中,WT_{uv} 表示用户 u 对项目 v 评分的时间权重,式中 t_{uv} 表示用户 u 对项目 v 的评分时间,$t_{\max}$ 表示用户 u 的最近评分时间,$t_{\min}$ 表示用户 u 的最早评分时间。WT_{uv} 是一个单调增函数,取值区间为 $[e^{-1},1]$。

改进后的相似度计算公式为:

$$sim(a,b)=\frac{\sum\limits_{v\in V_{ab}}(WT_{av}\cdot r_{av}-\overline{r_a})(WT_{bv}\cdot r_{bv}-\overline{r_b})}{\sqrt{\sum\limits_{v\in V_{ab}}(WT_{av}\cdot r_{av}-\overline{r_a})^2(WT_{bv}\cdot r_{bv}-\overline{r_b})^2}} \tag{12}$$

3.1.2 改进项目相似度计算

对于现有的基于项目的协同过滤推荐技术,常直接使用式(3)进行相似度计算,存在很多问题,例如项目 1 得到的评分为(-,-,-,-,1,-,-,-,-),项目 2 得到的评分为(-,4,4,-,1,2,-,1,2),项目 3 得到的评分为(-,5,4,-,1,2,-,1,2),通过相似度计算公式得 $sim(1,2)=1$,$sim(2,3)=0.98$,这与实际情况相悖,会直接影响推荐的准确率。对于上述问题,本文引入了 Jaccard 系数:

$$J(i,j)=\left|\frac{U_i\cap U_j}{U_i\cup U_j}\right| \tag{13}$$

其中,$J(i,j)$ 表示项目 i 和项目 j 的共同评分比,$U_i\cap U_j$ 表示共同评价过项目 i 和项目 j 的用户集合,$U_i\cup U_j$ 表示所有评价过项目 i 和项目 j 的用户集合。$J(i,j)$ 的取值区间为 $[0,1]$。

改进后的相似度计算公式为:

$$sim(i,j)=\frac{J(i,j)\sum\limits_{u\in U_{ij}}(r_{ui}-\overline{r_i})(r_{uj}-\overline{r_j})}{\sqrt{\sum\limits_{u\in U_{ij}}(r_{ui}-\overline{r_i})^2(r_{uj}-\overline{r_j})^2}} \tag{14}$$

通过上式可以看出共同评分项占总评分项的比重越大,$J(i,j)$ 则越大;共同评分项占总评分项的比重越小,$J(i,j)$ 则越小。在上述的例子中,项目 2 与项目 3 的共同评分项较多,所以更为相似,与实际情况相符。

3.2 混合推荐算法

一般基于用户的协同过滤推荐算法是根据最相似邻居的评分预测该用户的偏好,对于一个没有邻居集合的新用户来说推荐的准确度较低。但是,在实际应用中,产品需要不断扩展,吸引更多的新用户是商家尤其关注的问题。基于项目的协同过滤推荐算法推荐的是相似的项目,推荐的多样性不足,推荐惊喜度低,用户永远看不到新颖的物品。因此,仅用基于用户或基于物品的协同过滤都会忽略一些信息,从而导致推荐信息不够准确。

综上所述,提出一种融合基于用户和基于物品协同过滤的混合推荐算法,能有效提高推荐的准确率。计算用户 u 对项目 v 的预测评分,如式(15)所示:

$$P_{uv}=\alpha\left(\overline{r_u}+\frac{\sum\limits_{a\in U_x}sim(a,u)(r_{av}-\overline{r_a})}{\sum\limits_{a\in U_x}sim(a,u)}\right)+(1-\alpha)\left(\overline{r_v}+\frac{\sum\limits_{i\in V_x}sim(i,v)(r_{ui}-\overline{r_i})}{\sum\limits_{a\in U_x}sim(i,v)}\right) \tag{15}$$

比如给某个用户推荐一部电影,系统提示是某某与你有相似兴趣的人看了这部电影或是你看过的相似电影,这很难让用户信服,因为用户可能根本不认识那个人,相似的电影也许是很久之前看的。但假如给出的理由是因为这部电影与你有相似兴趣的人看了并且与你最近看过得某部电影相似,这样解释相对合理,用户可能就会接受推荐。

3.3 基于 SVD 填充矩阵的混合推荐模型及算法

在进行推荐时,协同过滤算法存在冷启动和数据稀疏等问题,实际操作中数据的稀疏性使得用户进行相似度计算时存在很大误差,导致推荐效率较低。因此,本文提出了一种基于 SVD 的混合推荐算法(具体见算法 1):首先对项目评分矩阵 $\boldsymbol{R}_{m\times n}$ 使用 SVD 技术得到矩阵 $\boldsymbol{U}$,$\boldsymbol{S}$,$\boldsymbol{V}$,然后对对角矩阵 $\boldsymbol{S}$ 进行降维,得到对应的降维后的用户和项目矩阵 $\boldsymbol{U}_{m\times k}$,$\boldsymbol{V}_{k\times n}$。使用传统填充公式计算填充值,并通过随机梯度下降法优化进而填充稀疏矩阵(见算法 1 第 1-5 行)。然后在填充后的矩阵 $\boldsymbol{R}_{SVD}$ 中,加入时间漂移权重以改进用户相似度计算公式,得到预测评分 P_{uuv};同时加入 Jaccard 相关系数改进项目相似度计算公式,得到预测评分 P_{vuv}(见算法 1 第 6-13 行)。最后利用参数 α 加权融合基于项目的 CF 和基于用户的 CF 得到预测评分 P_{uv},$P_{uv}=\alpha P_{uuv}+(1-\alpha)P_{vuv}$(见算法 15 第 17 行)。这样可以保证一些用户没有相似用户时进行项目相似推荐。同样,没有项目相似时采用相似用户推荐,缓解了冷启动和数据稀疏带来的推荐准确度较低的问题,最终减小了误差,提高了推荐精度。具体描述如算法 1 所示。

算法 1 基于 SVD 的混合推荐算法

输入:用户项目评分数据集

输出:P_{uv}

1. Caculate U,S,V by SVD algorithm
2. if r>6 //保留维度为 6
3. $S_{r\times r}\leftarrow 0$ //对对角矩阵 S 降维
4. end if
5. Caculate $\mathbf{R}_{SVD}$
6. for ∀u,v∈$\mathbf{R}_{SVD}$
7. Caculate $\mathbf{WT}_{uv}$

8. $\mathrm{sim}(a,b)\leftarrow\dfrac{\sum_{v\in \mathbf{V}_{ab}}(\mathbf{WT}_{av}\cdot r_{av}-\overline{r_a})(\mathbf{WT}_{bv}\cdot r_{bv}-\overline{r_b})}{\mathrm{sqrt}[\sum_{v\in \mathbf{V}_{ab}}(\mathbf{WT}_{av}\cdot r_{av}-\overline{r_a})^2(\mathbf{WT}_{bv}\cdot r_{bv}-\overline{r_b})^2]}$

9. Caculate P_{uuv}

10. For $\forall u,v\in \mathbf{R}_{SVD}$ Do

11. Caculate J(i,j)

12. $\mathrm{sim}(i,j)\leftarrow\dfrac{J(i,j)\sum_{u\in U_{ij}}(r_{ui}-\overline{r_i})(r_{uj}-\overline{r_j})}{\mathrm{sqrt}[\sum_{u\in U_{ij}}(r_{ui}-\overline{r_i})^2(r_{uj}-\overline{r_j})^2]}$

13. Caculate P_{vuv}

14. $P_{uv}=\alpha P_{uuv}+(1-\alpha)P_{vuv}$

15. end For

16. end for

17. return P_{uv}

3.4 复杂度分析

算法 1 的时间复杂度主要由 3 部分组成：第一部分是使用 SVD 分解矩阵，体现在算法 1 第 1—5 行，其中第 1 行进行矩阵分解运算，总共需要 $m\times n\times m$ 次乘法运算，其时间复杂度为 $O(m^2n)$，第 5 行利用随机梯度下降法填充稀疏矩阵的时间复杂度低于 $O(m^2n)$；第二部分是引入时间权重和 Jaccard 相关系数优化相似度运算，体现在第 8 行和第 12 行，计算用户与目标用户、项目与目标项目之间的相似性时需要$m\times n$ 次乘法运算，时间复杂度和传统的时间复杂度相同，为 O（mn）；第三部分是计算评分预测矩阵，体现在第 14 行，时间复杂度为为 O（mn）。综上所述，算法 1 的时间复杂度为$O(m^2n)$。

4 实验与评估

4.1 实验数据集

实验采用 MovieLens 和 Jester 两个公开数据集。其中，美国明尼苏达大学 GroupLens 研究项目组所收集到的 MovieLens 数据集[1)]，根据用户对电影的评分向其提供推荐列表，是推荐系统中经典的常用的数据集。该数据集包含 943 个用户对 1682 部电影的评分情况，评分分数在 1～5 之间，评分记录大约有 100000 条。数据稀疏度 φ 为：

$$\varphi=1-\frac{100000}{943\times1682}=93.7\%$$

由此可见，这个评分矩阵是稀疏的。本文选择的实验数据集中用户至少对 20 个电影进行过评分。

Jester 推荐系统数据是从 Jester Online Joke Recommender System 抓取的匿名用户对 Joke 的评分数据。数据集是 2006 年 11 月到 2009 年 5 月间收集的，包含了 24983 个用户对 101 个笑话的评分情况，评分分数在－10～10 之间。其中每个用户至少评价了 36 条笑话。

4.2 衡量标准

选用人们提出的推荐准确度评价标准评测一个推荐系统所推荐的信息是否是用户感兴趣的。用来评估推荐系统准确度的标准多种多样，但其本质都是通过计算预测评分与真实评分的偏差来评估推荐的准确度。预测评分越接近用户的实际评分，推荐系统的准确度越高，相反，预测评分与用户实际评分差别越大，则准确度越低。常用的评估标准中最经典的是平均绝对偏差（Mean Absolute Error，MAE），本文将从这个方面衡量 SVD-HCF 算法的有效性。

$$MAE=\frac{\sum_{u,i\in N}\boldsymbol{R}_{ui}-P_{ui}}{N}\tag{16}$$

其中，R_{ui} 表示用户 u 对项目 i 的真实评分，P_{ui} 表示通过推荐算法预测用户 u 对项目 i 的评分，N 表示协同过滤推荐算法预测的次数，$u,i\in N$ 表示测试集中的每一条用户-项目的评分数据，MAE 的值越小，表明预测分数与实际分数偏差越小，得到的预测数据准确度越高；反之，MAE 的值越大，表明预测分数与实际分数偏差越大，得到的预测数据准确度越低。

4.3 实验结果与分析

4.3.1 敏感度分析

为了得到较佳的实验结果，本文分别在 MoiveLens 数据集和 Jester 数据集上选取不同的填充比例，得到 SVD 填充算法的预测误差变化曲线，如图 4 所示。

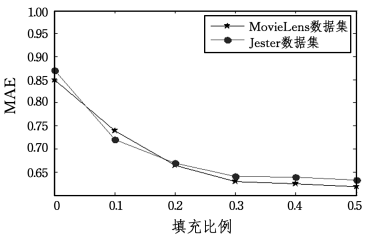

图 1　MAE 的对比结果

从图 1 中可以看出：从 0 开始，以步长 0.1 改变填充比例来计算 MAE 值并进行分析，当填充比例为 0.3 时，算法性能趋向稳定。

然后分析权重系数 α 对本文提出的 SVD-HCF 推荐精度的影响，本文在 MoiveLens 数据集上随机抽取数据集中的 20 个用户，从 0.1 开始，以步长 0.1 改变 α 的取值来计算他们的平均 MAE 并进行分析，然后选择最优参数，如图 2 所示。

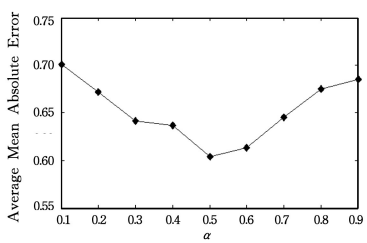

图 2　α 的取值对 MAE 的影响

图 2 中，X 轴为 α 的取值范围，取 0.1～0.9，Y 轴为平均 MAE，由上图可知，$\alpha=0.5$ 时，即在基于 SVD 的混合推荐算法中，ICF 和 UCF 各占一半权重时，平均 MAE 的值最小，SVD-HCF 算法的性能最佳。在接下来的实验中，本文取 $\alpha=0.5$ 进行性能对比分析。

4.3.2 性能对比分析

基于 3.2 节的 MAE 衡量标准，本文将 SVD-HCF 算法与 ICF，UCF，HCF 以及 NMF-CF 算法在不同数据集上进行比较。

图 3 表明不同算法在 MovieLens 数据集上 MAE 的对比结果。图 4 表明不同算法在 Jester 数据集上 MAE 的对比结果。从图中数据可以看出，总体上，UCF，ICF，HCF，NMF-CF 以及 SVD-HCF 5 种算法的 MAE 值随着邻居数目的增大都呈现减小趋势，且差距逐渐缩小。混合推荐的偏差相比基于

1) http://MovieLens.umn.edu

用户的协同过滤推荐算法和基于项目的协同过滤推荐算法有较大改进，NMF-CF 算法有了进一步的改进，而本文提出的基于奇异矩阵分解的混合推荐算法效果更佳。尤其当用户邻居数目为 20 时，平均绝对偏差最小，这是由于本文考虑了时间权重和 Jaccaed 相关系数，优先考虑了最近时间的情况，提高了推荐准确度。用户邻居数目超过 20 的情况下，由图可以看出：随着邻居数目不断的增大，用户的平均绝对偏差也不断增大，这是由于随着数量的增大，需要将与用户相似度较低的邻居加入计算组中，因此，预测分数与实际分数出现较大偏差，推荐精确度降低，平均绝对误差增大。对比实验表明：SVD-HCF 具有较强的适应性和较好的推荐准确性。综上，本文提出的基于 SVD 的混合推荐算法在 *MAE* 衡量标准上与 UCF，ICF，HCF 以及 NMF-CF 算法相比均有较好的表现。

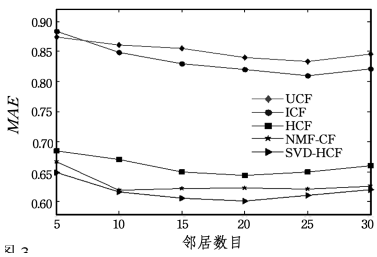

图 3 MovieLens 数据集上各算法的 *MAE* 对比

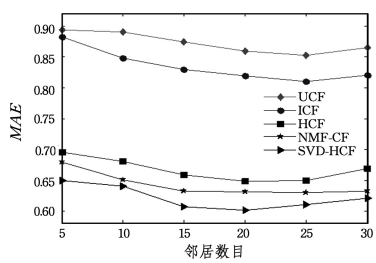

图 4 Jester 数据集上各算法的 *MAE* 对比

结束语 本文针对传统协同过滤存在的数据稀疏、冷启动等问题，提出一种基于 SVD 的混合推荐算法，该算法使用 SVD 分解原始评分矩阵，降低了数据维数，通过随机梯度下降法填充稀疏矩阵。然后利用参数融合基于用户和基于项目的协同过滤算法，并分别加入了时间权重和 Jarcrad 相关系数优化相似度计算，提高了预测分数的准确度，从而提高了推荐的效果。实验结果表明，SVD-HCF 算法可以有效缓解冷启动问题的推荐算法和提高推荐的准确性。然而推荐需要收集用户的大量信息，用户的隐私安全存在泄露风险，下一步的工作拟开展推荐算法的隐私保护研究，在提高推荐质量的同时保护用户的隐私安全。

参 考 文 献

[1] 孟祥武，胡勋，王立才，等. 移动推荐系统及其应用[J]. 软件学报，2013，24(1)：91-108.

[2] JANNACH D，NAVEED S，JUGOVAC M. User control in recommender systems：Overview and Interaction Challenges[C]// International Conference on Electronic Commerce and Web Technologies. 2016：21-33.

[3] RESNICK P，IACOVOU N，SUCHAK M，et al. GroupLens：an open architecture for collaborative filtering of netnews[C]// ACM Conference on Computer Supported Cooperative Work. ACM，1994：175-186.

[4] DAVIDSON J，LIEBALD B，LIU J，et al. The YouTube video recommendation system[C]// ACM Conference on Recommender Systems. ACM，2010：293-296.

[5] 荣辉桂，火生旭，胡春华，等. 基于用户相似度的协同过滤推荐算法[J]. 通信学报，2014，35(2)：16-24.

[6] SARWAR B，KARYPIS G，KONSTAN J，et al. Item-based collaborative filtering recommendation algorithms[C]// International Conference on World Wide Web. ACM，2001：285-295.

[7] DESHPANDE M，KARYPIS G. Item-based top- N recommendation algorithms[J]. ACM International Conference on Transactions on Information Systems，2004，22(1)：143-177.

[8] WU Q，LIN X，HE L. Unifying user-based and item-based algorithm to improve collaborative filtering accuracy[J]. Energy Procedia，2011，13：8231-8239.

[9] WANG B，HUANG J，OU L，et al. A collaborative filtering algorithm fusing user-based，item-based and social networks[C]// IEEE International Conference on Big Data. IEEE，2015：2337-2343.

[10] ZHENG X，LUO Y，et al. Tourism destination recommender system for the cold start problem[J]. KSII Transactions on Internet and Information Systems，2016，10(7)：3192-3212.

[11] KANT S，MAHARA T. Merging user and item based collaborative filtering to alleviate data sparsity[J]. International Journal of System Assurance Engineering and Management，2018，9(1)：173-179.

[12] MA C C. A guide to singular value decomposition for collaboratative filtering[J]. Computer，2009，42(3)：30-37.

[13] KOREN Y，BELL R，VOLINSKY C. Matrix factorization techniques for recommender systems[J]. IEEE Computer，2009，42(1)：30-37.

[14] REDDY M S，ADILAKSHMI T. Music recommendation system based on matrix factorization technique-SVD[C]// International Conference on Computer Communication and Informatics. IEEE，2014：1-6.

[15] WANG J，LI X，WU W，et al. An algorithm of collaborative filtering based on SVD and trust factors[J]. Journal of Chinese Computer Systems，2017，38(6)：1290-1293.

[16] VOZALIS M G，MARGARITIS K G. Applying SVD on item-based filtering[C]// International Conference on Intelligent Systems Design and Applications. IEEE，2005：464-469.

[17] ZHENG X，LUO Y，SUN L，et al. A new recommender systemusing context clustering based on matrix factorization techniques[J]. Chinese Journal of Electronics，2016，25(2)：334-340.

[18] ZHAO F，XIONG Y，LIANG X，et al. Privacy-preserving collaborative filtering based on time-drifting characteristic[J]. Chinese Journal of Electronics，2016，25(1)：20-25.