



引用格式:过金超,杨继纲. 基于改进的 SVD 算法和二分 K - 均值聚类算法的协同过滤算法 [J]. 轻工学报,2020,35(4):88 - 95.

中图分类号:TP301.6 文献标识码:A

DOI:10.12187/2020.04.012

文章编号:2096 - 1553(2020)04 - 0088 - 08

# 基于改进的 SVD 算法和 二分 K - 均值聚类算法的协同过滤算法

## Collaborative filtering algorithm based on the improved SVD algorithm and binary K-means clustering algorithm

过金超,杨继纲

GUO Jinchao, YANG Jigang

**关键词:**

个性化推荐; SVD 算法; 二分 K - 均值聚类算法; 协同过滤; 矩阵分解

**Key words:**

personalized recommendation;  
SVD algorithm;  
binary K-means clustering algorithm;  
collaborative filtering;  
matrix factorization

郑州轻工业大学 电气信息工程学院, 河南 郑州 450002

College of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

**摘要:**针对传统的协同过滤算法存在稀疏性较大和扩展性较差的问题,提出了基于改进的奇异值分解(SVD)算法和二分 K - 均值聚类算法的协同过滤算法. 该算法首先利用改进的 SVD 算法对稀疏的用户 - 项目评分矩阵进行降维,获得用户隐含特征矩阵,然后运用二分 K - 均值聚类算法对相似用户进行聚类来提升算法的可扩展性,最后利用最近邻居集的评分修正目标用户的评分,以减小因矩阵分解导致用户信息丢失造成的误差. 利用 MovieLens 100K 数据集进行的实验结果表明,与传统的基于用户的协同过滤算法、基于 K - 均值聚类的协同过滤算法和隐语义模型(LFM)算法相比,本文提出的算法能够有效提高推荐结果的准确性.

收稿日期:2020 - 04 - 11

基金项目:国家自然科学基金河南省联合项目(U1704149)

作者简介:过金超(1978—),男,河南省开封市人,郑州轻工业大学副教授,博士,主要研究方向为机器人智能控制与信息融合.

**Abstract:** Aiming at the problem of large sparseness and poor scalability of traditional collaborative filtering algorithms, a collaborative filtering algorithm based on the improved singular value decomposition (SVD) algorithm and binary K-means clustering algorithm was proposed. The algorithm firstly used the improved SVD algorithm to reduce the dimensionality of the sparse user-item rating matrix to obtain the user implicit feature matrix, then used the binary K-means clustering algorithm to cluster similar users to improve the scalability of the algorithm, and finally used the nearest neighbor set score to correct the target user's score to make up for the error caused by the loss of user information due to matrix factorization. Experimental results on the MovieLens 100K data set showed that compared with the traditional user-based collaborative filtering algorithm, K-means clustering-based collaborative filtering algorithm and latent factor model (LFM) algorithm, this method could effectively improve the accuracy of recommendation results.

## 0 引言

当前,随着网络技术的迅猛发展,人们可获取的网络信息量日益增多.在这个信息过载的年代,人们对于信息的筛选、分辨变得异常困难,如何更快更高效地获取所需要的信息已成为人们关注的焦点<sup>[1]</sup>.

为缓解信息过载、数据冗杂、选择困难等问题,研究人员将目光投向个性化推荐系统.推荐系统可以根据每个用户的历史行为,分析出用户潜在的喜好,将用户可能感兴趣的内容推荐给用户<sup>[2]</sup>.个性化推荐系统目前已广泛应用于电影、新闻、音乐、电子商务、社交网络等领域.

推荐算法作为推荐系统的核心内容,一直是国内外研究的热点.常见的推荐算法有基于矩阵分解的推荐算法、基于混合算法的推荐算法、基于协同过滤的推荐算法、基于深度学习的推荐算法、基于内容的推荐算法、基于关联规则的推荐算法等<sup>[3-4]</sup>.

协同过滤 CF(collaborative filtering)算法,是运用较早也是推荐效果较好的一种推荐算法.CF算法主要基于用户的历史行为构建用户和项目之间的评分矩阵,根据评分矩阵的相似性向用户做出推荐,一般分为两种情况:一种是基于用户的协同过滤推荐,另一种是基于项目的协同过滤推荐<sup>[5]</sup>.但是随着数据量的不断增大、评分矩阵较稀疏等原因,CF算法的推荐效

果不是很理想.近年来,为了改善数据稀疏性和算法可扩展性,研究人员做了许多研究,例如通过降维技术或聚类技术来改善传统 CF 算法的不足<sup>[6-9]</sup>.H. Koochi 等<sup>[6]</sup>提出了一种基于用户的模糊 C-均值方法,提高了推荐准确率和可扩展性,但没有解决数据稀疏的问题;X. Su 等<sup>[7]</sup>使用了基于用户聚类的协同过滤算法,通过对用户评分矩阵运用 K-均值聚类算法,把相似度较高的用户分到同一个簇中,以减少搜索最近邻的时间,但是由于初始质心是随机选取的,可能导致质心聚集造成推荐质量下降;孙小华等<sup>[8]</sup>在协同过滤中结合奇异值分解 SVD(singular value decomposition)和 K-近邻(KNN)算法改善了数据稀疏性造成的影响,但传统的 SVD 算法对较大数据集的处理效果仍不理想;刘艺等<sup>[9]</sup>采用 BP 神经网络来预测评分,虽然可以改善数据稀疏造成的影响,但需要花费更长的近邻查找时间.

鉴于此,本文拟提出一种基于改进的 SVD 算法和二分 K-均值聚类算法的协同过滤算法,通过改进的 SVD 算法获得用户隐含特征矩阵,然后使用二分 K-均值聚类算法将相似度较高的用户划分到同一个簇中,最后通过最近邻算法预测目标用户对项目的评分,以期降低由数据稀疏性带来的推荐结果误差,提高推荐准确率并提升算法的可扩展性.

# 1 算法设计

本文提出了基于改进的 SVD 算法和二分 K - 均值聚类算法的协同过滤推荐算法. 改进的 SVD 算法对高度稀疏的矩阵拥有较好的推荐效果; 二分 K - 均值聚类算法可以提高查找用户邻居集的效率, 提升算法的可扩展性.

## 1.1 传统 SVD 技术及其改进

SVD 是线性代数中常见的重要矩阵分解技术之一. 矩阵分解通过降低数据的维度提高算法的运行速度, 同时能够得出数据的主要信息, 减少数据特征的数量. 在推荐系统中, 基于 SVD 的矩阵分解算法一直以来都具有良好的性能和表现.

由于 SVD 分解矩阵需要矩阵的全部数据, 所以要先将稀疏的初始评分矩阵  $R$  中的每一个缺失数据都补充上相应的用户评分, B. Sarwar 等<sup>[10]</sup> 提出, 可以将  $R$  做归一化处理, 将评分矩阵的每一个元素都减去所有已知用户评分的平均值, 再对其做 SVD 分解. SVD 的分解公式为

$$R_{n \times r} = U_{n \times s} \Sigma_{s \times s} V_{s \times r}^T$$

其中,  $R$  是一个  $n$  行  $r$  列的矩阵, 左奇异矩阵  $U$  是一个  $n$  行  $s$  列的标准正交矩阵, 右奇异矩阵  $V^T$  是一个  $s$  行  $r$  列的标准正交矩阵,  $\Sigma$  是一个  $s$  行  $s$  列的对角矩阵.  $\Sigma$  矩阵对角元素是从大到小排列的, 被称为初始矩阵  $R$  的奇异值. 在实际应用中, 通常只取前  $m$  个奇异值, 它们保留了矩阵中 90% 的能量信息, 剩余的奇异值置零, 这就意味着在这个数据集中只有  $m$  个重要的数据特征, 其余的特征则是噪声或者是冗余的特征.

由于传统 SVD 技术需要将稀疏矩阵填充为稠密矩阵, 运算量较大, 计算过程较为复杂, 所以本文提出了一种新的矩阵分解方式: 将初始的评分矩阵  $R$  分解成两个小的矩阵  $X$  和  $Y$ ,  $X$  为用户隐含特征矩阵,  $Y$  为物品隐含特征矩阵. 其建模公式为

$$R_{n \times r} = X_{n \times J} Y_{r \times J}^T$$

其中,  $J$  表示隐含特征因子的个数, 矩阵  $X_{n \times J}$  每一行为用户的隐含特征向量, 矩阵  $Y_{r \times J}$  的每一行为项目的隐含特征向量. 利用用户和项目的隐含特征向量可以求解用户  $u$  对还未评分的项目  $i$  可能的预测评分, 其公式为

$$\hat{R}(u, i) = \sum_{j=1}^J X_{uj} Y_{ij} \tag{1}$$

其中,  $\hat{R}(u, i)$  是模型预估的用户  $u$  对项目  $i$  的预测评分,  $X_{uj}$  表示用户  $u$  的第  $j$  个特征值,  $Y_{ij}$  表示物品  $i$  的第  $j$  个特征值.

## 1.2 二分 K - 均值聚类算法

随着推荐系统中用户和项目数量指数式上涨, 导致算法的计算量大增, 查找邻居集所花的时间也越来越长. 所以本文采用聚类技术来缩短查找邻居集所用的时间, 进一步提高推荐的速度和实时性<sup>[7]</sup>. 聚类算法属于一种无监督学习算法, 其基本思路是把一堆物品或者数据按一些特定的规则进行分类, 从而使相似的物品或数据尽可能分在同一类别(簇)中.

K - 均值聚类算法是较流行的聚类算法之一, 其优点是实现难度较低, 聚类效果较好<sup>[11]</sup>. K - 均值聚类算法的基本思想是通过迭代将数据集划分为  $k$  个簇, 使得相同簇中数据之间的相似性较大, 不同簇中数据之间的相似性较小. 但是 K - 均值聚类算法存在收敛于局部最小值的问题, 因此, 研究人员提出了一种优化的算法, 即二分 K - 均值聚类算法. 该算法首先把所有点作为一个簇, 之后把这个簇一分为二, 然后再选择其中一个簇进行 K - 均值聚类 ( $k = 2$ ), 选取簇的标准是对该簇进行划分是否能最大可能地降低误差平方和的值, 误差平方和是一种用来度量聚类效果的指标, 其值越小说明数据点离所属簇的质心越近, 聚类效果越好. 通过这种方法不断划分, 直到最终得到的簇的数目与用户最开始给定的  $k$  值相等. 二分 K - 均值

聚类算法与K-均值聚类算法相比,不但受初始质心的影响更小,而且聚类速度更快,效果更好.

### 1.3 本文算法流程

**1.3.1 降维处理** 首先通过改进的SVD算法对稀疏的用户-项目评分矩阵进行降维,得到用户隐含特征矩阵.由式①可以得到改进的SVD的损失函数:

$$Loss = \sum_{(u,i) \in D} (R(u,i) - \hat{R}(u,i))^2 \quad (2)$$

其中, $D$ 为训练集中所有用户对项目的评分集合, $R(u,i)$ 是用户 $u$ 对项目 $i$ 的实际评分, $\hat{R}(u,i)$ 是模型预估的用户 $u$ 对项目 $i$ 的预测评分.为了防止过拟合,对式②进行 $L_2$ 正则化:

$$Loss = \sum_{(u,i) \in D} (R(u,i) - \sum_{j=1}^J X_{uj}Y_{ij})^2 + \alpha \|X_u\|^2 + \alpha \|Y_i\|^2 \quad (3)$$

其中, $\alpha$ 为正则化参数.

对式③采用随机梯度下降法进行优化,分别对参数 $X_{uj}$ 和 $Y_{ij}$ 求偏导:

$$\frac{\partial Loss}{\partial X_{uj}} = -2(R(u,i) - \hat{R}(u,i))Y_{ij} + 2\alpha X_{uj}$$

$$\frac{\partial Loss}{\partial Y_{ij}} = -2(R(u,i) - \hat{R}(u,i))X_{uj} + 2\alpha Y_{ij}$$

通过梯度下降得到 $X_{uj}$ 和 $Y_{ij}$ 的第 $n+1$ 次迭代结果:

$$X_{uj}^{(n+1)} = X_{uj}^{(n)} - \beta \frac{\partial Loss}{\partial X_{uj}^{(n)}}$$

$$Y_{ij}^{(n+1)} = Y_{ij}^{(n)} - \beta \frac{\partial Loss}{\partial Y_{ij}^{(n)}}$$

其中, $\beta$ 代表的是学习率.

在式③中,损失函数Loss中加入了正则化项,用于控制模型的复杂度.参数 $\alpha$ 的大小决定了正则化项对于模型复杂度的约束能力, $\alpha$ 过大可能导致约束力太大,造成一些重要参数的缺失,影响模型的精度; $\alpha$ 过小则会导致约束力无法减小模型的复杂度,造成过拟合.而参数 $\beta$ 则会影响损失函数找到极值点的概率, $\beta$ 越小,

越容易找到极值点,但也会需要更多的迭代次数.因此,选取合适的 $\alpha$ 和 $\beta$ 可以更好地提升模型的精度.

**1.3.2 聚类** 通过改进的SVD算法得到用户隐含特征矩阵之后,再通过二分K-均值聚类算法对矩阵进行聚类,得到相似用户的簇和质心.用户到质心的距离通过Pearson相关系数隐含,Pearson相关系数的计算公式为

$$P(A,B) = \frac{\sum_{f=1}^F (A_f - \bar{A})(B_f - \bar{B})}{\sqrt{\sum_{f=1}^F (A_f - \bar{A})^2} \times \sqrt{\sum_{f=1}^F (B_f - \bar{B})^2}}$$

其中, $A_f$ 表示用户 $A$ 对项目 $f$ 的评分, $B_f$ 表示用户 $B$ 对项目 $f$ 的评分, $\bar{A}$ 和 $\bar{B}$ 分别表示用户 $A$ 和用户 $B$ 对项目打分的平均值, $P(A,B)$ 即为所求Pearson相关系数的相似度.

**1.3.3 预测评分** 根据Pearson相关系数计算出用户与同簇中其他用户的相似度,再根据相似度的大小,选取与用户相似度最高的 $N_u$ 个用户作为用户的最近邻居集.通过基于最近邻居集的预测方法来计算目标用户 $u$ 对尚未评分项目 $i$ 的预测评分.具体评分公式如下:

$$\hat{R}(u,i) = \frac{\sum_j X_{uj}Y_{ij} + \sum_{v \in N_u} P(u,v)(R(v,i) - \sum_j X_{vj}Y_{ij})}{\sum_{v \in N_u} |P(u,v)|}$$

其中, $P(u,v)$ 代表的是用户 $u$ 和用户 $v$ 之间的相似性, $R(v,i)$ 是用户 $v$ 对项目 $i$ 的实际评分.

最后根据预测的结果进行Top-N推荐,生成推荐列表.本文算法流程如图1所示.

## 2 实验结果与分析

### 2.1 实验数据集与评估指标

本次实验采用的数据集是由美国明尼苏达大学GroupLens项目组的研究人员所收集的

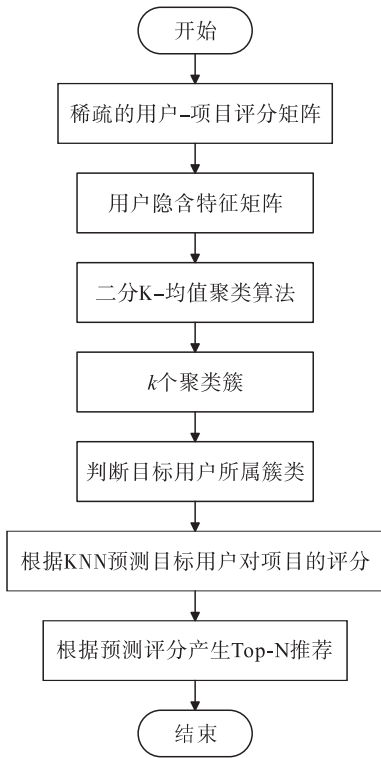


图1 本文算法流程图

Fig. 1 Algorithm flow chart

MovieLens 100K 数据集<sup>[12]</sup>. 该数据集包含 943 名用户对 1682 部电影的评分数据, 共计 100 000 条, 其中每一个用户都至少对其中 20 部以上的电影做出了评分, 数据集的稀疏性约为 93.695 3%. 用户的评分为 1—5 之间的整数, 评分越高代表用户对电影的喜爱程度越高.

本实验分别采用平均绝对误差和准确率作为预测精度和分类误差的衡量标准. 在推荐系统中, 定义  $N$  为测试集中所有用户对项目做出评分的集合, 计算出每一个实际评分与推荐算法预测出的评分之间的绝对误差, 最后求出其平均值, 得到平均绝对误差, 计算公式为

$$MAE = \frac{\sum_{(u,i) \in N} |R(u,i) - \hat{R}(u,i)|}{|N|}$$

其中,  $|N|$  为测试集的数据大小.  $MAE$  越小, 表示模型预测值与真实值之间的差值越小, 表明推荐算法的准确度越高.

准确率可以直接反映推荐列表对于用户来

说是否满意, 如果推荐列表的准确率较低, 说明在这个推荐列表中用户喜欢的项目较少, 用户体验较差. 准确率计算公式为

$$Precision = \frac{\sum_{u \in U} |P(u) \cap T(u)|}{\sum_{u \in U} |P(u)|}$$

其中,  $P(u)$  代表的是训练集中根据用户的行为做出的推荐列表, 而  $T(u)$  则是测试集中用户的行为列表.

### 2.2 参数选取

本文提出的基于改进的 SVD 算法和二分 K - 均值聚类算法的协同过滤算法中的主要参数包括: 隐含特征个数  $J$ 、正则化参数  $\alpha$ 、聚类簇数  $k$ 、学习速率  $\beta$ 、迭代次数  $M$  和最近邻个数  $N_u$ . 由于学习速率  $\beta$  和迭代次数  $M$  不会对推荐算法的质量产生太大的影响, 根据前期实验和经验, 本文选取  $M = 80, \beta = 0.015$ <sup>[13]</sup>. 根据算法准确度评估指标  $MAE$  的变化来调整本文算法的其他参数.

**2.2.1 隐含特征个数** 隐含特征个数  $J$  的大小会影响矩阵分解过程中保留特征信息的多少.  $J$  越大, 可以保留更多的原始矩阵信息, 但也会造成模型的时间和空间复杂度过高;  $J$  越小, 会造成原始矩阵信息丢失过多, 增大模型最终预测推荐结果的误差. 为了确定本文算法中  $J$  的最优取值, 在  $k = 10, N_u = 25, M = 80, \beta = 0.015, \alpha = 0.01$  的条件下进行实验, 结果如图 2 所示.

由图 2 可知, 随着  $J$  不断增大, 原始评分矩阵中被保留下来的隐含信息逐渐增多, 最终推荐结果的准确度也越高, 从  $J = 60$  开始, 本文算法的推荐准确度逐渐趋于稳定.

**2.2.2 正则化参数** 正则化参数  $\alpha$  的大小决定了正则化项对于模型复杂度的约束能力. 为了确定本文算法中  $\alpha$  的最优取值, 在  $k = 10, N_u = 25, M = 80, \beta = 0.015, J = 60$  的条件下进行实验, 结果如图 3 所示.

由图3可知,当正则化参数 $\alpha = 0.02$ 时,改进算法的MAE达到最小值,当 $\alpha$ 继续上升时,由于参数太大使得算法出现欠拟合的现象,导致算法的预测能力越来越差。

**2.2.3 聚类簇数** 为了确定本文算法中聚类簇数 $k$ 的最优取值,在 $N_u = 25, M = 80, \beta = 0.015, J = 60, \alpha = 0.02$ 的条件下进行实验,结果如图4所示。

由图4可知,本文算法的MAE值随着聚类簇数 $k$ 的不断增大先减小后增大,当 $k = 16$ 时,MAE达到最小值。

**2.2.4 最近邻个数** 最近邻个数 $N_u$ 表示选取与目标用户相似度最高的 $N_u$ 个用户,如果选取的相似用户太多,则会消耗大量的时间,影响

推荐的实时性;如果选取的相似用户太少,则会使得最近邻居集对最终预测评分的影响较小。为了确定本文算法中 $N_u$ 的最优取值,在 $\alpha = 0.02, k = 16, M = 80, \beta = 0.015, J = 60$ 的条件下进行实验,结果如图5所示。

由图5可知,本文算法的MAE值随着最近邻个数 $N_u$ 的不断增大而减小,矩阵分解过程中用户特征信息的损失问题得到一定改善。当 $N_u = 40$ 时趋于平稳,之后继续增加对预测评分产生的影响较小。

综上,本文算法的最优参数确定为 $J = 60, \alpha = 0.02, k = 10, \beta = 0.015, N_u = 40, M = 80$ 。

**2.3 实验结果**

为了验证本文算法在实际应用中的推荐效果,将本文算法与基于用户的协同过滤算法、基于K-均值聚类的协同过滤算法、隐语义模型

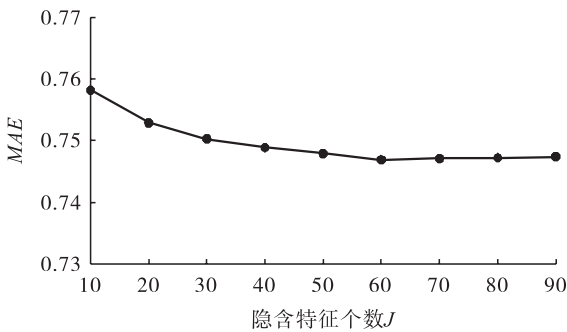


图2 本文算法在不同隐含特征个数 J 下的 MAE

Fig. 2 MAE of improved algorithm under different number of hidden features J

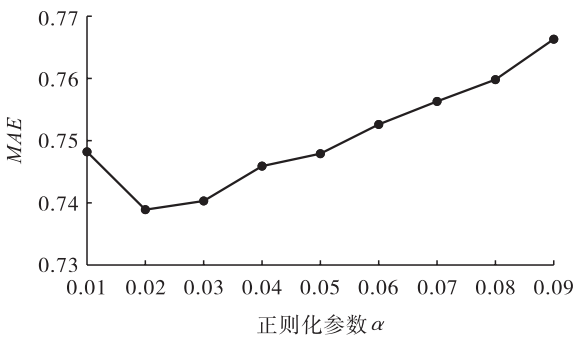


图3 本文算法在不同正则化参数 alpha 下的 MAE

Fig. 3 MAE of improved algorithm under different regularization parameters alpha

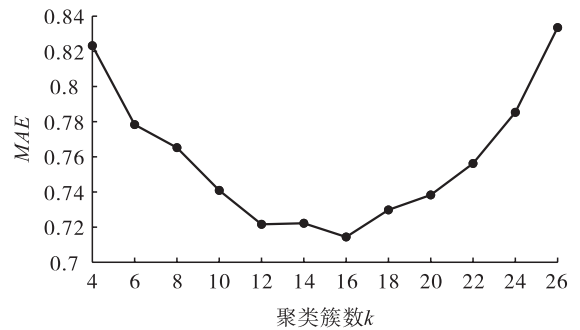


图4 本文算法在不同聚类簇数 k 下的 MAE

Fig. 4 MAE of improved algorithm under different clustering number k

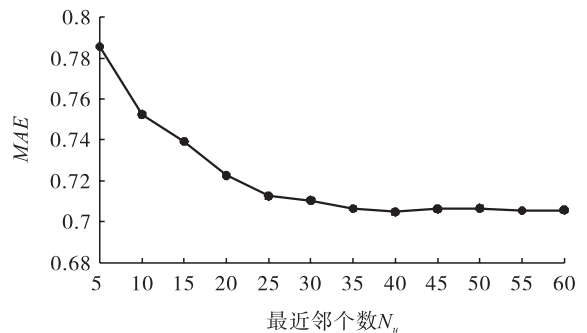


图5 本文算法在不同最近邻个数 Nu 下的 MAE

Fig. 5 MAE of improved algorithm under different numbers of nearest neighbors Nu

(LFM)算法进行对比实验,结果如图6和图7所示.

由图6可知,本文算法的MAE始终低于基于用户的协同过滤算法和基于K-均值聚类的协同过滤算法. LFM算法的推荐准确度与最近邻个数 $N_u$ 无关,不会随着 $N_u$ 的变化而变化. 在 $N_u < 20$ 时,LFM算法的推荐效果最好,但是随着 $N_u$ 的不断增加,本文算法逐渐优于LFM算法,当 $N_u = 40$ 时,LFM算法的 $MAE = 0.7189$ ,本文算法的 $MAE = 0.7048$ ,推荐效果提升了约1.4%.

由图7可知,当最近邻个数 $N_u < 25$ 时,本文算法的准确率低于LFM算法,但高于基于K-均值聚类的协同过滤算法和基于用户的协同过滤算法;当 $N_u > 25$ 时,本文算法的准确率

明显优于其他3种算法,并且在 $N_u = 50$ 时趋于稳定,说明本文算法具有良好的推荐效果.

### 3 结语

本文将矩阵分解与聚类技术相结合,提出了基于改进的SVD算法和二分K-均值聚类算法的协同过滤算法,该算法使用改进的SVD算法得到用户隐含特征矩阵,改善了数据集稀疏导致推荐结果误差较大的问题,并且通过二分K-均值聚类算法提升了推荐算法的可扩展性,通过实验对比证实了本文算法的推荐性能优于基于用户的协同过滤算法、基于K-均值聚类的协同过滤算法和LFM算法. 用户的显性行为数据虽然能够直接用于推荐算法的计算,但是这些数据只是实际应用场景数据中的一小部分,用户的点击次数、浏览记录等隐性行为数据也会对推荐算法的准确度提升产生至关重要的影响,后续将进一步研究如何将显性行为数据和隐性行为数据相结合以提升推荐算法的准确率.

### 参考文献:

[1] SUN P, LI Z, HAN Z, et al. An overview of collaborative filtering recommendation algorithm [J]. Advanced Materials Research, 2013, 756 - 759: 3899.

[2] 张琳, 闫强. 基于管理和消费者行为视角的个性化推荐研究与展望[J]. 北京邮电大学学报(社会科学版), 2016, 18(6): 24.

[3] BRUSILOVSKY P, KOBSA A, NEJDL W. The adaptive web: Methods and strategies of web personalization [J]. Lecture Notes in Computer Science, 2007, 2002(5): 377.

[4] LI B, ZHU X Q, LI R J, et al. Cross-domain collaborative filtering over time [C] // Proceedings of the 22nd International Joint Conference on Artificial Intelligence. Palo Alto: AAAI Press,

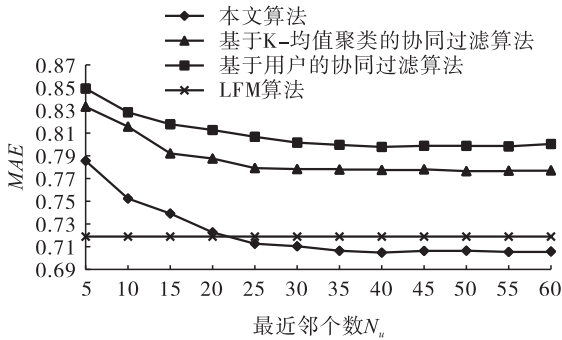


图6 各个算法在不同最近邻个数 $N_u$ 下的MAE

Fig. 6 MAE of each algorithm under different numbers of nearest neighbors  $N_u$

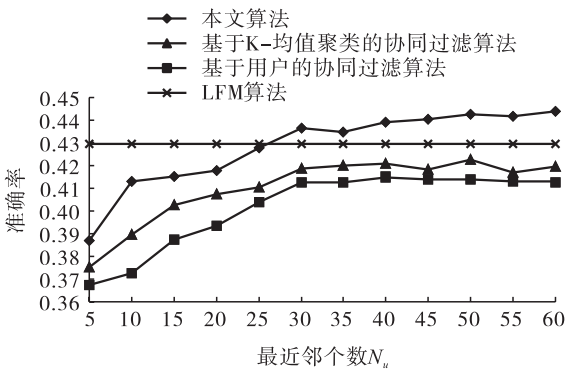


图7 各个算法在不同最近邻个数 $N_u$ 下的准确率

Fig. 7 Precision of each algorithm under different numbers of nearest neighbors  $N_u$

- 2011;2293.
- [5] SUGANESHWAR G, IBRAHIM S P. A survey on collaborative filtering based recommendation system[C] // 3rd International Symposium on Big Data and Cloud Computing. Chengdu: Springer-Verlag, 2016:503.
- [6] KOOHI H, KIANI K. User based collaborative filtering using fuzzy C-means[J]. Measurement, 2016, 91(1):134.
- [7] SU X, KHOSHGOFTAAR T M. Collaborative filtering for multi-class data using belief nets algorithms[J]. International Journal on Artificial Intelligence Tools, 2008, 17(1):71.
- [8] 孙小华, 陈洪, 孔繁胜. 在协同过滤中结合奇异值分解与最近邻方法[J]. 计算机应用研究, 2006, 23(9):206.
- [9] 刘艺, 冯钧, 魏童童, 等. 一种改进的协同过滤推荐算法[J]. 计算机与现代化, 2017(1):1.
- [10] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms[C] // Proceedings of the 10th International Conference on World Wide Web. New York: ACM, 2001:285.
- [11] PANDEY A, PANDEY R. Elective recommendation support through K-means clustering using R-tool[C] // International Conference on Computational Intelligence and Communication Networks. Kolkata: ICRICIN, 2016:851.
- [12] HARPER F M, KONSTAN J A. The MovieLens datasets: History and context[J]. ACM Transactions on Interactive Intelligent Systems, 2015, 5(4):1.
- [13] 陈清洁. 基于SVD的协同过滤推荐算法研究[D]. 成都: 西南交通大学, 2017.
- 
- (上接第87页)
- [15] 李大勇, 王瑜, 黎灿兵, 等. 基于无线射频技术的防窃电开箱记录仪设计[J]. 电测与仪表, 2008, 45(10):51.
- [16] 孙凤杰, 刘争芳, 张永灿. 基于GPRS无线传输的防窃电系统[J]. 电力系统通信, 2007, 28(171):53.
- [17] 余昌华, 谢剑英. Winsocket技术在电力远程监控系统中的应用[J]. 计算机工程, 2000, 26(10):81.
- [18] 窦健, 陈秀群, 张海龙, 等. 一种具有约束条件的用电异常检测模型: 201711154836. 7[P]. 2018-05-22.
- [19] 吴迪, 王学伟, 窦健, 等. 基于大数据的防窃电模型与方法[J]. 北京化工大学学报(自然科学版), 2018, 45(6):79.
- [20] 庄池杰, 张斌, 胡军, 等. 基于无监督学习的电力用户异常用电模式检测[J]. 中国电机工程学报, 2016, 36(2):379.
- [21] 程超, 张汉敬, 景志敏, 等. 基于离群点算法和用电信息采集系统的反窃电研究[J]. 电力系统保护与控制, 2015, 43(17):69.
- [22] 王新霞, 王珂, 焦东翔, 等. 基于正态分布离群点算法的反窃电研究[J]. 电气应用, 2017, 36(7):60.
- [23] 窦健, 刘宣, 卢继哲, 等. 基于用电信息采集大数据的防窃电方法研究[J]. 电测与仪表, 2018, 55(21):43.
- [24] 任玮蒙, 许庆, 谢智奕, 等. 基于电量、电压和电流异常分析的异常用电判断方法: 201410706073. 2[P]. 2015-03-11.