第 36 卷 第 6 期
2021 年 12 月

天津科技大学学报
Journal of Tianjin University of Science & Technology

Vol. 36 No. 6
Dec. 2021

DOI:10.13364/j.issn.1672-6510.20210135

基于 K-means 的改进协同过滤算法

吴婷婷，李孝忠，刘徐洲
（天津科技大学人工智能学院，天津 300457）

摘　要：协同过滤算法在推荐系统中得到了广泛的应用，但是随着数据的不断增长，用户相似度低、推荐准确性不高等问题也逐渐显现. 针对上述问题，提出一种基于 K-means 的改进协同过滤算法. 首先，通过 K-means 聚类算法将相似的用户进行聚类，在聚类过程中，利用欧几里得公式计算数据之间的距离，该算法得到聚类效果最好的簇数 K；其次，将 K 值作为二分 K-means 算法的输入，通过该聚类算法得到最终的聚类结果；再次，通过改进之后的相似度公式得到目标用户的邻居用户集合；最后，通过预测评分公式预测项目的分值. 实验表明，该算法在准确率、召回率以及 F_1 指标上都有一定程度的提高.

关键词：协同过滤算法；K-means 算法；二分 K-means 算法
中图分类号：TP393　　**文献标志码**：A　　**文章编号**：1672-6510(2021)06-0044-05

Improved Collaborative Filtering Algorithm Based on K-means

WU Tingting，LI Xiaozhong，LIU Xuzhou
(College of Artificial Intelligence, Tianjin University of Science & Technology, Tianjin 300457, China)

Abstract: Collaborative filtering algorithm has been widely used in recommendation systems. However, with the continuous growth of data, such problems as low user similarity and low recommendation accuracy have gradually emerged. To address these problems, an improved collaborative filtering algorithm based on K-means is proposed in this article. First, similar users are clustered by K-means clustering algorithm, using the Euclidean formula for the best clustering number K. Second, the K value is used as the input of the dichotomous K-means algorithm to obtain the final clustering result. Third, the neighbor user set of the target user is obtained by improved similarity formula. Finally, the project score is predicted by the prediction score formula. Experiments show that the proposed algorithm achieves some improvement in precision accuracy, recall and F_1 metrics.

Key words: collaborative filtering algorithm; K-means algorithm; dichotomous K-means algorithm

互联网上信息资源的爆炸式增长给用户带来了信息过载问题，不明确的用户需求更是对搜索引擎提出了更大的挑战[1−2]. 针对这一问题，推荐系统作为一种高效便捷的自动化筛选信息工具受到广泛关注[3]. 已有的推荐系统可分为基于内容的推荐、基于关联规则的推荐、协同过滤推荐、混合推荐[4]. 在目前的推荐系统中，基于协同过滤的推荐算法使用范围最广[5]，具有较高的研究和发展价值. 常用的协同过滤推荐算法主要有两类[6−7]：(1)基于用户的协同过滤(user-based collaborative filtering，User-based CF)算法，通过用户的历史行为判断用户对项目的喜爱程度，根据不同的用户对同一项目的偏好计算用户之间的关系，并在具有相同偏好的用户之间进行项目的推荐；(2)基于物品的协同过滤(item-based collaborative filtering，Item-based CF)算法，根据计算不同用户对不同项目的喜爱获得项目之间的关系. 但是在使用协同过滤推荐算法中存在着数据稀疏、用户相似度不高等问题，导致推荐结果的准确度较低. Kanimozhi[8]

收稿日期：2021−06−02；**修回日期**：2021−07−16
基金项目：天津市自然科学基金资助项目(18JCQNJC69500)
作者简介：吴婷婷（1997—），女，河北人，硕士研究生；**通信作者**：李孝忠，教授，lixz@tust.edu.cn

通过对商品进行聚类，较大程度上缩小了商品的最近邻居搜索范围，提高了算法的运行效率. 但是，该算法并没有利用用户对商品的评级信息，忽视了用户的历史行为记录，从而导致了推荐结果准确性的提高存在一定程度的限制. Tsaic 等[9]将协同过滤算法与聚类算法相结合，有效地提高了推荐系统的准确度，但是在计算用户的相似度时仅使用了皮尔森相似度公式，存在着用户相似度可能不高的问题. 李顺勇等[10]将 K-means 聚类算法运用到协同过滤算法中，并利用改进的相似度公式寻找用户的邻居集合，在一定程度上提高了推荐结果的准确性，但是在运用相似度公式时未考虑用户共同评级的影响. 岳希等[11]先用基于用户的协同过滤算法对用户未评价的项目进行预测，然后将预测分数对原始用户–项目评分矩阵进行填充，从而缓解了数据的稀疏性. 但是第一次运用协同过滤算法时采用皮尔森相似度进行计算，导致预测评分存在一定的偏差，影响原数据的准确性以及后续的推荐结果. Feng 等[12]在计算用户相似度时考虑多种因素的影响，提高了用户之间相似度性. 但是计算较为复杂，计算量较大，影响算法的整体效率. Koohi 等[13]将模糊聚类融合到协同过滤算法中，得到最佳的推荐结果，但是在数据量较大时，要获得确定的聚类结论可能有一定的困难，影响用户之间的相似度.

本文主要针对基于用户的协同过滤算法中的用户相似度计算部分以及 K-means 聚类算法进行改进. 相似度公式主要考虑用户之间共同评分的项目的数量以及不同用户针对同一项目评分标准之间的差异这两个影响因素，进而提高用户之间的相似度；聚类过程中主要在 K-means 聚类算法的基础上引入了二分 K-means 聚类算法，从而提高聚类结果以及推荐结果的准确性.

1　基础工作

1.1　基于用户的协同过滤算法

基于用户的协同过滤（User-based CF）算法于 1992 年提出并在邮件过滤系统中应用成功，1994 年被研究机构 GroupLens 在新闻过滤中成功使用，直到 2000 年成为了推荐系统领域中应用最广泛的算法之一. 该算法收集用户偏好的数据后，使用 KNN 算法计算用户的最近聚类，从而得出用户的共同偏好，最终根据共同偏好程度向用户推荐共同偏好.

在 User-based CF 算法中，对目标用户进行推荐需要利用用户–项目评分矩阵搜索与目标用户相似的邻居用户，利用邻居用户产生预测评分. 算法主要有 3 个步骤：相似度计算、邻居用户的选择和评分预测[14]. User-based CF 算法的流程图如图 1 所示.

相似度计算过程是算法的核心部分，常见的相似度计算方法公式有皮尔森（Pearson）相似度、杰卡德（Jaccard）相似度等，这些相似度计算方法是将一个用户对所评价项目的分值作为一个向量计算用户之间的相似度. 随着数据信息的快速增长，用户和项目数量都在急剧增加，这种情况导致了用户–项目评分矩阵非常稀疏，上述传统相似度计算方法的效果并不理想. 因此，近几年来有很多针对相似度计算的研究.

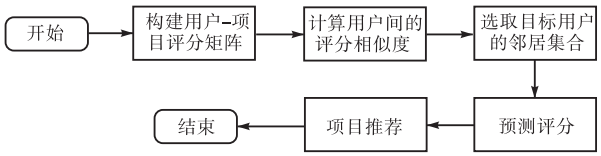

图 1　User-based CF 算法的流程图

Fig. 1　Flowchart of the User-based CF algorithm

1.2　K-means 算法

聚类是以相似性为基础的，也就是将相似的东西划分为一组，在聚类的过程中并不知道某一类是什么，而最终的目标仅仅是把相似的东西聚到一起. 聚类一般为数值聚类，对数据提取 M 种特征并组成一个 M 维向量，进而得到一个从原始数据到 M 维向量空间的映射，然后基于某种方法或者规则进行划分，使得同组的数据具有最大的相似性. 其中，K-means 算法是聚类算法中应用最广泛、最简洁的一种算法.

K-means 算法是基于距离的聚类算法中的一种[15]. 该算法的目标是根据输入的参数 K（聚类目标数），将所有的数据划分为 K 个类. 基本思想是将每个数据点归为距离它最近的聚类中心点所在的簇类. 其具体步骤如下：

（1）随机在数据点中选取 K 个数据点作为初始聚类中心点.

（2）对于每一个数据点，利用欧几里得公式计算该点到每一个聚类中心的距离并将其划分到最近的类. 其中，欧几里得公式为

$$E(x,y)=\sqrt{\sum_{i=1}^{n}(r_{u_i}-r_{v_i})^2} \tag{1}$$

其中：x、y 为用户 u、v 评价项目的分值；r_{u_i}、r_{v_i} 表示用户 u、v 对项目 i 的评分；n 为项目个数.

（3）利用公式（2）重新计算每一个类别中数据点的平均值，并将得到的平均值点作为新的聚类中心

点. 若没有数据点和平均值点相同，利用公式(1)计算数据点到平均值点的距离，将距离平均值点最近的数据点作为新的聚类中心点.

$$\boldsymbol{c}_i = \frac{1}{m_i}\sum_{x\in C_i}\boldsymbol{x} \tag{2}$$

其中：C_i 表示第 i 个类别，$\boldsymbol{c}_i$ 代表第 i 类数据的平均值点，$\boldsymbol{x}$ 是第 i 个类别中的数据点，m_i 为第 i 个类别中数据点的个数.

(4) 如果聚类中心的变化没有超出预设的阈值，则收敛；否则转到步骤(2).

K-means 聚类算法简单，计算速度快，同时在处理大量数据时具有相对可伸缩性.

1.3　二分 K-means 算法

当质心相对较远时，K-means 聚类算法不能很好地在簇与簇之间重新计算分布质点，因而聚类结果不佳[16]. 为了改善这一问题，采用二分 K-means 算法，该算法属于分层聚类的一种. 通常分层聚类有两种策略：聚合，一种自底向上的办法，将每一个观察者初始化本身为一类，然后进行两两相结合；分裂，一种自顶向下，将所有的观察者都初始化为一类，然后进行递归分裂.

二分 K-means 算法为了得到 K 个簇，将所有数据作为一个簇集合并分裂成两个簇，直到划分为 K 个簇. 该算法的具体步骤如下：

(1) 将所有数据点当成一个簇.

(2) 对每一个簇计算簇内误差平方和(sum of squared error，SSE). SSE 表示某一簇内其他数据点到聚类中心的距离总和，该值越小，说明该簇越紧凑，聚类效果最好. 在给定的簇上进行 K-means 聚类($K = 2$)，并计算将该簇分裂后的总 SSE. 其中，SSE 的计算公式为

$$\mathrm{SSE} = \sum_{i=1}^{k}\sum_{x\in C_i}\mathrm{dist}(\boldsymbol{c}_i, \boldsymbol{x})^2 \tag{3}$$

其中：$\boldsymbol{c}_i$ 表示质心，$\boldsymbol{x}$ 表示以 $\boldsymbol{c}_i$ 为质心的簇内的数据，dist 表示两个向量的欧几里得距离.

(3) 选择可以最大程度降低 SSE 值的簇进行划分.

(4) 重复步骤(2)和(3)，直到达到指定的簇数时停止.

因为二分 K-means 算法在计算相似度的数量少，所以该算法可以加速 K-means 算法的执行速度，同时能够克服 K-means 算法收敛于局部最小的缺点.

2　本文改进的协同过滤算法

2.1　相似度的改进

传统的相似度计算方法在计算用户之间的相似度时都存在着各自的缺陷. 比如，Jaccard 相似度在计算时仅考虑了用户之间共同的评分项目的数量，Pearson 相似度在计算过程中仅考虑了用户之间不同评分标准的差异性. 针对传统的相似度计算方法存在的缺陷进行如下改进：

(1) 考虑用户共同评分项目的数量对相似度的影响. 在计算相似度过程时，会发现两个用户共同评分项目的数量越多，两者相似度就会越高.

(2) 考虑用户评分标准的差异对相似度的影响. 用户对同一个项目的评分标准又是不同的. 比如，用户 1 不喜欢项目 1，对其评为 3 分；用户 2 同样不喜欢项目 1，对其评为 1 分；而用户 3 喜欢项目 1，对其评为 3 分.

在计算相似度时，将 Jaccard 相似度和 Pearson 相似度进行结合，同时考虑了用户共同评分项目的数量和不同用户评分标准的差异性对相似度的影响，最终的相似度公式为

$$\mathrm{sim}(u,v) = \mathrm{sim}_{\mathrm{Jaccard}}(u,v)\times\mathrm{sim}_{\mathrm{Pearson}}(u,v) \tag{4}$$

2.2　本文算法

K-means 算法与协同过滤算法相组合的算法对传统的协同过滤算法做出了进一步的改进，使得传统推荐算法的准确率有了一定程度的提高. 比如，文献[7]将层次聚类算法与协同过滤算法进行融合，减少了用户搜索邻居的范围，提高推荐速度，有效地提高了推荐结果的准确性，但该方法计算的复杂度较高. 文献[17]将密度聚类算法与协同过滤算法进行结合，进一步提高推荐结果的准确性，但是该方法对于圈的半径以及阈值这两个参数的设置较为敏感.

考虑上述单一的聚类算法与协同过滤算法进行结合时出现的问题，将 K-means 算法和二分 K-means 算法进行结合对协同过滤算法进行改进. 首先将数据集转为用户–项目评分矩阵(缺失值用 0 补充)；然后在通过 K-means 聚类算法进行第一次聚类，根据轮廓系数得到聚类效果最好的 K 值；接着将 K 值作为二分 K-means 聚类算法的输入，再根据二分 K-means 算法进行第二次聚类，得到最终的聚类结果；最后根据目标用户的相似邻居已评分而目标用户未评分的项目进行预测评分. 轮廓系数($S(i)$)、预测评

分（$\hat{r}_{u_i}$）按照式（5）和式（6）计算.

$$S(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \tag{5}$$

其中：$b(i)$ 表示样本点 i 距离下一个最近簇中所有点的平均距离，$a(i)$ 表示样本点 i 与同一簇内其他点的平均距离. $S(i)$ 值越大，聚类效果越好.

$$\hat{r}_{u_i}=\overline{r}_u+\frac{\sum_{v\in N_m}\mathrm{sim}(u,v)(r_{v_i}-\overline{r}_v)}{\sum_{v\in N_m}|\mathrm{sim}(u,v)|} \tag{6}$$

其中：$\hat{r}_{u_i}$ 是用户 u 对项目 i 的预测评分，$\overline{r}$ 是用户 u 在所有项目上的平均得分，N_m 表示与用户 u 最为相近的前 m 个用户.

改进之后的算法流程如图 2 所示，具体步骤如下：

（1）将数据进行处理并转为用户–项目评分矩阵，缺失值用 0 填充.

（2）将用户–项目评分矩阵作为 K-means 聚类算法的输入，根据式（5）得到聚类效果最好的分类数目 K. 其中，利用式（1）计算样本点与中心点的距离.

（3）将用户–项目评分矩阵和聚类数目 K 输入到二分 K-means 算法中，得到每位用户所属的簇.

（4）计算目标用户所属的簇以及该簇内最为相似的前 m 个相似用户，利用式（4）作为相似度计算公式.

（5）根据邻居用户集合，利用式（6）对目标用户未评分的项目进行预测评分.

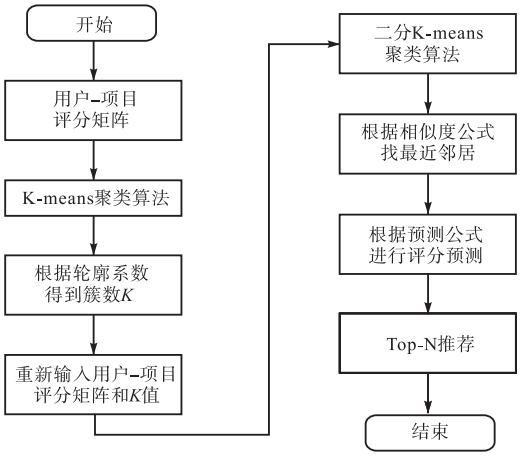

图 2　本文算法的流程图

Fig. 2　Flowchart of the algorithm in this article

3　实　验

3.1　数据集

采用豆瓣电影数据集，该数据集包含了 947 位用户对 1 494 部电影的 91 319 次评分，评分范围为 1 ~ 5，每一位用户评价至少 10 部电影. 数据集按照 70%、30% 的比例划分为训练集和测试集. 数据集属性包括用户 ID、产品 ID 以及相应的评分.

3.2　评价标准

采用平均绝对误差（MAE）、准确率（Precision）、召回率（Recall）以及 F_1 指标检测算法的推荐结果，公式为

$$\mathrm{MAE}=\frac{\sum_{i=1}^{N}|\hat{r}_i-r_i|}{N} \tag{7}$$

$$\mathrm{Precision}=\frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|R(u)|} \tag{8}$$

$$\mathrm{Recall}=\frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|T(u)|} \tag{9}$$

$$F_1=\frac{2\times\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \tag{10}$$

其中：$\hat{r}_i$ 为预测评分，r_i 为实际评分，N 为实际预测评分的个数，$R(u)$ 为算法针对用户 u 产生的推荐列表，$T(u)$ 为用户 u 在测试集上的所对应的行为记录. 准确率和召回率在某种程度上存在相互作用，即一个高的准确率可能会得到较低的召回率，为避免这一个问题，使用 F_1 指标进行综合考虑，F_1 指标越高，推荐效果越好.

3.3　实验结果与分析

为了得到聚类效果最好的簇数，使用轮廓系数作为评价聚类效果的标准. 不同聚类数目 K 下的轮廓系数值见表 1.

表 1　不同聚类数目下的轮廓系数

Tab. 1　Contour coefficients under different clustering numbers

聚类数目 K	轮廓系数 $S(i)$
6	0.309 183
8	0.502 895
10	0.653 229
12	0.640 921
14	0.599 815
16	0.498 730

由表 1 可知：当聚类数目 K 为 10 时，K-means 聚类算法的轮廓系数最高，即聚类效果最佳.

为了验证本文所提的算法能够有效地提高推荐结果的准确性，本文选取传统的基于用户的协同过滤算法，仅改进相似度的协同过滤算法，基于 K-means 的协同过滤算法作为本文的对比算法. 对比结果如图 3—图 6 所示.

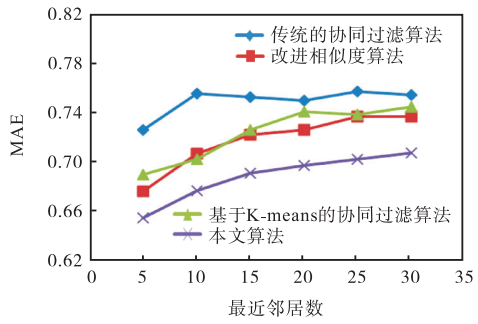

图 3　不同算法的平均绝对误差对比

Fig. 3　MAE comparison of different algorithms

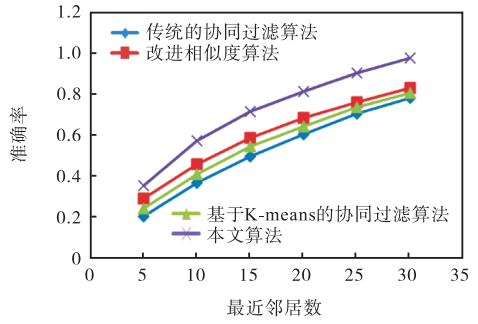

图 4　不同算法的准确率对比

Fig. 4　Precision comparison of different algorithms

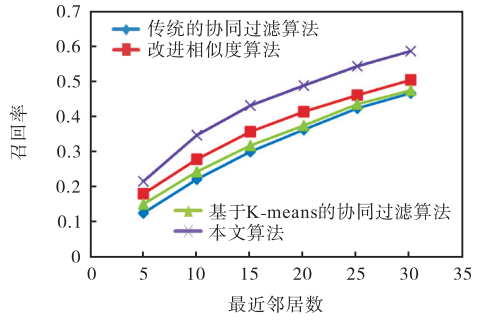

图 5　不同算法的召回率对比

Fig. 5　Recall comparison of different algorithms

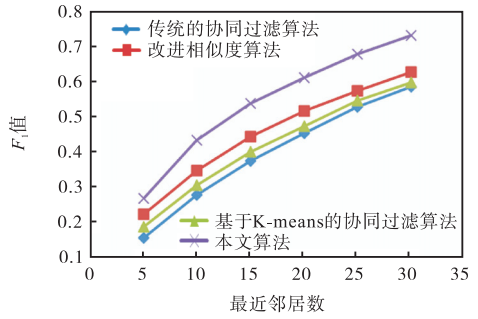

图 6　不同算法的 F_1 值对比

Fig. 6　Comparison of F_1 values of different algorithms

由图 3 可以看出，仅改变相似度计算方法或仅基于 K-means 的协同过滤算法的 MAE 值均比传统协同过滤算法的 MAE 值低，但本文算法的 MAE 值是最低的，这也进一步说明本文算法预测的分值与实际的分值更接近. 通过图 4—图 6 可以看出，上述算法的准确率、召回率以及 F_1 值均随着目标用户的最近邻居数的增加而增加. 首先，通过观察发现仅改变相似度计算方法或者仅基于 K-means 的协同过滤算法均能使传统的协同过滤算法的推荐效果提高，但推荐效果都不如本文的算法. 其次，通过实验结果发现寻找目标用户的邻居时，仅改变相似度的计算方法和仅基于 K-means 的协同过滤算法都在一定程度上使得用户间的相似度得到提高，但是本文将两者进行结合使得用户之间的相识度更加准确，进一步验证了相似度的计算是推荐算法的核心部分.

4　结　语

本文将聚类算法和协同过滤算法进行结合，并对聚类算法和相似度计算进行相应的改进，提出了基于 K-means 的改进协同过滤算法. 该算法在聚类过程中既考虑了 K 值选取的影响又考虑了随机质心选取的影响，使得在寻找最近邻居时的相似度更加准确. 采用豆瓣电影数据集进行对比实验，实验结果表明本文算法进一步提高了推荐的效果. 但是，面对数据集极度稀疏以及用户冷启动的问题还是无法解决，后续工作应该对算法进行相应的改进，以求进一步提高算法的准确度.

参考文献：

[1] 金丹，张娇娇，李依玲，等. 一种改进的协同过滤算法研究：以电影推荐系统为例[J]. 国际商务（对外经济贸易大学学报），2020（1）：128–141.

[2] 张怡文，王冉，杨安桔，等. 基于用户偏好度的双极协同过滤推荐算法[J]. 南京理工大学学报，2020，44（3）：313–319.

[3] LIU X. An improved clustering-based collaborative filtering recommendation algorithm[J]. Cluster computing，2017，20（2）：1281–1288.

[4] 姜婧. 推荐系统中协同过滤算法的研究与改进[D]. 南昌：南昌大学，2019.

[5] 刘超慧，韩传福，陈天成，等. 融合惩罚因子和时间权重的协同过滤推荐算法[J]. 信息技术与网络安全，2020，39（5）：17–21.

[6] 韩胜宝，伊华伟，李晓会，等. 基于融合相似度和层次聚类的冷启动推荐算法[J/OL]. 小型微型计算机系

（下转第 54 页）

ence on Computer Vision(ICCV). New York: 2019: 6022–6031.

[12] WU S, LI X, WANG X. IoU-aware single-stage object detector for accurate localization[J]. Image and vision computing, 2020, 97: 103911.

[13] 张勇, 张强, 徐林嘉, 等. 一种基于 YOLOv3 的静态手势实时识别方法: 201811137932.5[P]. 2019–02–12.

[14] ZHANG Q, ZHANG Y, LIU Z, et al. Real-time hand gesture recognition method based on improved YOLOv3[J]. Computer engineering, 2020, 46(3): 237–245.

[15] REDMON J, FARHADI A. YOLO9000: better, faster, stronger[C]//IEEE. 2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). New York: IEEE, 2017: 8100173.

[16] 李光华. 基于计算机视觉的手势识别技术研究与应用[D]. 成都: 电子科技大学, 2014.

责任编辑: 郎婧

(上接第 48 页)

统 : 1–8[2021–05–27]. http://kns.cnki.net/kcms/detail/21.1106.TP.20210517.1243.006.html.

[7] 印国成. 基于 K-means 的语义协同过滤推荐算法[J]. 扬州大学学报(自然科学版), 2018, 21(1): 46–49.

[8] KANIMONZHI S. Effective constraint based clustering approach for collaborative filtering recommendation using social network analysis[J]. Bonfring international journal of data mining, 2014, 1(1): 12–17.

[9] TSAIC F, HUNG C. Cluster ensembles in collaborative filtering recommendation[J]. Applied soft computing, 2012, 12(4): 1417–1425.

[10] 李顺勇, 张钰嘉, 张海玉. 基于 NKL 和 K-means 聚类的协同过滤推荐算法[J]. 河南科学, 2020, 38(1): 6–12.

[11] 岳希, 唐聃, 舒红平, 等. 基于数据稀疏性的协同过滤推荐算法改进研究[J]. 工程科学与技术, 2020, 52(1): 198–202.

[12] FENG C J, LIANG J Y, SONG P, et al. A fusion collaborative filtering method for sparse data in recommender system[J]. Information sciences, 2020, 521: 365–379.

[13] KOOHI H, KIANI K. User based collaborative filtering using fuzzy C-means[J]. Measurement, 2016, 91: 134–139.

[14] HERLOCKER J, KONSTAN J A, RIED J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms[J]. Information retrieval journal, 2002, 5(4): 287–310.

[15] 杨晓君. K-means 聚类算法研究及在股票投资的应用[D]. 重庆: 重庆大学, 2019.

[16] 吴金李, 张建明. 基于二分 K-means 的协同过滤推荐算法[J]. 软件导刊, 2017, 16(1): 26–29.

[17] 武建伟, 俞晓红, 陈文清. 基于密度的动态协同过滤图书推荐算法[J]. 计算机应用研究, 2010, 27(8): 3013–3015.

责任编辑: 郎婧