Modeling and Simulation 建模与仿真, 2024, 13(3), 2133-2146
Published Online May 2024 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2024.133196

Hans汉斯

基于合格邻居和异常检测的社区增强协同过滤

苏　湛，陈惠鑫，艾　均

上海理工大学光电信息与计算机工程学院，上海

收稿日期：2024年4月3日；录用日期：2024年5月6日；发布日期：2024年5月14日

摘　要

为了解决协同过滤推荐算法存在较大预测误差和推荐列表准确度不高的问题，提出一种结合异常检测和网络社区并基于所有合格邻居的协同过滤推荐算法。该算法使用修改的拉依达准则检测标记数据异常，在协同过滤相似度计算阶段降低与异常用户之间的相似性权重；使用得到的用户相似性建立网络模型，利用K核分解进行网络社区检测，在得到用户间的社区信息后对社区中用户进行相似性权重处理。基于MovieLens数据集并与五种同类型算法进行对比实验，结果表明，提出的算法可以有效降低预测误差以及提升推荐列表的排序准确度。

关键词

协同过滤，异常检测，相似性网络，社区检测，K-Core分解

Community-Augmented Cosine Collaborative Filtering Based on All Qualified Neighbors and Abnormal Detection

Zhan Su, Huixin Chen, Jun Ai

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Apr. 3rd, 2024; accepted: May. 6th, 2024; published: May. 14th, 2024

Abstract

In order to address the issue of the significant prediction errors and the low accuracy of recom-

文章引用: 苏湛, 陈惠鑫, 艾均. 基于合格邻居和异常检测的社区增强协同过滤[J]. 建模与仿真, 2024, 13(3): 2133-2146. DOI: 10.12677/mos.2024.133196

mendation lists in the collaborative filtering recommendation algorithm, a collaborative filtering recommendation algorithm that combines anomaly detection and network community based on all qualified neighbor is proposed. The algorithm uses the modified Pauta criterion for data anomaly detection and mark, and during the collaborative filtering similarity calculation stage reduces the similarity weight with abnormal users; Builds a network model using the obtained user similarity, and uses K-core decomposition for network community detection, and processes the similarity weights for users in the community after obtaining community information between users. Compared with five recommendation algorithms of the same type, and based on the MovieLens dataset, the experimental results showed that the proposed algorithm can effectively reduce prediction error and improve the accuracy of recommendation list rankings.

Keywords

Collaborative Filtering, Abnormal Detection, Similarity Network, Community Detection, K-Core Decomposition

Copyright © 2024 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 引言

推荐系统目前已经成为各类在线平台或在线应用的不可或缺的一部分。解决人们在互联网时代如何从大量数据中获取有用信息的问题，推荐系统是最有用的工具之一[1]。作为一种信息过滤系统，其主要目标是提供个性化的信息服务，帮助用户在海量的信息中找到他们真正需要和感兴趣的内容[2]。

推荐系统对于互联网大数据时代具有非同小可的意义和价值，因此，作为领域中的关键技术——协同过滤(Collaborative Filtering, CF)，得到了大量研究人员的重视[3] [4] [5]。但是，该项研究一方面仍存在着数据稀疏性[6]、冷启动[7]、可拓展性[8]等经典挑战，另一方面在邻居使用上的问题引起了研究者的关注[9]。

异常检测(anomaly detection, AD)是一种用于识别数据集中与正常模式不符的数据点或行为的技术[10]，检测算法可以分为七类，其中包含了 52 种异常检测算法[11]。拉依达准则(Pauta criterion)是最简单有效的异常检测算法之一[12]。

相似性网络(Similarity network)是推荐系统中的一种常见技术，是一个将用户或物品视为节点、用户或物品之间的相似度视为节点间链接权重的链式模型。Ai 等人[13]基于相似性网络设计了一种网络内资源分配的用户物品链接预测算法，缓解了推荐算法中预测准确性与多样性之间矛盾的问题以及存在的可扩展性问题。

本文旨在通过研究异常检测技术对异常数据的处理，以及结合相似性网络中相关技术的应用，解决协同过滤面对冷启动挑战带来的预测误差偏大的问题和面对海量信息带来的推荐列表排序不佳的问题。

2. 提出的算法

为了应对协同过滤算法存在较大的预测误差和推荐列表准确度不高的挑战，本文提出一种基于合格邻居和异常检测的社区增强协同过滤算法。如图 1 所示，算法包含了三个关键步骤：

1) 采用拉依达准则作为数据异常检测算法用于进行评分数据的检测和标记；

2) 在使用余弦相似度量计算完用户间的相似性后，选用训练集中所有合格邻居来构建网络；

3) 在构建的用户相似性网络中利用 K 核分解算法进行社区检测及标记。

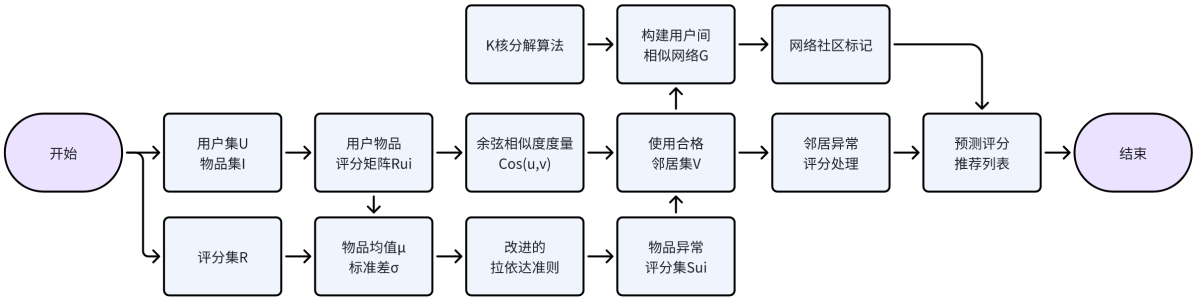

Figure 1. Algorithm structural diagram
图 1. 算法结构图

本文算法的总体流程如下：

1) 从评分集 R 获得每个物品的评分均值 μ 和标准差 σ，使用改进的拉依达准则筛选异常评分构成物品异常评分集 S_{ui}；

2) 以用户为行、物品为列构建用户物品评分矩阵 R_{ui}，其中未评分的数据设置为 0，基于对物品有共同评分的用户使用余弦相似度度量用户间的相似性，并依此选取每一位用户所有的合格邻居构成合格邻居集 V；

3) 构建以用户为节点、相似度为边的用户相似性网络 G，利用 K 核分解算法对相似性网络进行网络裁剪及社区标记，获得用户的社区信息；

4) 结合异常评分集，对合格邻居进行异常判断并处理，并且结合社区信息，最终用于用户项目的评分预测以及生成推荐列表。

2.1. 基于拉依达准则进行异常检测

拉依达准则又被称为 3σ 准则，准则要求被检测数据集基于正态分布的特性，认为数据点落在均值加减 3 倍标准差之外的概率很小，可以将这些数据点视为异常值。相关公式如式(1)：

$$P\left(\left|x-\mu\right|>3\sigma\right)\le 0.003 \tag{1}$$

式中，x 为数据值，μ 和 σ 分别为数据集的平均值和标准差，$P\left(X\right)$ 为符合条件下的概率。

为了研究分析海量数据中可能存在的异常数据对推荐算法的影响，本文对拉依达准则的应用进行了调整，相关公式如式(2)所示：

$$S_{ui}=\left(\left|r_{ui}-\overline{r_i}\right|>s\sigma_i\right) \tag{2}$$

式中，r_{ui} 代表用户 u 对物品 i 的评分，$\overline{r_i}$ 和 σ_i 分别代表物品 i 的平均分和标准差，S_{ui} 代表了用户 u 对物品 i 的异常评分集，即把用户对物品的评分与物品的平均分的差值与 s 倍物品标准差比较后，由满足条件的用户物品对构成。

算法的具体步骤如下：

1) 计算数据集中每个物品评分的均值 μ 以及标准差 σ；

2) 基于均值和标准差，进行项目异常评分的标记。计算物品评分及其均值的差值，并将差值大于 s 倍 σ 的物品及评分标记在物品异常评分集中，如式(2)；

3) 返回物品异常评分集。

2.2. 基于合格邻居的协同过滤和网络建模

本文算法是一种基于使用合格邻居的协同过滤。合格邻居需要满足下列条件：1) 与目标用户有超过一项共同评分的物品；2) 与目标用户可以计算余弦相似度；3) 可以进行异常判断，存在异常评分。

算法的具体流程包括以下三步：

1) 构建评分矩阵，计算用户间相似性 s_{uv}。本文算法在计算用户间相似度时是通过用户物品交互矩阵来计算用户间的余弦相似度。计算公式如式(3)所示：

$$s_{uv} = Cos(u,v) = \frac{\sum_{i \in I_u \cap I_v} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_u \cap I_v} r_{ui}^{\,2}} \cdot \sqrt{\sum_{i \in I_u \cap I_v} r_{vi}^{\,2}}} \tag{3}$$

式中，I_u 和 I_v 代表用户 u 和用户 v 评过分的物品集，r_{ui} 和 r_{vi} 代表用户 u 和用户 v 对物品 i 的评分。

2) 根据异常评分集，判断处理合格邻居。基于异常数据检测得到的异常检测评分集，其中包含了异常的用户物品对，通过对目标用户的合格邻居进行循环判断，若邻居对目标物品的评分属于异常集，则对这些异常邻居与目标用户的相似度进行 p 次幂处理。相关公式如式(4)：

$$s_{uv} = (s_{uv})^{p} \tag{4}$$

式中，s_{uv} 代表用户 u 和用户 v 的相似度，p 代表降幂权重。

3) 使用用户相似度进行网络建模。通过上述两步可以得到合格用户间的相似度，相似度网络建模基于每个用户为网络节点，用户间相似度为网络链接权重，建立无向链接，构成复杂网络。

2.3. K 核分解网络及社区划分

社区是相似性网络中的重要结构，通过将相似性网络中的一组相似度更高的节点归为一类，从而将网络划分成不同的社区。Ai 等人[14]基于用户之间和项目之间的相似性分别建立了用户－用户和项目－项目的网络，通过揭示网络中意义非凡的结构信息，改善了系统的预测和推荐结果。

然而，在复杂的相似性网络中通过社区检测来进行用户或项目的聚类会面临着大量的计算。因此，Ai 等人[15]通过结合模糊链接重要性，利用 K 核分解算法为社区检测降低了计算复杂度。K 核分解算法是依据节点度值从低到高依次去除节点的，一旦分解达到最高的核，即节点度值大于 K，每个节点及其关联节点都将会被标记在一个社区。具体流程如图 2 所示。

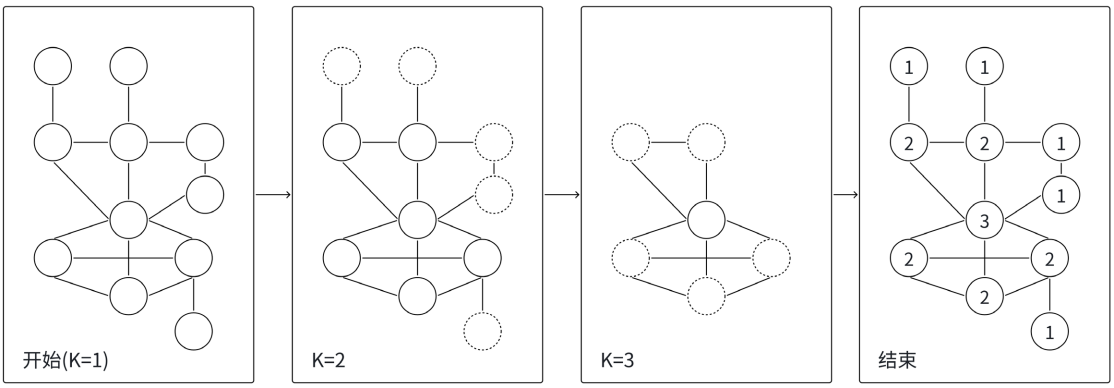

Figure 2. K-core decomposition flowchart (K = 3)
图 2. K-core 分解流程图(K = 3)

为了改善异常检测算法对协同过滤的影响以及增强算法模型在面对冷启动的挑战时的推荐性能，本文进行了相似性网络建模，并且采用 K 核分解算法进行了优化。结合图 2，社区划分步骤如下：

第一步，将网络中的每一节点都视为独立社区，并且从度值为 1 的节点开始进行第一轮递归检测，最大度值为 t 时，则需要进行 t 轮递归。

第二步，在第 k 次($k \le t$)递归中，删除度值等于或小于 k 的节点，直到网络中合格节点的最小度值大于 k，并且记录当前网络结构中度值最小节点的邻居。

第三步，如果邻居集为空，那么被删除节点的社区保持不变。如果邻居集为非空，那么从中找出权重最高的节点，并将其社区分配给该节点，再将依赖者的社区也合并到该节点的社区中。

第四步，如果网络中仍存在未被记录的节点，则从第二步开始进行下一轮的递归；否则，结束递归。

2.4. 基于异常数据和社区信息进行评分预测

在传统的基于用户的余弦协同过滤模型中，通常依据用户相似度矩阵来选择相似邻居计算目标用户对物品的评分，算法的评分预测公式如式(5)所示：

$$\tilde{r}_{ui} = \overline{r}_u + \frac{\sum_{v=1}^{k}\left(r_{vi} - \overline{r}_i\right)\cdot s_{uv}}{\sum_{v=1}^{k}\left|s_{uv}\right|} \tag{5}$$

式中，$\tilde{r}_{ui}$ 代表用户 u 对物品 i 的预测评分，$\overline{r}_u$ 代表用户 u 的平均评分，r_{vi} 代表邻居用户 v 对物品 i 的评分，$\overline{r}_i$ 代表物品 i 的平均评分，$\left|s_{uv}\right|$ 代表用户 u 和用户 v 之间相似度的绝对值。

在基于用户相似性建立网络模型后，通过 K 核算法对网络进行分解，得到标记好的社区，可以对目标用户与不同社区下用户邻居的权重进行调整，定义如式(6)：

$$\varphi = \begin{cases} \varphi_1 \in [1,2], c_u = c_v \\ \varphi_2 \in (0,1], c_u \ne c_v \end{cases} \tag{6}$$

式中，φ 代表用户 u 和用户 v 的社区权重，c_u 和 c_v 代表用户 u 和用户 v 所在的社区。

综上，结合式(4)~(6)，本文算法最终的预测评分公式如式(7)所示：

$$\tilde{r}_{ui} = \begin{cases} \overline{r}_u + \dfrac{\sum_{v=1}^{k}\left(r_{vi} - \overline{r}_i\right)\cdot {s_{uv}}^{p}\cdot\varphi}{\sum_{v=1}^{k}\left|{s_{uv}}^{p}\right|}, v \in S_{ui} \\ \overline{r}_u + \dfrac{\sum_{v=1}^{k}\left(r_{vi} - \overline{r}_i\right)\cdot s_{uv}\cdot\varphi}{\sum_{v=1}^{k}\left|{s_{uv}}^{p}\right|}, v \notin S_{ui} \end{cases} \tag{7}$$

3. 实验

3.1. 实验设计

本文采用了 MovieLens 最新的小型数据集进行实验。该数据集包含 100,836 个评分和 3683 个标签应用程序，涉及 9742 部电影，由 610 名用户在 1996 年 3 月 29 日至 2018 年 9 月 24 日期间创建。

本文实验在处理器为 Intel(R) Core(TM) i5-7300HQ CPU @ 2.50 GHz，8.0 GB 内存，操作系统为 64 位 Windows 10 的环境下进行，采用了相同的编程语言进行算法实现和实验结果可视化。

实验采用的验证方法是折五验证，即把数据集随机分成五份，选取一份作为测试集，其余四份作为训练集，依次选取每一份测试集并循环实验五次，将五次的实验结果求和取平均作为最终的实验结果。

3.2. 对比算法与评价指标

3.2.1. 对比算法

本文所提出的算法是对协同过滤推荐算法的改进。首先，通过使用异常检测技术来减少异常邻居评分，避免异常值对预测准确度的影响；然后，通过使用社区检测技术来分解相似度网络，削弱不相似的用户间的影响；最后，同时使用所有合格邻居进行预测和推荐，提升算法效率。为了验证算法性能的优劣，本文选用如下五种算法进行对比实验：

1) 用户观点传播算法(UOS) [16]：一种通过将用户观点传播过程与协同过滤相结合的评分预测方法。

2) 向量相似性算法(VS) [17]：一种根据项目特征在多维度上使用向量测量用户相似性的链式预测模型。

3) 相似性网络资源分配算法(SRA) [13]：通过将二分图模型和相似性网络相结合，实现资源分配并利用中心性和社区的网络特征预测用户物品链接。

4) 信息熵协同过滤算法(Entropy) [18]：通过利用用户评分的信息熵来改进相似性度量，从而反映用户对物品的全局评分行为。

5) 巴氏算法(Bhattacharyya) [19]：通过使用每个目标用户的最相似的邻居来预测物品，一般用于处理数据集稀疏的情况。

3.2.2. 评价指标

为了评估算法模型的综合性能，本文分别从评分预测准确性、分类预测准确性和推荐排序准确性三方面进行测评。平均绝对误差(Mean Absolute Error, MAE)和均方根误差(Root Mean Squared Error, RMSE)用于评估评分预测误差；准确性(Accuracy)和精确性(Precision)用于评估分类预测准确性；归一化折损累计增益(Normalized Discounted Cumulative Gain, NDCG)、排序准确性(Sorting Accuracy, SA)用于评估推荐列表排序准确性。

MAE 计算了算法预测值与真实值的差值之和的均值，而 RMSE 计算了算法预测值与真实值的差值平方和求平均后的开平方结果。相比于 MAE，RMSE 在对于异常值的敏感程度更高，受偏差值的影响更大。MAE 和 RMSE 的实验结果越小，预测精度越高，算法性能越好。两者计算公式如式(8)、式(9)所示：

$$\mathrm{MAE}=\frac{1}{n}\sum_{i=1}^{n}\left|r_{ui}-\tilde{r}_{ui}\right| \tag{8}$$

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(r_{ui}-\tilde{r}_{ui}\right)^{2}} \tag{9}$$

其中，n 是测试集中的样本数量；r_{ui} 和 $\tilde{r}_{ui}$ 分别代表用户 u 对物品 i 的预测评分和实际评分。

Precision 和 Accuracy 两个指标都是用于评估推荐算法的准确度度量。Precision 侧重考虑推荐结果的精确度，即推荐给用户喜欢的物品占所有推荐物品的比重；而 Accuracy 侧重考虑算法结果分类的准确度，即能正确地将喜欢的推荐给用户而不将不喜欢的推荐给用户占所有物品的比重。两者公式如式(10)、(11)：

$$\mathrm{Precision}=\frac{TP}{TP+FP} \tag{10}$$

$$\mathrm{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

其中，TP、TN、FP、FN 分别代表真阳性(推荐了用户喜欢的物品)，真阴性(没有将用户不喜欢的物品推荐给用户)，假阳性(推荐了用户不喜欢的物品)，假阴性(没有将用户喜欢的物品推荐给用户)。

NDCG 用于衡量推荐算法排序质量优劣，能够反映出推荐结果的排序准确性和相关性，越高相关性的物品出现在推荐列表越靠前的位置时，NDCG 指标越优。其计算公式如式(12)、(13)所示：

$$\mathrm{DCG}=\sum_{i=1}^{b}R_i+\sum_{i=b+1}^{N}\frac{R_i}{\log_b r_i} \tag{12}$$

$$\mathrm{NDCG}=\frac{\mathrm{DCG}}{\mathrm{DCG}_{\max}} \tag{13}$$

其中，R_i 代表推荐列表中第 i 个物品的相关度，若用户对该物品评分超过其平均评分，则 $R_i=1$，否则 $R_i=0$；b 代表折损程度，本文实验设置为 2；N 代表列表中所有物品集合；r_i 代表物品 i 在列表中的排列名次。

SA 是一种评估算法结果排序准确度的指标[20]。其计算公式如式(14)所示：

$$SA=\frac{\sum_{i=1}^{|L|}\prod_{j=i+1}^{|L|}I\left(S_i\geq S_{j+1}\right)}{|L|} \tag{14}$$

其中，S_i 和 S_{j+1} 是用户在推荐列表对 i 和 j 位置的物品评分；I 是指标函数，满足条件返 1，否则为 0；L 为推荐列表。

3.3. 参数选择和结果分析

3.3.1. 参数选择

参数 s 用于异常评分数据的筛选。由式(1)、式(2)可知，s 的取值处在 3 左右，因此，本文选取不同的 s 值对评分总数进行了过滤统计，如图 3 所示。

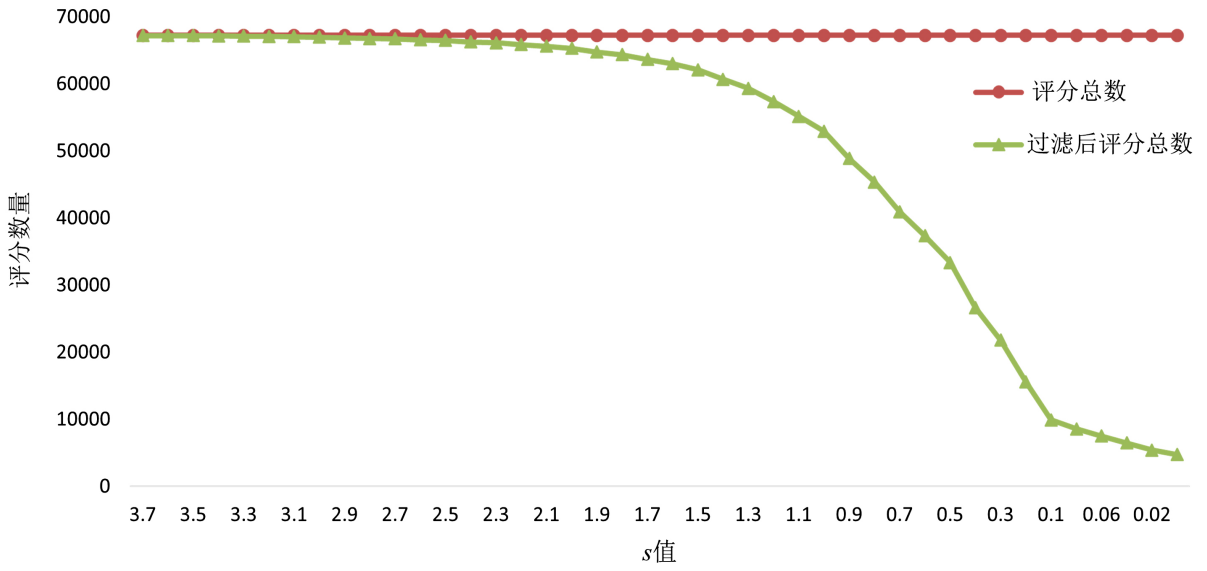

Figure 3. The sum of ratings under different s values
图 3. 不同 s 值下的评分总数

由图可知，当 s 值小于 2 时，评分数量才有明显减少；而当 s 值小于 1 时，评分数量快速减少。因此，本文对参数 s 在[1, 2]取值进行对比实验，实验结果的 MAE 指标如图 4 所示。当 s 取值为 1.5 时，MAE 值最低，这代表此时的 s 值能使算法误差降到最低。

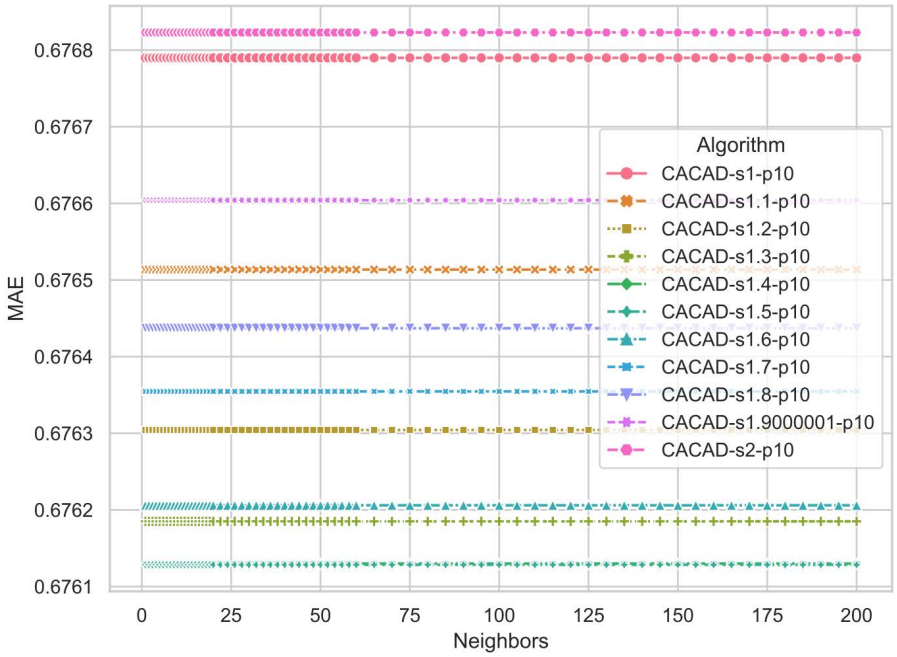

Figure4. MAE for algorithms under different *s* values
图 4. 不同 *s* 值下算法的 MAE

参数 *p* 用于对异常邻居与目标用户的相似度进行次幂处理。由式(4)可知，当参数 *p* 数值越大时，对相似度降权的程度越大。图 5、图 6 展示了算法在选择不同 *p* 值时的 MAE 指标和 NDCG 指标。由图可知 *p* 值越大，MAE 值越小，算法误差越小，但与此同时 NDCG 指标下降，算法分类性能降低。

综上所述，为了综合提高算法的性能，本文设置参数 $s = 1.5$，$p = 10$。

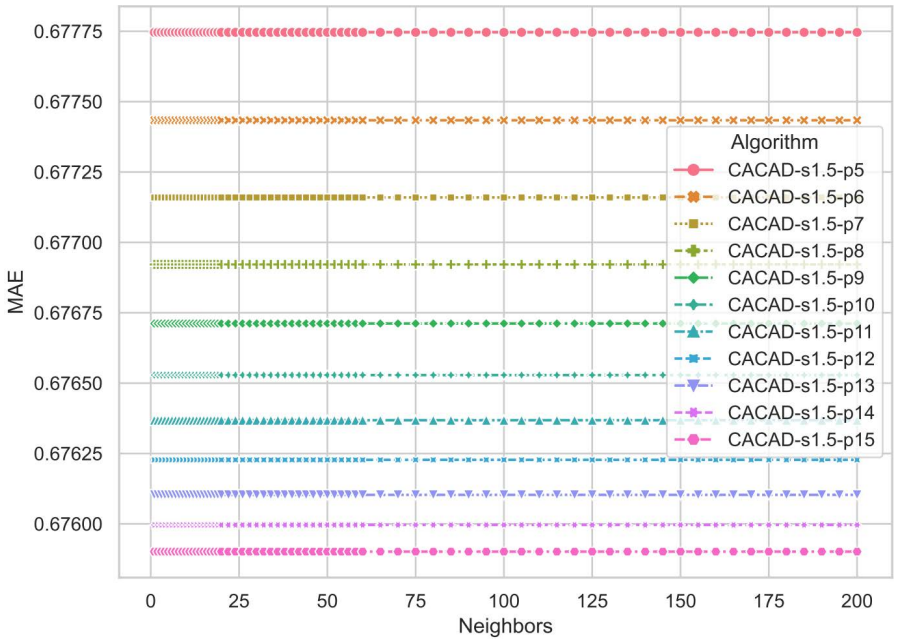

Figure 5. MAE for algorithms under different *p* values
图 5. 不同 *p* 值下算法的 MAE

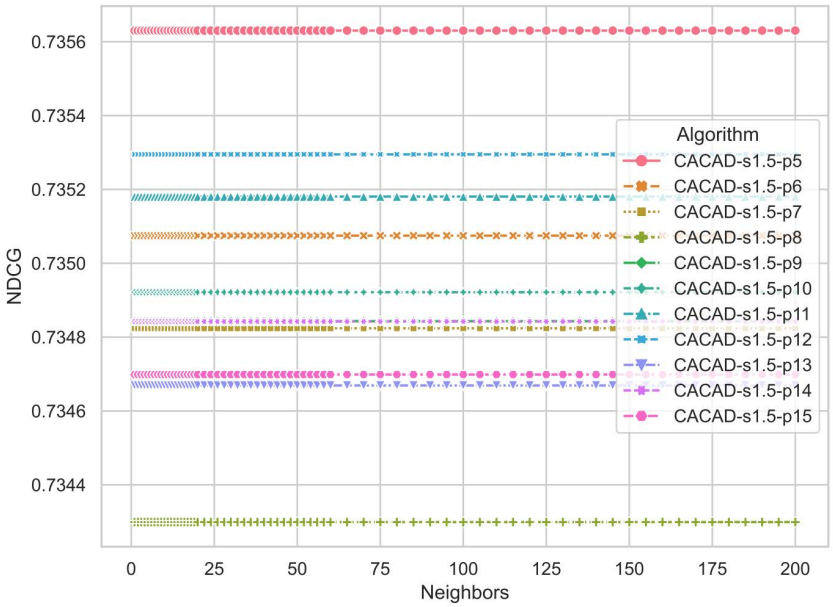

Figure 6. NDCG for algorithms under different p values
图 6. 不同 p 值下算法的 NDCG

3.3.2. 结果分析

图 7 展示了六种算法随邻居数量变化而变化的 MAE 实验结果。随着邻居数量的增多，五种对比算法都逐步达到误差的最低值，而本文算法的 MAE 误差因使用所有邻居的缘故误差值一直保持在 0.6802，并且保持最低位。较新的 SRA 算法误差值在 0.6809 附近位居第二，其次是 UOS (0.6843)、Entropy (0.6870)、VS (0.6924)，最后是巴氏算法(0.6949)。结果表明，本文算法能将 MAE 误差降低 0.59%~2.11%，能够较快速的实现更精准的预测结果。

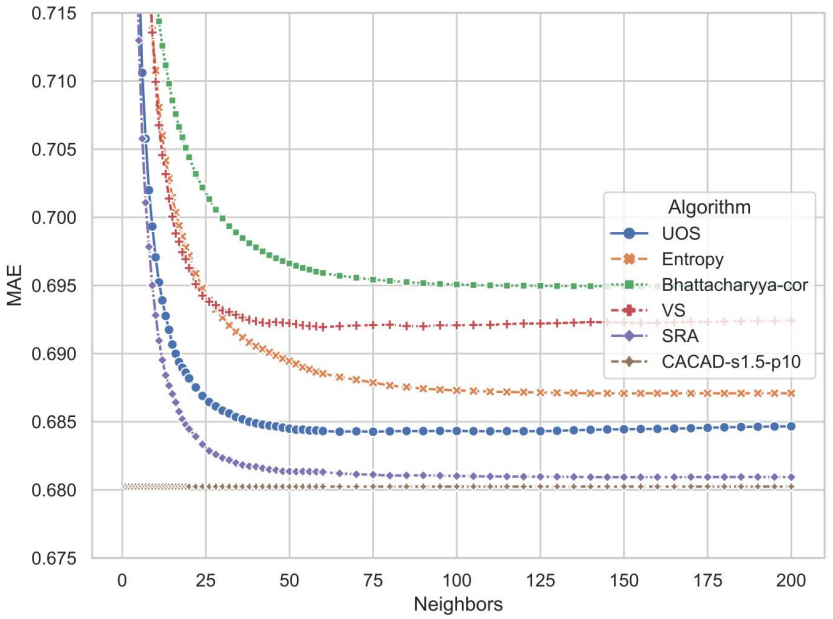

Figure 7. MAE for different algorithms
图 7. 不同算法的 MAE

图 8 展示的是六种算法随邻居数量变化而变化的 RMSE 误差结果。RMSE 对于误差的判断更加灵敏及严格。由图可见，随着邻居数量的增加，当五种对比算法的误差都达到最低值时，本文算法的得分仍然处于比较中的最低值，并且相比于 MAE，与其他算法得分拉开了更大的距离。本文算法将 RMSE 降低了 0.16%~3.08%，由此可见，本文算法无论在小误差还是大误差下，都能保持很好的预测准确性。

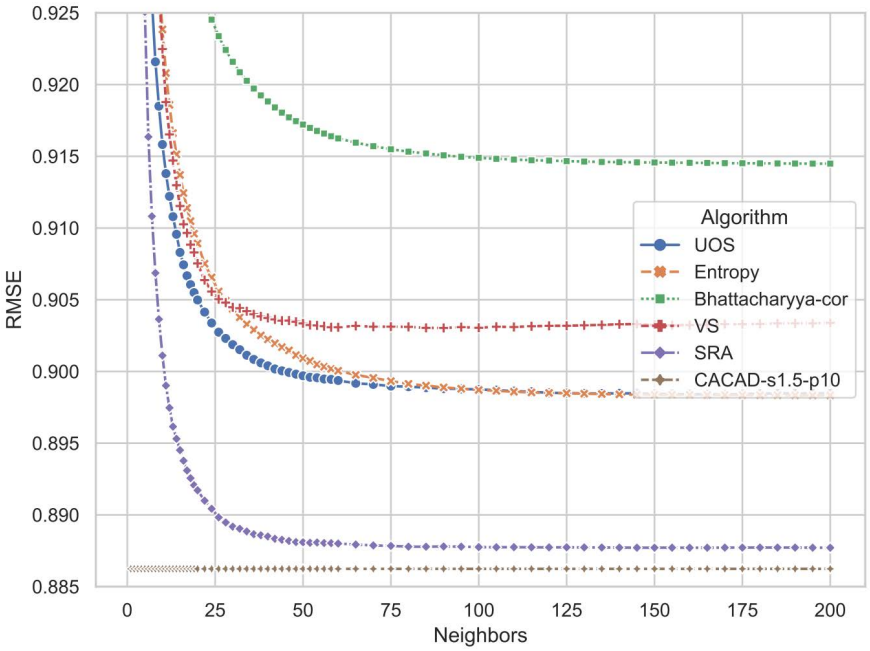

Figure 8. RMSE for different algorithms
图 8. 不同算法的 RMSE

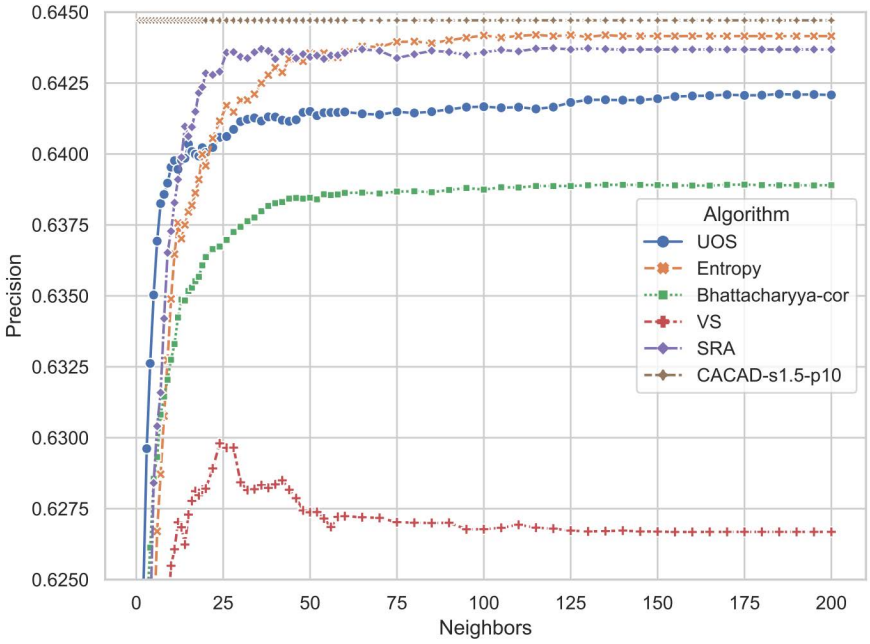

Figure 9. Precision for different algorithms
图 9. 不同算法的 Precision

如图 9 所示，本文算法的 Precision 得分(0.6447)在六种算法中排名第一，领先于 Entropy 算法(0.6441)第三位的 SRA 算法(0.6436)，排在后面的分别是 UOS (0.6421)、Bhattacharyya (0.6389)和 VS (0.6266)。与五种对比算法相比本文算法将 Precision 得分提高了 0.09%~2.89%，这代表本文算法在推荐用户喜欢的物品时占所有推荐的物品比例更高，能给用户带来更好的体验。

如图 10 所示，本文算法的 Accuracy 得分(0.6306)在六种算法中是第三。一方面，第一位的 Bhattacharyya 算法和第二位的 UOS 算法的 Accuracy 得分分别为 0.6360 和 0.6358，两者不相上下但领先于本文算法得分约 0.82%。另一方面，本文算法 Accuracy 得分比排在第四位的 Entropy (0.6301)、第五位的 SRA (0.6296)和第六位 VS (0.6257)高出 0.08%~0.78%。

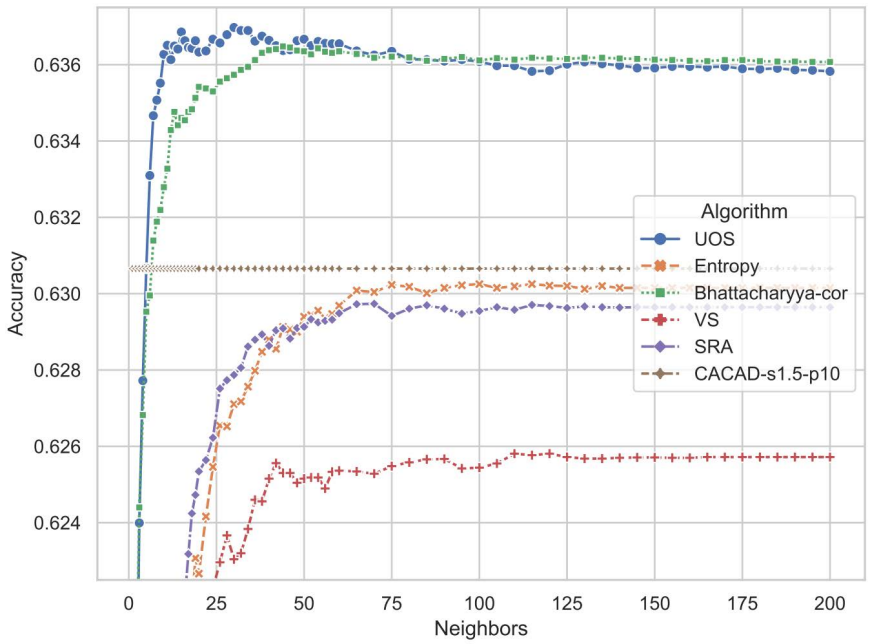

Figure 10. Accuracy for different algorithms
图 10. 不同算法的 Accuracy

根据式(10)、(11)以及对应的概念，结合两种指标得分可知，本文算法在推荐结果准确性上性能表现优异，而在分类准确性上性能表现一般。但在实际应用过程中，推荐结果的准确与否直接影响了用户对系统的体验感受，因此，本文认为 Precision 指标结果更具有参考价值。

如图 11 所示，本文算法的 NDCG 得分(0.7104)一直处于领先的位置。最新的 SRA 其 NDCG 得分略低与本文算法，随着用户数量的增加，其稳定在 0.7090。UOS 在用户数为 15 时达到最高得分 0.7059，但随着用户数增加，其得分持续降低，最终平衡在 0.7002。随后的 Entropy 和巴氏算法，其 NDCG 得分随用户数的增长呈上升趋势，最后得分稳定在 0.7084 和 0.7048。VS 的得分整体趋势先增后降，受邻居数量影响较大，最终得分为 0.6495。本文算法相比其他算法，NDCG 指标的提升范围在 0.20%~9.38%。

图 12 展示的是不同算法在 SA 排序指标上的得分结果。由图可知，相比于五种对比算法，本文算法得分(0.4802)整体上居于第一位。在用户数为 56 时，巴氏算法达到最高得分(0.4803)超过本文算法，但用户数的增加让其得分下降至最后的 0.4788。Entropy 算法和 SRA 算法的得分随用户数的增加稳步上升，最终排名稳定在 0.4795 和 0.4784。UOS 算法和 VS 算法的 SA 得分受用户数影响较大，用户数少时得分较高，但后续降低至 0.4732 和 0.4408。本文算法在 SA 指标上的提升达到了 0.15%~9.35%。

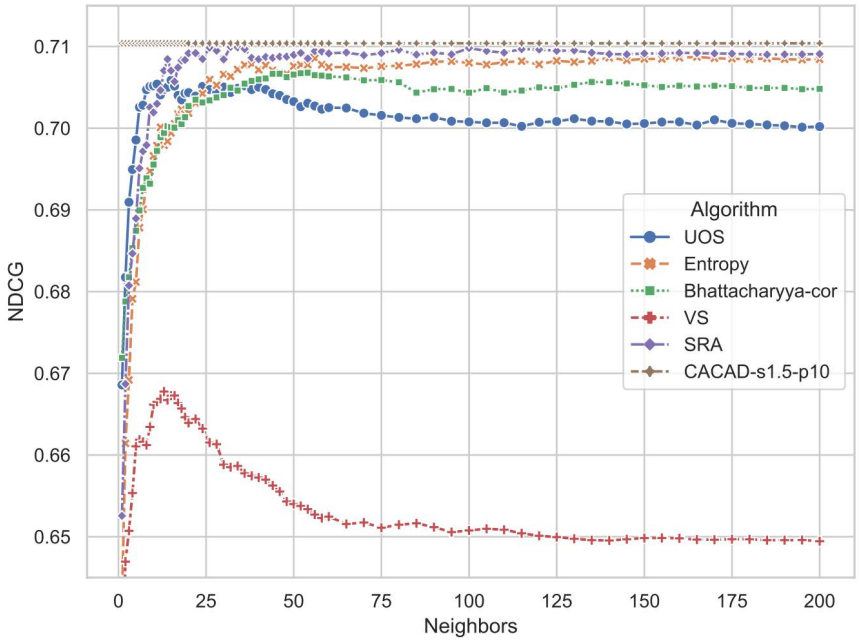

Figure 11. NDCG for different algorithms
图 11. 不同算法的 NDCG

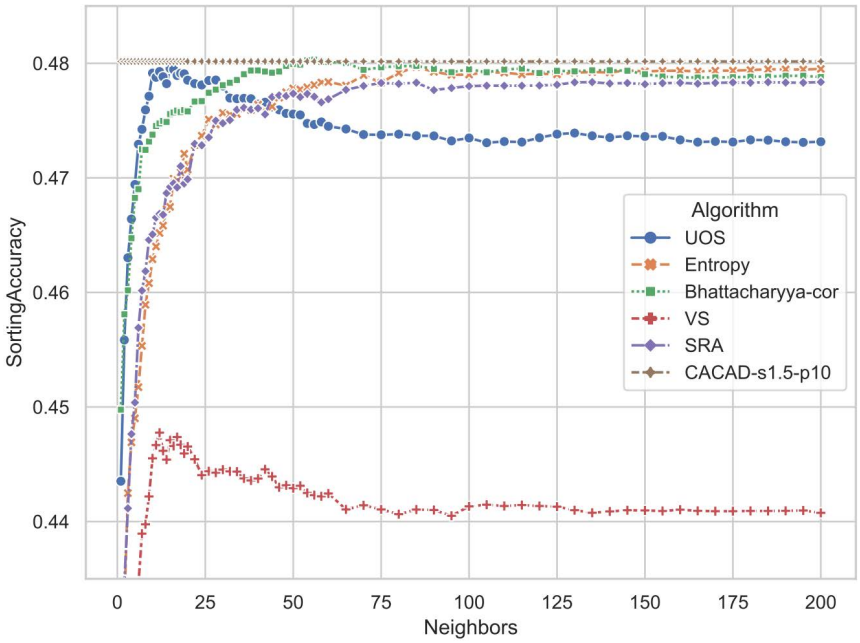

Figure 12. SA for different algorithms
图 12. 不同算法的 SA

4. 结论

针对协同过滤存在较大的预测误差和推荐列表准确度不高的问题，本文提出了一种基于合格邻居和异常检测的社区增强协同过滤算法。算法引入异常数据检测技术进行数据处理，同时结合相似性网络中的社区检测技术进行用户相似增强，最后基于所有的合格邻居进行协同过滤。

算法在 MovieLens 数据集中与其他算法进行对比，实验证明，本文提出的算法模在预测准确性上提升了 3%，在排序准确性上提高了 9%。然而，本文算法在分类准确性方面还有待提升。在未来的工作中，研究人员可以继续研究结合不同的异常检测算法进一步提高协同过滤算法的综合性能。

基金项目

国家自然科学基金项目(61803264)。

参考文献

[1] Yu, J., Yin, H., Xia, X., *et al*. (2023) Self-Supervised Learning for Recommender Systems: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, **36**, 335-355. https://doi.org/10.1109/TKDE.2023.3282907

[2] Gao, C., Zheng, Y., Li, N., *et al*. (2023) A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. *ACM Transactions on Recommender Systems*, **1**, 1-51. https://doi.org/10.1145/3568022

[3] Sun, N., Luo, Q., Ran, L., *et al*. (2023) Similarity Matrix Enhanced Collaborative Filtering for E-Government Recommendation. *Data and Knowledge Engineering*, **145**, Article 102179. https://doi.org/10.1016/j.datak.2023.102179

[4] Hu, P., Yang, E., Pan, W., *et al*. (2022) Federated One-Class Collaborative Filtering via Privacy-Aware Non-Sampling Matrix Factorization. *Knowledge-Based Systems*, **253**, Article 109441. https://doi.org/10.1016/j.knosys.2022.109441

[5] Wang, Q., Wu, S., Bai, Y., *et al*. (2023) Neighbor Importance-Aware Graph Collaborative Filtering for Item Recommendation. *Neurocomputing*, **549**, Article 126429. https://doi.org/10.1016/j.neucom.2023.126429

[6] Kuo, R.J. and Li, S.-S. (2023) Applying Particle Swarm Optimization Algorithm-Based Collaborative Filtering Recommender System Considering Rating and Review. *Applied Soft Computing*, **135**, Article 110038. https://doi.org/10.1016/j.asoc.2023.110038

[7] Hu, Z., Zhou, X., He, Z., *et al*. (2023) Discrete Limited Attentional Collaborative Filtering for Fast Social Recommendation. *Engineering Applications of Artificial Intelligence*, **123**, Article 106437. https://doi.org/10.1016/j.engappai.2023.106437

[8] Tao, Y., Kong, F., Shi, Y., *et al*. (2023) Efficient, Secure and Verifiable Outsourcing Scheme for SVD-Based Collaborative Filtering Recommender System. *Future Generation Computer Systems*, **149**, 445-454. https://doi.org/10.1016/j.future.2023.07.042

[9] Su, Z., Huang, Z., Ai, J., *et al*. (2022) Enhancing the Scalability of Distance-Based Link Prediction Algorithms in Recommender Systems through Similarity Selection. *PLOS ONE*, **17**, e0271891. https://doi.org/10.1371/journal.pone.0271891

[10] Yang, X. and Li, X. (2023) ATDAD: One-Class Adversarial Learning for Tabular Data Anomaly Detection. *Computers and Security*, **134**, Article 103449. https://doi.org/10.1016/j.cose.2023.103449

[11] Rodríguez, M., Tobón, D.P. and Múnera, D. (2023) Anomaly Classification in Industrial Internet of Things: A Review. *Intelligent Systems with Applications*, **18**, Article 200232. https://doi.org/10.1016/j.iswa.2023.200232

[12] Kumar, A., Parkash, C., Tang, H., *et al*. (2023) Intelligent Framework for Degradation Monitoring, Defect Identification and Estimation of Remaining Useful Life (RUL) of Bearing. *Advanced Engineering Informatics*, **58**, Article 102206. https://doi.org/10.1016/j.aei.2023.102206

[13] Ai, J., Cai, Y., Su, Z., *et al*. (2022) Predicting User-Item Links in Recommender Systems Based on Similarity-Network Resource Allocation. *Chaos, Solitons and Fractals*, **158**, Article 112032. https://doi.org/10.1016/j.chaos.2022.112032

[14] Ai, J., Liu, Y., Su, Z., *et al*. (2019) Link Prediction in Recommender Systems Based on Multi-Factor Network Modeling and Community Detection. *Europhysics Letters*, **126**, Article 38003. https://doi.org/10.1209/0295-5075/126/38003

[15] Ai, J., Liu, Y., Su, Z., *et al*. (2021) K-Core Decomposition in Recommender Systems Improves Accuracy of Rating Prediction. *International Journal of Modern Physics C*, **32**, Article 2150087. https://doi.org/10.1142/S012918312150087X

[16] He, X.-S., Zhou, M.-Y., Zhuo, Z., *et al*. (2015) Predicting Online Ratings Based on the Opinion Spreading Process. *Physica A*: *Statistical Mechanics and Its Applications*, **436**, 658-664. https://doi.org/10.1016/j.physa.2015.05.066

[17] Su, Z., Zheng, X., Ai, J., *et al*. (2020) Link Prediction in Recommender Systems Based on Vector Similarity. *Physica A*: *Statistical Mechanics and Its Applications*, **560**, Article 125154. https://doi.org/10.1016/j.physa.2020.125154

[18] Lee, S. (2020) Using Entropy for Similarity Measures in Collaborative Filtering. *Journal of Ambient Intelligence and Humanized Computing*, **11**, 363-374. https://doi.org/10.1007/s12652-019-01226-0

[19] Singh, P.K., Sinha, M., Das, S., *et al*. (2020) Enhancing Recommendation Accuracy of Item-Based Collaborative Fil-

tering Using Bhattacharyya Coefficient and Most Similar Item. *Applied Intelligence*, **50**, 4708-4731. https://doi.org/10.1007/s10489-020-01775-4

[20] Su, Z., Yang, H. and Ai, J. (2023) FPLV: Enhancing Recommender Systems with Fuzzy Preference, Vector Similarity, and User Community for Rating Prediction. *PLOS ONE*, **18**, e0290622. https://doi.org/10.1371/journal.pone.0290622