

# 基于改进型相似度的协同过滤算法的研究

吴锦昆, 单剑锋

(南京邮电大学 电子与光学工程学院、微电子学院, 江苏 南京 210023)

**摘要:** 如今的用户面对着大量的信息, 从中选取与自己联系密切的相关信息就显得非常困难了。电子商务平台、社区团购平台以及视频平台, 面对日益增长的用户数据, 需要从中挖掘有利于提高平台效率的信息, 因此对用户的个性化推荐提出了更高的要求。各平台都希望能够把握每一个用户的动态信息, 实施更加精准的个性化推荐, 个性化推荐不仅能够提高平台的效率, 也能够为用户带来极致的体验。由于传统的协同过滤算法评价体系中并没有考虑到不同用户存在评分的差异, 所以在准确性、精确性、差异性等方面仍然需要进一步的提高。该文主要针对在电影评分推荐协同过滤算法中精准化的需求, 引入用户差异因子  $w_v^u$ , 融合原有的皮尔逊相似度计算, 从而解决传统的协同过滤算法相似性计算中, 针对不同用户具有不同评价体系存在一定偏差的问题。采用 Movielens 1m 电影评分数据集进行仿真, 证明了改进后相似度计算的协同过滤算法能够降低 MAE 值。

**关键词:** 相似度; 协同过滤; 个性化推荐; 数据处理; 用户差异因子

中图分类号: TP393

文献标识码: A

文章编号: 1673-629X(2022)04-0039-05

doi:10.3969/j.issn.1673-629X.2022.04.007

## Research on Collaborative Filtering Algorithm Based on Improved Similarity

WU Jin-kun, SHAN Jian-feng

(School of Electronic and Optical Engineering and Microelectronics, NJUPT, Nanjing 210023, China)

**Abstract:** Nowadays, users are faced with a large amount of information, and it is very difficult to select relevant information that has close contact with them. E-commerce platforms, community group buying platforms and video platforms, in the face of ever-increasing user data, need to dig out information that is conducive to improving the efficiency of the platform, so it puts forward higher requirements on personalized recommendations for users. All platforms hope to be able to grasp the dynamic information of each user and implement more accurate personalized recommendations. Personalized recommendations can not only improve the efficiency of the platform, but also bring the ultimate experience to users. Because the traditional collaborative filtering algorithm evaluation system does not take into account the differences in the ratings of different users, the accuracy, precision, and differences still need to be further improved. We mainly aim at the need for precision in the movie rating recommendation collaborative filtering algorithm, introducing user difference factors  $w_v^u$ , fusing the original person similarity calculations, so as to solve the problem that there is a certain deviation in the traditional collaborative filtering algorithm similarity calculation for different users with different evaluation systems. The simulation on Movielens 1m movie scoring data set shows that the improved collaborative filtering algorithm for similarity calculation can reduce the MAE value.

**Key words:** similarity; collaborative filtering; personalized recommendation; data processing; user difference factor

## 0 引言

在大数据时代, 每个个体在日常生活上网产生的一些行为, 都会生成大量的数据。这些数据对于互联网平台来说是描述用户的重要信息来源, 再利用这些数据结合利用协同过滤算法, 能够促进互联网平台的高效运行。当人们处在面对大量的信息与平台推送各

种各样的短信时, 要从海量的信息中获取有用的信息是困难的, 面对海量的信息更是一个令人头疼至极的问题。所以大数据被运用的场合越开越普遍, 抖音、拼多多、淘宝等这些互联网平台都会根据用户的历史行为以及对产品的偏好, 来对用户进行个性化、精准化推荐, 给用户带来更加良好的体验。

收稿日期: 2021-06-02

修回日期: 2021-10-08

基金项目: 江苏省教育科学“十三五”规划 2020 年度课题(B-a/2020/01/01)

作者简介: 吴锦昆(1997-), 男, 硕士研究生, 研究方向为数据挖掘、协同过滤算法; 单剑锋, 博士, 副教授, 硕导, 研究方向为调制识别、数据挖掘、故障诊断。

不论是基于用户还是基于物品的协同过滤算法,数据总是存在稀疏性的问题。文献[1]中利用随机性对原有的用户-商品评分矩阵进行填充再进行矩阵变换,文献[2]提出挖掘用户的相关信息,文献[3]提出利用 BP 神经网络来预测用户的评分,从而改进数据集稀疏性的问题。文献[4]中考虑用户的偏好以及物品的内在属性从而改进相似度计算,改进了协同过滤算法相似计算。文献[5]提出了一种融合型余弦相似度计算方法,其中包含了相似度修正参数和基于用户属性的特征向量,解决了用户之间不同评价体系和用户的不同属性的问题。文献[6]提出了一种全新的相似度计算方法,利用两个用户对相同物品的评价的比值作为相似度计算的依据,使提出的协同具有更高的精确度和较低的平均绝对误差。文献[7]利用项目的标签及评级信息,引入了时间加权因子来解决数据的稀疏性的问题。文献[8]根据不同用户的评分范围,将用户分为正常、严格、宽裕和中间四种,提出了基于标准差比例的改进和统一的基线改进模型,以便消减经典基线估计模型的局限性。文献[9]根据 Translation-based 模型在稀疏的数据集上的良好表现,将这种模型与递归神经网络模型进行融合,提出了 Recurrent Translation-based Network 模型。文献[10]采取随机梯度下降法优化了矩阵分解,结合谱聚类方法填充方法,有效降低了数据的稀疏性对于准确性的影响。文献[11]采用主成分分析法,对用户-项目评分矩阵进行降维,再对 K-means 算法进行改进,提高了聚类的速度,有效提高了准确率和召回率。

在上述研究的基础上,该文从计算的相似度出发,结合其他改进相似度方法,根据用户自身评分的差异,引入了用户差异因子  $w_v^u$ ,以凸显用户的评价体系,用来改进皮尔逊相似度计算方法,突出用户评分具有一定的差异,从而降低了 MAE 值。

## 1 协同过滤算法

现今,协同过滤算法大致分为三类:一类是基于用户的协同过滤算法,一类是基于物品的协同过滤算法,还有一类是基于模型的协同过滤算法。

协同过滤算法被广泛应用于人们日常生活的方方面面。例如音乐、电影、网购等。为了使用户能够更加迅速地浏览与自己偏好相关的音乐、视频、商品,各个平台都在寻找更加适合平台的推荐算法,给用户带来极致的体验。

协同过滤算法,主要功能是找出与某一用户相似的用户,将相似用户的所评级或消费的商品推荐给这个用户;或者根据用户历史消费过的商品,推荐给用户相似的商品<sup>[12]</sup>。分别是基于用户的协同过滤算法和

基于物品的协同过滤算法。

### 1.1 相似度计算

在传统的协同推荐算法中,计算相似度的方法主要有两种:皮尔逊相似度、余弦相似度。该文主要针对协同过滤算法中的皮尔逊相似度计算进行改进。皮尔逊相似度计算公式如下:

$$\text{sim}(u_1, u_2) = \frac{\sum_{i \in I_{u_1, u_2}} (r_{u_1, i} - \bar{r}_{u_1}) (r_{u_2, i} - \bar{r}_{u_2})}{\sqrt{\sum_{i \in I_{u_1, u_2}} (r_{u_1, i} - \bar{r}_{u_1})^2 \sum_{i \in I_{u_1, u_2}} (r_{u_2, i} - \bar{r}_{u_2})^2}} \quad (1)$$

式中,  $u_1, u_2$  表示两个不同的用户,  $I_{u_1, u_2}$  表示  $u_1, u_2$  两个用户共同评价过的电影的集合;  $r_{u_1, i}, r_{u_2, i}$  表示用户  $u_1, u_2$  对于电影  $i$  的评分;  $\bar{r}_{u_1}, \bar{r}_{u_2}$  表示用户  $u_1, u_2$  对于所有电影的平均评分。

余弦相似度类似于求两个向量形成夹角的余弦值,根据用户对电影的评分形成空间向量<sup>[13]</sup>,计算空间向量形成夹角的余弦值,确定不同用户之间的相似度。

余弦相似度计算公式如下:

$$\text{Sim}(u_1, u_2) = \frac{\sum_{i \in I} r_{u_1, i} r_{u_2, i}}{\sqrt{\sum_{i \in I} r_{u_1, i}^2} \sqrt{\sum_{i \in I} r_{u_2, i}^2}} \quad (2)$$

$$\text{Sim}(i_1, i_2) = \frac{\sum_{u \in U} r_{u, i_1} r_{u, i_2}}{\sqrt{\sum_{u \in U} r_{u, i_1}^2} \sqrt{\sum_{u \in U} r_{u, i_2}^2}} \quad (3)$$

式中,  $I, U$  分别表示  $u_1, u_2$  评价过的项目的交集,对  $i_1, i_2$  评价过用户的交集。相较于皮尔逊相似度,余弦相似度中没有考虑到用户评价的平均分,忽略了用户自身带有的评价体系。

### 1.2 协同过滤算法准确度的评价准则

准确度的评价指标,是衡量协同过滤算法准确度的一个重要方面。选取合适的评价指标能够很好地衡量算法改进前后的效果。文献[14]中,详细阐述了推荐系统中的各种评价指标,从预测评分的准确度上有平均绝对误差、均方根误差,根据最终的平均绝对误差值的大小,从而确定算法的优劣,得到的平均绝对值较小的协同过滤算法推荐的准确度较高。从分类的准确度上有准确率、召回率,还有一些其他评价指标,深入分析与比较各个指标之间的差异。该文采用平均绝对误差(MAE)来衡量改进相似度计算后的推荐算法的性能。

#### 1.2.1 平均绝对误差

平均绝对误差是最经典的计算真实评分与预测评分的差异。表达式如下:

$$\text{MAE} = \frac{1}{|U_T|} \sum_{u,i \in U_T} |r_{u,i} - \hat{r}_{u,i}| \quad (4)$$

式中,  $U_T$  表示测试集,  $r_{u,i}$  表示测试集中用户  $u$  对项目  $i$  的真实评分,  $\hat{r}_{u,i}$  表示测试集中用户  $u$  对项目  $i$  的预测评分。

### 1.2.2 均方根绝对误差

均方根绝对误差,是将绝对误差求平方之后,再求平方根。表达式如下:

$$\text{RMSE} = \sqrt{\frac{1}{|U_T|} \sum_{u,i \in U_T} (r_{u,i} - \hat{r}_{u,i})^2} \quad (5)$$

## 2 改进的相似度计算方法

在原来的皮尔逊相似度计算中,没有考虑每个用户的评分范围与用户之间的评分差异。在文献[15]中,引入用户评价的平均分来区分用户的差异。该文引入用户差异因子  $w_{u_i}^u$ ,利用用户评价的最大值可以

$$\text{sim}(u_1, u_2) = \frac{0.5^2 + 0.5^2 + 0.5^2 + 0.5^2}{\sqrt{(0.5^2 + 0.5^2 + 0.5^2 + 0.5^2) \times (0.5^2 + 0.5^2 + 0.5^2 + 0.5^2)}} = 1 \quad (8)$$

$$\text{sim}(u_1, u_3) = \frac{0.5^2 + 0.5^2 + 0.5^2 + 0.5^2}{\sqrt{(0.5^2 + 0.5^2 + 0.5^2 + 0.5^2) \times (0.5^2 + 0.5^2 + 0.5^2 + 0.5^2)}} = 1 \quad (9)$$

$$\text{sim}(u_1, u_4) = \frac{0.5^2 + 0.5^2 + 0.5^2 + 0.5^2}{\sqrt{(0.5^2 + 0.5^2 + 0.5^2 + 0.5^2) \times (0.5^2 + 0.5^2 + 0.5^2 + 0.5^2)}} = 1 \quad (10)$$

$$\text{sim}(u_2, u_3) = \frac{0.5^2 + 0.5^2 + 0.5^2 + 0.5^2}{\sqrt{(0.5^2 + 0.5^2 + 0.5^2 + 0.5^2) \times (0.5^2 + 0.5^2 + 0.5^2 + 0.5^2)}} = 1 \quad (11)$$

用户  $u_1, u_2$ , 用户  $u_1, u_3$  以及用户  $u_1, u_4$  的相似度都为 1, 可以判断出用户  $u_1, u_2, u_3, u_4$  之间具有相同的相似度, 但是从用户  $u_1, u_2, u_3, u_4$  对电影的评分的不同可以看出他们之间的相似度应该有一定的差异, 可以从表中的数据判断出用户  $u_1, u_2$  之间的相似度应该高于用户  $u_1, u_3$  之间的相似度, 用户  $u_1, u_4$  之间的相似度

$$\text{sim}(u_1, u_2) = \frac{0.5^2 + 0.5^2 + 0.5^2 + 0.5^2}{\sqrt{(0.5^2 + 0.5^2 + 0.5^2 + 0.5^2) \times (0.5^2 + 0.5^2 + 0.5^2 + 0.5^2)}} \times \frac{2}{4} = 0.5 \quad (12)$$

$$\text{sim}(u_1, u_3) = \frac{0.5^2 + 0.5^2 + 0.5^2 + 0.5^2}{\sqrt{(0.5^2 + 0.5^2 + 0.5^2 + 0.5^2) \times (0.5^2 + 0.5^2 + 0.5^2 + 0.5^2)}} \times \frac{2}{5} = 0.4 \quad (13)$$

$$\text{sim}(u_1, u_4) = \frac{0.5^2 + 0.5^2 + 0.5^2 + 0.5^2}{\sqrt{(0.5^2 + 0.5^2 + 0.5^2 + 0.5^2) \times (0.5^2 + 0.5^2 + 0.5^2 + 0.5^2)}} \times \frac{2}{3} \approx 0.67 \quad (14)$$

$$\text{sim}(u_2, u_3) = \frac{0.5^2 + 0.5^2 + 0.5^2 + 0.5^2}{\sqrt{(0.5^2 + 0.5^2 + 0.5^2 + 0.5^2) \times (0.5^2 + 0.5^2 + 0.5^2 + 0.5^2)}} \times \frac{4}{5} = 0.8 \quad (15)$$

用户  $u_1, u_2$  之间的相似度为 0.5, 用户  $u_1, u_3$  之间的相似度为 0.4, 用户  $u_1, u_4$  之间的相似度大约为 0.67, 用户  $u_2, u_3$  之间的相似度为 0.8, 可以看出改进后的相似度计算方式得出用户  $u_1, u_2, u_3, u_4$  之间具有不同的相似度。经过改进后的皮尔逊相似度得到的计算结果表明, 引入用户差异因子改进的皮尔逊相似度计算方式, 更加能够体现出不同用户之间的差异, 能够对之前相似性较高的用户进行更加精细的区分。

衡量出不同用户评价的苛刻性, 来区别出不同用户之间的差异性。改进后的相似度计算公式如下:

$$\text{sim}(u_1, u_2) = \frac{\sum_{i \in I_{u_1, u_2}} (r_{u_1, i} - \bar{r}_{u_1}) (r_{u_2, i} - \bar{r}_{u_2})}{\sqrt{\sum_{i \in I_{u_1, u_2}} (r_{u_1, i} - \bar{r}_{u_1})^2 \sum_{i \in I_{u_1, u_2}} (r_{u_2, i} - \bar{r}_{u_2})^2}} \times w_{u_2}^{u_1} \quad (6)$$

$$w_{u_2}^{u_1} = \frac{\min(r_{u_1, \max}, r_{u_2, \max})}{\max(r_{u_1, \max}, r_{u_2, \max})} \quad (7)$$

式中,  $w_{u_2}^{u_1}$  表示用户之间的差异因子,  $r_{u_1, \max}, r_{u_2, \max}$  表示用户对于物品的评分的最高分。

皮尔逊相似度改进前与改进后的相似度计算结果比较:

在表 1 中, 存在 4 个用户, 分别对四个物品都进行了评分, 依据式(1)可以计算出用户之间的皮尔逊相似度, 计算结果如式(8)~式(11)所示。

应该高于用户  $u_1, u_2$  和用户  $u_1, u_3$  之间的相似度, 而不是四者之间具有相同的相似度。

为了解决上述计算结果中用户之间存在差异不明显的问题, 引进用户差异因子来区分用户, 改进后的皮尔逊相似度计算过程如式(12)~式(15)所示。

表 1 不同用户对已购商品的评分

	$i_1$	$i_2$	$i_3$	$i_4$
$u_1$	1	2	1	2
$u_2$	3	4	3	4
$u_3$	4	5	4	5
$u_4$	2	3	2	3

### 3 实验步骤及结果分析

#### 3.1 实验步骤

仿真过程建立在 Spyder 平台上基于 Python3.8 版本进行,主要建模仿真步骤如下:

(1)导入相关实验相关的 Movielens 1m 相关数据集。每一次都将数据集按照 7:3 的比例划分为训练集和测试集,经过三次得到不同的测试集与训练集,部分数据集如表 2 所示。

(2)构建用户-项目评分矩阵,分别采用皮尔逊相似度与改进后的皮尔逊相似度计算公式,来计算用户之间的相似度。

(3)根据计算得到的相似度,选择邻居个数,预测用户对测试项目的评分。

(4)根据预测得到的评分计算 MAE 值。

表 2 Movielens 1m 部分数据集

UserID	MovieID	ratings	timestamp
1598	674	4	974734718
1599	3053	4	974822095
1600	1444	4	974775887
1601	2278	3	978578456
1602	407	2	974732598
1603	3623	3	974731325
1604	535	5	974730792
1605	2092	2	975383190
1606	1370	5	974741744
1607	920	1	974733478
1608	913	5	974730646
1609	1393	4	974747989
1610	1240	4	974730336
1611	2054	2	981617193
1612	2531	3	974729699
1613	3798	3	974730200
1614	2723	2	974729868
1615	2158	3	974729037
1616	7	4	974734116
1617	778	1	975339950

在表 2 中 UserID 代表用户编号,MovieID 代表电影编号,部分电影编号对应的电影如表 3 所示,ratings 代表对应的用户给对应电影的评分,timestamp 代表时间戳,即用户对电影评分的时间。

表 3 部分电影编号与对应电影

MovieID	Name(电影名称)
674	Barbarella
3053	Messenger: The Story of Joan of Arc

续表 3

MovieID	Name(电影名称)
1444	Guantanamo
2278	Ronin
407	In the Mouth of Madness
3623	Mission: Impossible 2
535	Short Cuts
2092	Return of Jafar, The
1370	Die Hard 2
920	Gone with the Wind

#### 3.2 实验数据集介绍及数据集处理

该文采用的数据集来自 Movielens 数据集集中的 ml-1m 数据集。目前主要的协同过滤算法基本上都采用 Movielens 类数据集。在 ml-1m 数据集中包含了 3 952 部电影,6 040 个用户,评分在 1~5 之间。将数据集划分为训练集和测试集,训练集用来计算用户之间的相似度,测试集用来根据训练集计算出的相似度进行预测评分<sup>[13]</sup>。

#### 3.3 实验结果及分析

对于最近邻数分别选取  $k=25, 50, 75, 100$ , 根据划分不同的训练集与测试集分别进行 3 次仿真实验,对同一次划分的训练集和测试集,先用改进前的协同过滤算法进行仿真,然后用改进后的协同过滤算法对划分好的数据集进行仿真,仿真后得到 MAE 值,然后再对三次得到的 MAE 值求平均值,以消除仿真得到的 MAE 值存在一定的偶然性误差。经过仿真实验得到不同的最近邻数的结果分别如表 4、表 5 所示。

表 4 算法改进前 MAE 值

	25	50	75	100
1	0.735 66	0.726 84	0.721 42	0.722 52
2	0.736 88	0.726 63	0.724 05	0.720 44
3	0.735 82	0.726 61	0.724 08	0.721 47
平均值	0.736 12	0.726 69	0.723 18	0.721 48

表 5 算法改进后 MAE 值

	25	50	75	100
1	0.732 65	0.723 80	0.721 22	0.719 52
2	0.735 18	0.723 90	0.722 29	0.720 50
3	0.733 41	0.723 46	0.721 08	0.721 22
平均值	0.733 74	0.723 72	0.721 53	0.720 41

根据表 4、表 5 中的第四行中的平均值,得到的 MAE 平均误差曲线如图 1 所示。

从图 1 中可以看出,改进后协同过滤算法的平均 MAE 值基本上都低于改进前的平均 MAE 值,在近邻  $k$  值选取较小的值时,MAE 值下降比较明显,特别是



在最近邻数  $k$  选取 50 时, MAE 值有较大的下降。在近邻  $k$  取较大值, 由于选取的近邻  $k$  值较大, 会降低协同过滤算法相似度计算的作用, 导致 MAE 值并没有明显的下降。改进后的算法在最近邻数  $k$  选取较大时, 效果不够明显, 也是未来算法改进的一个方向。

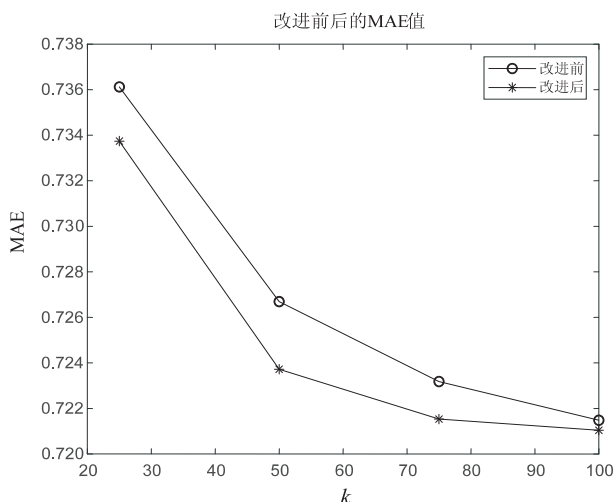


图1 改进前后 MAE 值曲线

#### 4 结束语

因为传统的协同过滤推荐算法在皮尔逊相似度计算中只考虑用户之间的线性关系和用户的平均分, 没有考虑到每个用户更加深层次的评价体系, 用户本身的评价体系会对用户对商品的评分产生一定的影响, 所以在利用皮尔逊相似度计算用户之间的相似度时会出现一定的误差。

该文考虑到用户的评分最大值可能一定程度上构成了用户评价体系的一部分, 提出了一种改进型相似度的协同过滤算法, 即引入了用户差异因子, 改进了原有的相似度计算方式, 在相似度计算中, 能够起到一定的区分用户的作用, 因此能够得到较为准确的相似度, 从而提升推荐的准确度。

根据在 Spyder 平台基于 python3.8 仿真后得到的数据表明, 引入用户差异因子后的改进型皮尔逊相似度计算方式能够降低平均绝对误差 (MAE), 可以说该改进型协同过滤算法在准确度方面的性能有一定的提升。但是当邻居个数 ( $k$ ) 选取较大时, MAE 值没有较大的改善, 未来值得进一步探讨。

尽管经过改进后的协同过滤算法改善了传统的协同过滤推荐算法忽略每个用户拥有各自的评分差异, 但是依然存在一些问题。首先, 该算法增加了一定的计算量, 提升了计算的复杂度; 其次, 没有解决协同过

滤算法存在的冷启动问题; 最后, 将数据集划分为测试集与训练集并没有解决 ml-1m 数据集中存在稀疏性的问题。因此, 改进的协同过滤算法仍然需要进一步的完善。

#### 参考文献:

- [1] YANG Yongjian, XU Yuanbo, WANG En, et al. Improving existing collaborative filtering recommendations via serendipity-based algorithm[J]. IEEE Transactions on Multimedia, 2018, 20(7): 1888-1890.
- [2] 李 锐. 基于矩阵分解的个性化推荐算法研究[D]. 哈尔滨: 哈尔滨理工大学, 2020.
- [3] 孙 乐. 基于矩阵填充与 BP 神经网络的混合推荐算法[J]. 信息与电脑, 2020, 32(21): 41-43.
- [4] 郑小楠, 谭钦红, 马 浩, 等. 基于用户偏好矩阵填充的改进混合推荐算法[J]. 计算机工程与设计, 2020, 41(10): 2784-2790.
- [5] 徐志超, 单剑锋. 基于改进型协同过滤算法的研究[J]. 计算机技术与发展, 2019, 29(10): 196-200.
- [6] WU Xiaokun, CHENG Bo, CHEN Junliang. Collaborative filtering service recommendation based on a novel similarity computation method[J]. IEEE Transactions on Services Computing, 2017, 10(3): 352-365.
- [7] ZHANG Chunxia, YANG Ming, YANG Wanqi. An improved hybrid collaborative filtering algorithm based on tags and time factor[J]. Big Data Mining and Analytics, 2018, 1(2): 128-136.
- [8] TAN Zhenhua, HE Liangliang, WU Danke, et al. Personalized standard deviations improve the baseline estimation of collaborative filtering recommendation[J]. Applied Sciences, 2020, 10(14): 4756-4776.
- [9] CHAIRATANAKUL N, MURATA T, LIU X. Recurrent translation-based network for top-n sparse sequential recommendation[J]. IEEE Access, 2019, 7: 131567-131576.
- [10] 王璇璇, 陈宁江, 练林明, 等. 基于谱聚类和矩阵分解的改进协同过滤推荐算法[J]. 广西大学学报: 自然科学版, 2020, 45(2): 313-320.
- [11] 陈 希. 基于降维和聚类的推荐算法研究与应用[D]. 南京: 南京邮电大学, 2020.
- [12] 郝立燕, 王 靖. 基于项目流行度的协同过滤 TopN 推荐算法[J]. 计算机工程与设计, 2013, 34(10): 3497-3501.
- [13] 吴彦文, 齐 旻, 杨 悦. 一种基于改进型协同过滤算法的新闻推进系统[J]. 计算机工程与科学, 2017, 39(6): 1179-1185.
- [14] 朱郁筱, 吕琳媛. 推荐系统评价指标综述[J]. 电子科技大学学报, 2012, 41(2): 163-175.