文章编号:1007－6735(2020)03－0275－08　DOI: 10.13255/j.cnki.jusst.20190420001

一种基于协同过滤和混合相似性模型的推荐算法

丁家满[1,2]，　沈书琳[1,2]，　贾连印[1,2]，　游进国[1,2]，　李润鑫[1,2]
（1. 昆明理工大学 信息工程与自动化学院，昆明 650500；2. 云南省人工智能重点实验室，昆明 650500）

摘要：针对推荐系统协同过滤方法中存在的数据稀疏和冷启动等问题，提出一种基于协同过滤和混合相似性模型的推荐算法。该算法首先计算用户在不同项目间的相似性，然后结合项目特性和标签信息权重来描述用户、项目、特性和标签之间的关系；其次，设定用户偏好因子和不对称因子调整不同用户间的评分偏好；最后，结合用户间相似性、项目综合权重，以及评分偏好构建混合相似性模型，并加入用户时间权重信息解决项目冷启动问题。在公开的 MovieLens 数据集上的实验表明，该算法在各种评估指标上比其他相关方法获得更显著的结果。

关键词：推荐算法；协同过滤；混合相似；冷启动
中图分类号：TP 31　　**文献标志码：**A

Recommendation algorithm based on collaborative filtering and hybrid similarity model

DING Jiaman[1,2], SHEN Shulin[1,2], JIA Lianyin[1,2], YOU Jinguo[1,2], LI Runxin[1,2]
(1. *Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China*; 2. *Artificial Intelligence Key Laboratory of Yunnan Province, Kunming 650500, China*)

Abstract: Like other recommendation paradigms, the algorithm based on collaborative filtering also suffers from the curse of sparse data and cold-start problem. Nevertheless, a recommendation algorithm based on collaborative filtering and hybrid similarity model was proposed. First, the similarity of users among different items was calculated by the algorithm, and then the relationship among users, items, features and tags was described according to property weights and tag weights. Next, the score preference between different users was adjusted by setting the user preference factor and the asymmetry factor. After that, a hybrid similarity model was constructed based on the user similarity, item weights and rating preferences, and the user-time weight information was also added to solve the project cold-start problem. Experiments on publicly accessible MovieLens data sets demonstrate that the algorithm achieves more prominent results than other related approaches in various evaluation metrics.

Keywords: *recommendation algorithm*; *collaborative filtering*; *hybrid similarity*; *cold-start*

收稿日期: 2019−04−20
基金项目: 国家自然科学基金资助项目（51467007，61562054）
第一作者: 丁家满（1974−），男，副教授．研究方向：数据挖掘．E-mail：tjoman@126.com

随着大数据时代的到来，信息与日俱增，使得信息消费者难以从大量数据中获取对自己有用的信息，而数据生产者也无法使自己的信息在众多信息中得到用户的关注，这就是信息过载问题[1]。推荐系统通过收集用户的爱好、行为习惯等信息并加以分析，帮助用户快速准确地找到自己想要的信息，被认为是缓解信息过载问题最有效的方法之一[2]。目前推荐系统所采用的推荐技术主要包括关联规则(association rules)[3]、基于内容的推荐(content-based recommendation)[4]、协同过滤(collaborative filtering, CF)、基于效用的推荐(utility-based recommendation)[5]和混合推荐(hybrid approach)[6-7]等。其中，协同过滤推荐算法是推荐系统中应用最广泛最成功的推荐技术之一[8]，其目标是根据已知的信息计算未知的评级[9]。主要包括基于用户(user-based)的协同过滤算法[10]、基于项目(item-based)的协同过滤算法[11]、基于模型(model-based)的协同过滤算法[12-13]，它们在针对不同目标时都取得了很好的推荐效果。

但协同过滤算法仍存在着许多问题，最主要的是数据稀疏问题和冷启动问题。现实生活中用户的评分数量较少，占总项目数的比例较低，导致提取的用户偏好特征较少，推荐结果不准确，这就是数据稀疏问题。为了解决协同过滤算法中的数据稀疏问题，文献[14]从用户偏好和全局角度提出了一种基于用户偏好聚类的有效协同过滤算法，以减少数据稀疏的影响。文献[15]提出了一种基于优化用户相似度的协同推荐算法。在传统的余弦相似算法中加入了一个平衡因子，用于计算不同用户之间的项目评价尺度差异。文献[16]提出一种混合用户相似性方法，该方法在特定领域取得了较好推荐结果，但仍然存在冷启动方面的问题。为了有效解决冷启动问题，文献[17]提出了将主题模型与矩阵分解(MF)相结合的LDA_MF模型，并融合改进的结合内容与行为的LDA_CF，Item_CF算法形成混合算法，提高长尾应用的推荐率。文献[18]提出了用户时间权重信息概念，结合项目属性信息解决完全新项目冷启动问题，但过于依赖共同评分项目，在数据稀疏环境下，效果不佳。文献[19]提出一种利用多群组智慧协同过滤算法，并结合用户偏好模型，对于解决用户冷启动问题取得了较好的效果，但对于多群组数据缺失或者稀疏情况，推荐效果有待提高。

以上文献分别在解决数据稀疏和冷启动问题方面取得不错的成果，但如何有效地同时解决这两个问题有待改进。为此，本文提出一种基于协同过滤和混合相似模型的推荐算法，利用PSS(proximity-significance-singularity)模型计算不同用户在不同项目间的相似度，并将不同项目间的评分关系作为权重调整用户相似度；接着，采用用户–项目–特性和用户–项目–标签的三分图形式描述用户、项目、特性、标签之间的关系，得到项目特性和项目标签的关系权重；同时，设定用户偏好因子和不对称因子作为权重调整不同用户间的评分偏好并提高模型的可靠性；最后，结合用户时间权重信息，构成基于协同过滤和混合相似模型的推荐算法。

1 问题定义

定义1 用户相似度

假设两个用户 u，v 分别评价两个项目 i，j，评分等级分别为 r_{ui} 和 r_{vj}，项目评分权重为 S_{item}，则用户相似度定义如式(1)所示。

$$sim(u,v)=\sum_{i\in I_u}\sum_{j\in I_v}S_{\text{item}}(i,j)S_1(r_{ui},r_{vj}) \tag{1}$$

式中，函数 $S_1(r_{ui},r_{vj})$ 代表仅基于用户不同项评级的相似度，利用文献[16]中调整后的PSS模型进行计算。函数 $S_1(r_{ui},r_{vj})$ 具体公式如式(2)所示。

$$S_1(r_{ui},r_{vj})=Proximity(r_{ui},r_{vj})\cdot Significance(r_{ui},r_{vj})Singularity(r_{ui},r_{vj}) \tag{2}$$

其中，

$$Proximity(r_{ui},r_{vj})=1-\frac{1}{1+\exp\left(-\left|r_{ui}-r_{vj}\right|\right)} \tag{3}$$

$$Significance(r_{ui},r_{vj})=\frac{1}{1+\exp\left(-\left|r_{ui}-r_{\text{med}}\right|\left|r_{vj}-r_{\text{med}}\right|\right)} \tag{4}$$

$$Singularity(r_{ui},r_{vj})=1-\frac{1}{1+\exp\left(-\left|\frac{r_{ui}+r_{vj}}{2}-\frac{\mu_i+\mu_j}{2}\right|\right)} \tag{5}$$

Proximity 函数(见式(3))用于描述用户间的评分绝对差值对相似度的影响；*Significance*(见式(4))函数描述用户评分与评分域中值之间的关系；r_{med} 表示评分域中值，如果用户评级距离评分域中值较远，则此评级更为重要；*Singularity*(见式

(5))函数描述一个评级对的均值与这两个项目的全局评分均值的绝对差值对相似度的影响，μ_i，μ_j 表示项目 i，j 的平均评分值。

S_1 函数采用的是不同用户在不同项目间的相似度计算，为了使不同项目具有可比性，本文使用 S_{item} 函数表示项目间的评分关系，并将此作为权重因子来调整用户之间的相似度。为了使其能够充分利用项目 i，j 的所有评分来解决共同评分的问题，这里式(6)采用调整后的标准化欧式距离进行计算。

$$D(i,j)=\sqrt{\sum_{k=1}^{n}\left(\frac{r_{ik}-r_{jk}}{s_k}\right)^2} \tag{6}$$

式中：r_{ik}，r_{jk} 表示某一用户对项目 i 和项目 j 的评分值；s_k 表示用户评分的标准差。这里考虑没有评分的项以用户评分均值代替。为了使项目间的相似权重更为准确，将标准欧式距离进行归一化处理，使其结果在(0,1]之间，得到项目评分权重函数 S_{item} 如式(7)所示。

$$S_{\text{item}}=\frac{1}{1+D(i,j)} \tag{7}$$

定义 2 项目特性信息

设 f_{ui} 表示用户 u 是否评价过项目 i，f_{ia} 表示项目 i 是否具有特性 a，C_i 表示项目 i 具有的特性个数，C_a 表示共同拥有特性 a 的项目个数，则项目特性信息如式(8)所示。

$$S_a^i(u)=\sum_{a=1}^{m}\frac{1}{C_a}\sum_{i=1}^{n}\frac{f_{ui}f_{ia}}{C_i} \tag{8}$$

项目特性信息描述用户、项目与特性之间的关系。如果用户 u 评价过项目 i，则 f_{ui}=1，反之 f_{ui}=0；同样，如果项目 i 具有特性 a，则 f_{ia}=1，反之，f_{ia}=0。

定义 3 项目标签信息

设 f_{ui} 表示用户 u 是否评价过项目 i，f_{ti} 表示标签 t 是否标注过项目 i，C_i 表示项目 i 被标注的标签个数，C_t 表示标签 t 标记过的项目个数，即拥有共同标签的项目个数，则项目标签信息如式(9)所示。

$$S_t^i(u)=\sum_{t=1}^{q}\frac{1}{C_t}\sum_{i=1}^{n}\frac{f_{ui}f_{ti}}{C_i} \tag{9}$$

项目标签信息描述用户、项目与标签之间的关系。如果项目 i 具有标签 t，则 f_{ti}=1，反之，f_{ti}=0。

定义 4 不对称因子

设 I_u 为用户 u 的所有评分项目数量，I_v 为用户 v 的所有评分项目数量，则不对称因子定义如式(10)所示。

$$S_2(u,v)=\frac{1}{1+\exp(-|I_u\cap I_v|/|I_u|)} \tag{10}$$

不对称因子 S_2 通过描述两个用户的共同评分项目数量与目标用户评分项目数量的比例来强调用户间的不对称性。如果共同评分项目数量与用户 u 的所有评分项目数量的比例大于其与用户 v 的所有评分项目数量的比例，则表示共同评分项目对用户 u 的影响要大于用户 v。

定义 5 偏好因子

设 μ_u 和 μ_v 表示用户 u 和用户 v 的评分均值，σ_u 和 σ_v 表示用户 u 和用户 v 的评分标准差。偏好因子 S_3 如式(11)所示。

$$S_3(u,v)=1-\frac{1}{1+\exp(-|\mu_u-\mu_v||\sigma_u-\sigma_v|)} \tag{11}$$

偏好因子 S_3 通过描述用户之间评分均值和标准方差的绝对差值乘积来调整用户的偏好，减少偏好差异带来的影响。

定义 6 用户时间权重

在现实生活中，不同用户有不同的评级偏好。一些用户更喜欢追求新事物，这些用户为“积极用户”，其评论时间与项目发布时间间隔较短。相反，有些用户更喜欢别人给出过评分的项目，这类用户为“消极用户”，其评论时间与项目发布时间间隔较长。将用户评论时间与项目发布时间间隔作为用户时间权重信息，在推荐新项目时，优先推荐给积极用户。用户时间权重为用户评价项目总数与用户对项目的评价时间和项目发布时间的时间间隔总和的比值。设 sum 为用户 u 评价项目总数，$time_{ui}$ 表示用户 u 对项目 i 的评价时间，$date_i$ 表示项目 i 的发布时间，则用户 u 的时间权重如式(12)所示。

$$w_u=\frac{sum}{\sum_{i=1}^{sum}(time_{ui}-data_i)} \tag{12}$$

可以看出，($time_{ui}$−$date_i$)越小，用户时间权重 w_u 越大，说明该用户越喜爱新的事物，为积极用户；反之，说明该用户为消极用户。

算例 1 为了更好理解，以用户 u_1，u_2，u_3 和项目 i_1，i_2，i_3，i_4 为例描述用户时间权重信息，具体信息如表 1 所示。

表1 用户评价时间表

Tab.1 User appraisal schedule

项目	u_1		u_2		u_3	
	time	*date*	*time*	*date*	*time*	*date*
i_1	2016	2001	2018	2001	2012	2001
i_2	2016	2013	2018	2013	2017	2013
i_3	2015	2015	2017	2015	2016	2015
i_4	0	2017	0	2017	0	2017

其中 *time* 代表用户评价时间，*date* 代表项目发布时间，根据式(12)可以得到用户的时间权重信息如下：

$$w_{u_1}=\frac{sum}{\sum_{i=1}^{sum}\left(time_{u_1 i}-data_i\right)}=\frac{3}{15+3+0}=0.170$$

$$w_{u_2}=\frac{sum}{\sum_{i=1}^{sum}\left(time_{u_2 i}-data_i\right)}=\frac{3}{17+5+2}=0.125$$

$$w_{u_3}=\frac{sum}{\sum_{i=1}^{sum}\left(time_{u_3 i}-data_i\right)}=\frac{3}{11+4+1}=0.188$$

由例子可以看出，用户 u_3 的时间权重最高，即用户 u_3 较用户 u_1 和 u_2 更喜欢体验新事物，为积极用户，则优先将新项目推荐给 u_3。

定义7 预测评分值

为了得到用户对未评级项的预测评分值，需要计算目标用户与其他用户的相似度，筛选出与目标用户相似度最高的前 K 个邻居用户组成邻近集，再根据预测评分公式进行计算。设 pre_{ui} 表示用户 u 对未评级项目 i 的预测评分值，$\bar{r}_u$和$\bar{r}_v$分别表示用户 u 和用户 v 的平均评分值，r_{vi} 表示用户 v 对项目 i 的评分，K 表示用户 u 的近邻集合，$S(u,v)$表示用户 u 和用户 v 的相似度，则预测评分值如式(13)所示。

$$pre_{ui}=\bar{r}_u+\frac{\sum_{v\in K}S(u,v)(r_{vi}-\bar{r}_v)}{\sum_{v\in K}|S(u,v)|} \tag{13}$$

2 基于协同过滤和混合相似模型的推荐算法

传统的相似性度量多采用用户间的共同评分项进行线性计算[20]，但实际情况中，数据稀疏问题使得共同评分项并不常见，变量之间通常也不存在线性关系，因此传统推荐算法的准确度也随之降低。而为了解决共同评分项问题，采用不同用户在不同项之间的评分来进行相似度计算。并且为了更好地适应非线性情况，考虑累积所有可能的评分，并将不同项之间的评分关系作为权重因子调整用户相似度。

在考虑用户评分的基础上，还要考虑项目本身的特性，特别是新项目本身没有用户行为，因此更需要结合项目特性来解决项目冷启动问题。同时，标签既能表达用户对项目的自我理解，也能体现出项目本身所具备的特征，因此结合标签信息进行推荐更能体现用户的兴趣和项目特点。

传统的相似性度量在计算用户相似度时通常将其看作对称模式[21]，即两个用户间的相似影响相同。但在实际情况中，两个用户的评分数一般并不相同，相似影响也不相同。因此，设计一个不对称因子，用以强调用户间相互影响的不对称性。同时，考虑到用户都有各自的评分偏好，有些用户更喜欢评价高分，即使项目并不理想；也有些用户更倾向于评价低分，哪怕项目很好也不会给很高的评分。为了平衡用户偏好，使结果更为理想，设计偏好因子。

综上考虑，提出一种混合相似性模型如式(14)所示。

$$\begin{aligned}S(u,v)=&S_2(u,v)S_3(u,v)\cdot\\&\sum_{i\in I_u}\sum_{j\in I_v}(\alpha S_a((u,i)(v,j))+\\&\beta S_t((u,i)(v,j)))S_{\text{item}}(i,j)S_1(r_{ui},r_{vj})\end{aligned} \tag{14}$$

其中，

$$S_a((u,i)(v,j))=\frac{S_a^i(u)S_a^j(v)+1}{sumitem+1}$$

$$S_t((u,i)(v,j))=\frac{S_t^i(u)S_t^j(v)+1}{sumitem+1}$$

即利用PSS模型计算用户间的相似度 S_1，并使用项目间评分关系权重 S_{item} 调整用户相似度；接着，描述用户、项目、特性、标签之间的关系，得到项目特性权重$S_a^i(u)S_a^j(v)$和项目标签权重$S_t^i(u)S_t^j(v)$；同时，设定用户不对称因子 S_2 和偏好因子 S_3 作为权重调整不同用户间的评分偏好并提高模型的可靠性，构成混合相似性模型。式中：α，β代表信息偏好权重，即用户对项目特性信息和标签信息的偏好比重，取值范围为(0,1)，且

$\alpha+\beta=1$；*sumitem* 为项目总数。

最后结合用户时间权重信息，使用预测评分公式(14)得到目标用户对未评级项目的预测评分值，结合用户时间权重信息式(12)组成式(15)，实现推荐。

$$pre_{ui}=\begin{cases}\bar{r}_u+\dfrac{\sum\limits_{v\in K}S(u,v)(r_{vi}-\bar{r}_v)}{\sum\limits_{v\in K}|S(u,v)|}, & i\text{不是新项目}\\ \bar{r}_u+(w_u+S_a^i(u))r_{\max}, & i\text{是新项目}\end{cases} \tag{15}$$

算例 2 为了更好地理解以上公式，下面根据用户–项目评分矩阵(表 2)，项目–特性矩阵(表 3)以及项目–标签矩阵(表 4)给出的例子进行分步说明。

表 2 评分为 1，2，3，4，5 共 5 个等级；表 3 表示如果项目具有某一属性，则设值为 1，否则设值为 0；表 4 同理。首先以用户 u_1 为研究目标，根据表 2 给出的用户评分矩阵，结合式(1)计算用户 u_1 分别与 u_2，u_3 的相似度。这里以 u_1，u_3 分别评价项目 i_1，i_2 为例，进行具体公式计算如下：

$$S_1(r_{u_1i_1},r_{u_3i_2})=0.268\,9\times0.5\times0.437\,8=0.058\,9$$

$$D(i_1,i_2)=\sqrt{\left(\frac{3-3.5}{0.5}\right)^2\left(\frac{4.3-5}{0.94}\right)^2\left(\frac{3-4}{0.49}\right)^2}=5.710$$

$$S_{\text{item}}=\frac{1}{1+D(i_1,i_2)}=0.149\,0$$

为了更好地理解和计算项目特性权重信息和项目标签权重信息，以项目特性权重信息为例，将用户–项目矩阵、项目–特性矩阵转化为用户–项目–特性三分图形式，具体如图 1 所示。

表 2　用户–项目评分矩阵

Tab.2　User-item rate matrix

用户	项目					
	i_1	i_2	i_3	i_4	i_5	i_6
u_1	3	–	4	–	–	–
u_2	–	5	–	3	5	–
u_3	3	4	4	3	4	–

表 3　特性–项目矩阵

Tab.3　Attribute-item matrix

特性	项目					
	i_1	i_2	i_3	i_4	i_5	i_6
a_1	1	1	1	0	0	1
a_2	0	0	1	0	1	0
a_3	1	0	1	1	0	0
a_4	0	1	0	1	0	1
a_5	0	0	0	1	1	0

表 4　标签–项目矩阵

Tab.4　Tag-item matrix

标签	项目					
	i_1	i_2	i_3	i_4	i_5	i_6
t_1	1	1	0	0	0	0
t_2	0	1	1	0	1	0
t_3	1	0	0	0	0	0
t_4	0	1	1	1	0	0
t_5	0	0	1	0	1	0

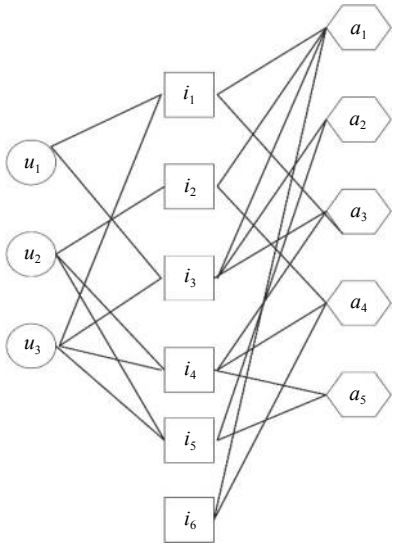

图 1　用户–项目–特性信息

Fig. 1　User-item-attribute information

图 1 中：u 代表用户；i 代表项目；a 代表特性。如果用户 u 评价项目 i，则使用一条直线连接用户 u 和项目 i；同理，如果项目 i 拥有特性 a，则连接项目 i 和特性 a。继续以用户 u_1 为研究目标，根据图 1 可知，项目 i_1 拥有特性 a_1 和 a_3，而有 3 个项目共同拥有特性 a_1，同样有 3 个项目共同拥有 a_3，结合式(10)计算项目 i_1 的项目特性权重为

$$S_a^i(u_1)=S_{a_1}\times\frac{1}{4}+S_{a_2}\times\frac{1}{3}=\left(\frac{1\times1}{2}+\frac{0\times1}{2}+\frac{1\times1}{3}+\frac{0\times1}{2}\right)\times\frac{1}{4}+\left(\frac{1\times1}{2}+\frac{1\times1}{3}+\frac{0\times1}{3}\right)\times\frac{1}{3}=0.486\,1$$

同理，使用式(9)计算项目 i_1 的项目标签关系。然后，根据式(14)可以得到最终的用户间的相似性，如表5所示。

表5 用户相似值

Tab.5 Values of the user similarity

用户	u_1	u_2	u_3
u_1	–	0.013 0	0.032 6
u_2	0.013 0	–	0.050 4
u_3	0.026 7	0.044 5	–

由表5可以看出，用户 u_1 和 u_2 没有共同评分项，但仍能根据其他信息得到相似度，说明本文方法并不依赖共同评分项，并且用户间的相似影响基本不同。最后，根据式(15)计算用户对未评级项目的评分，若该项目是新项目，则直接采用用户时间权重信息与新项目特性信息结合得到预测评分值。最后对用户的项目评级进行排序，完成 Top-N 推荐。

基于协同过滤和混合相似性模型的推荐算法描述如表6所示。

表6 RCFHSM算法描述

Tab.6 Description of RCFHSM algorithm

Input	user-item rates matrix, user u, user v, attribute-items matrix, tags-items matrix, user appraisal schedule	
Output	predict values	
算法步骤		功能释义
Step1	$S_1(u_i,v_j)\leftarrow Proximity(u_i,v_j)Significance(u_i,v_j)Singularity(u_i,v_j)$	计算基于不同项目的用户相似度
Step2	$S_{\text{item}}(u_i,v_j)\leftarrow 1/(1+D(i,j))$	计算项目间评分关系作为权重
Step 3	$S_a((u,i),(v,j))\leftarrow S_a^i(u)S_a^j(v)+1/(sumitem+1)$	计算项目特性权重
Step 4	$S_t((u,i),(v,j))\leftarrow S_t^i(u)S_t^j(v)+1/(sumitem+1)$	计算项目标签权重
Step 5	$S_2(u,v)\leftarrow 1/(1+\exp(-\lvert I_u\cap I_v\rvert/\lvert I_u\rvert))$	计算项目标签权重
Step 6	$S_3(u,v)\leftarrow 1\text{-}1/(1+\exp(-\lvert\mu_u\text{-}\mu_v\rvert\lvert\sigma_u\text{-}\sigma_v\rvert))$	计算偏好因子
Step 7	$Sim(u,v)\leftarrow\sum_{i=1}\sum_{j=1}S_1(u_i,v_j)S_{\text{item}}(u_i,v_j)(\alpha S_a((u,i),(v,j))+\beta S_t((u,i),(v,j))$	–
Step 8	$S(u,v)\leftarrow S_2(u,v)S_3(u.v)Sim(u,v)$	计算混合相似度
Step 9	$w_u\leftarrow sum(u\text{ rated }i)\Big/\sum_{i=1}(time_{ui}\text{-}date_i)$	计算用户时间权重
Step 10	If no rated item i is new item Then	判断未评级项目是否是新项目
Step 11	$pre_{ui}\leftarrow r_{\text{mean}}+r_{\max}(S_a^i(u)+w_u)$	计算新项目的预测评分值
Step 12	Else	–
Step 13	$pre_{ui}\leftarrow r_{u\text{med}}+\sum_v S(u,v)(r_{v,}r_{v\text{med}})\Big/\sum_v S(u,v)$	计算一般项目的预测评分值

3 实验结果和分析

3.1 数据集预处理和度量标准

本实验采用的数据集是由美国 Minnesota 大学的 Grouplens 研究小组创建并维护的 MovieLens 数据集[22]，在预处理后，将其划分为互不相交的训练集和测试集，训练集占 80%，测试集占 20%。

3.2 实验结果和分析

本文进行两类实验，首先采用 MAE，F1-measure 方法进行对照实验，验证算法的有效性；然后对数据进行再处理，采用新颖度指标进行对照实验，验证推荐算法对解决新项目冷启动问题的有效性。

实验 1 推荐算法的准确度。

将参数 α 和 β 取多种不同的值进行组合，分别进行准确度实验。由于推荐个数少时准确率较高，因此，首先在推荐个数为 5 的情况下，计算不同参数取值组和近邻集的 K 值影响，对 K 值进行选择，如图2所示。

由图2可以看出，在不同的参数组情况下取 K=20 时准确度最高，因此，限于篇幅，只列出 K=20 时一部分不同推荐个数的准确度对比，如表7所示。

本文进行多组实验后发现，其实参数 α，β 对实验结果的影响并不大，说明算法本身对参数取

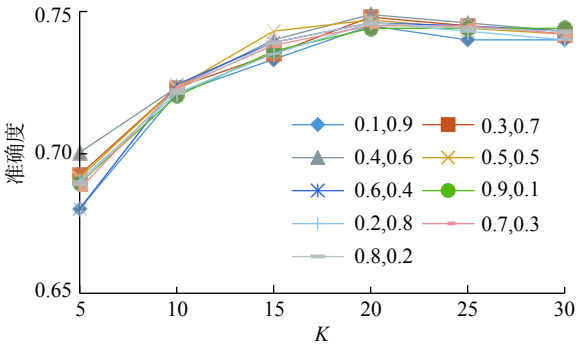

图 2　不同 K 取值的准确度

Fig. 2　Accuracy of different K values

表 7　参数取值

Tab.7　Parameter values

组号	参数取值	推荐个数			
	(α, β)	5	10	20	30
1	0.1, 0.9	0.745	0.690	0.611	0.525
2	0.3, 0.7	0.748	0.690	0.611	0.527
4	0.4, 0.6	0.749	0.691	0.613	0.529
5	0.5, 0.5	0.747	0.692	0.613	0.526
6	0.6, 0.4	0.746	0.692	0.612	0.526
7	0.9, 0.1	0.744	0.897	0.610	0.525

值依赖性不大，但仍选取 α=0.4，β=0.6，此时，算法在此数据集中准确率相对较好。将本文算法（简称 RCFHSM）与文献[16]中的基于用户推荐的 UPCC 算法、基于物品推荐的 IPCC 算法和混合用户相似模型（以下简称 HUSM）算法进行 MAE，F1-measure 对比，结果如图 3 和图 4 所示。

观察两图可知，本文提出的基于协同过滤和混合相似模型的推荐算法在评估方法 MAE，F1-measure 上都优于其他推荐算法。这是因为相比其他算法，本文在累积所有评分项来解决共同评分项较少所导致的数据稀疏问题时，同时加入了特

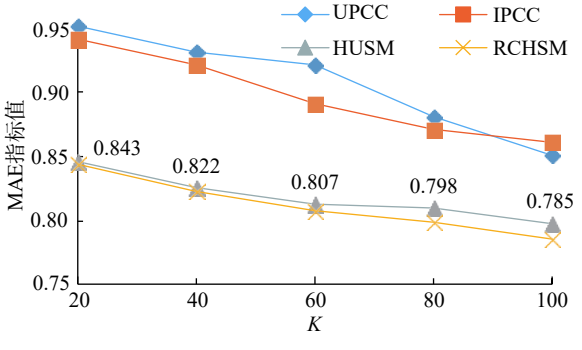

图 3　MAE 对比

Fig. 3　MAE for different algorithms

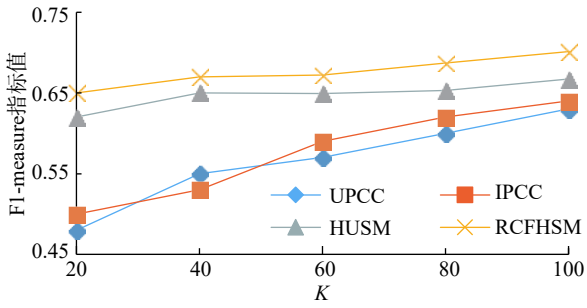

图 4　F1-measure 对比

Fig. 4　F1-measure for different algorithms

性权重信息和标签权重信息，提高了推荐的效果。

实验 2　解决新项目冷启动问题。

原数据集中不存在新项目，即用户未进行评级的项目，因此对原数据集进行再处理，在测试集中随机抽取 300，500，1 000 个项目依次作为新项目，训练集中对应的 300，500，1 000 个项目的评分信息及标签信息依次设为 0。使用处理后的数据集对本文算法进行准确度和新颖度实验，因为前面的几个算法更多基于评分计算，在解决项目冷启动方面效果并不好，为此加入文献[18]的 CUTATime 推荐算法结果进行对比。

从表 8 和表 9 中可以看出，本文算法的新颖度优于 CUTATime 算法。这是因为在计算新颖度时

表 8　CUTATime 算法准确度和新颖度

Tab.8　Accuracy and novelty of CUTATime algorithm

推荐个数	300		500		1 000	
	准确度	新颖度	准确度	新颖度	准确度	新颖度
5	0.709	0.811	0.664	0.777	0.624	0.824
10	0.617	0.734	0.576	0.705	0.518	0.763
15	0.560	0.703	0.525	0.685	0.466	0.731
20	0.517	0.688	0.488	0.670	0.427	0.711
25	0.458	0.664	0.430	0.658	0.375	0.696

表 9　RCFHSM 算法准确度和新颖度

Tab.9　Accuracy and novelty of RCFHSM algorithm

推荐个数	300		500		1 000	
	准确度	新颖度	准确度	新颖度	准确度	新颖度
5	0.712	0.824	0.673	0.801	0.633	0.811
10	0.633	0.763	0.597	0.724	0.525	0.764
15	0.578	0.734	0.578	0.688	0.472	0.733
20	0.529	0.703	0.525	0.679	0.430	0.724
25	0.486	0.691	0.470	0.649	0.405	0.697

加入了用户评分均值，从而提高了新项目的推荐比例，使新项目能够得到更多的推荐。

4 结 论

针对推荐算法一直以来的数据稀疏和冷启动问题，提出一种基于协同过滤和混合相似性模型的推荐算法。该算法通过计算用户在不同项目间的相似性来解决用户间共同评分项较少所带来的数据稀疏问题；并将项目特性权重和用户时间权重信息相结合用于解决新项目冷启动问题。在此基础上进行了对比实验，实验结果表明，本文提出的混合相似性模型合理可行，有较高的准确度和新颖度，并能有效地解决数据稀疏和冷启动问题。

参考文献：

[1] 王嵘冰, 徐红艳, 冯勇, 等. 融合似然比相似度的协同过滤推荐算法研究[J]. 小型微型计算机系统, 2018, 39(7): 1478–1481.

[2] 李平, 张路遥, 曹霞, 等. 基于潜在主题的混合上下文推荐算法[J]. 电子与信息学报, 2018, 40(4): 957–963.

[3] 陈平华, 陈传瑜, 洪英汉. 一种结合关联规则的协同过滤推荐算法[J]. 小型微型计算机系统, 2016, 37(2): 287–292.

[4] 杨武, 唐瑞, 卢玲. 基于内容的推荐与协同过滤融合的新闻推荐方法[J]. 计算机应用, 2016, 36(2): 414–418.

[5] 尹祎, 冯丹, 施展. 一种基于效用的个性化文章推荐方法[J]. 计算机学报, 2017, 40(12): 2797–2811.

[6] 陶永才, 李俊艳, 石磊, 等. 基于地理位置的个性化新闻混合推荐研究[J]. 小型微型计算机系统, 2016, 37(5): 943–947.

[7] 张敏, 丁弼原, 马为之, 等. 基于深度学习加强的混合推荐方法[J]. 清华大学学报(自然科学版), 2017, 57(10): 1014–1021.

[8] 伊华伟, 张付志, 巢进波. 基于模糊核聚类和支持向量机的鲁棒协同推荐算法[J]. 电子与信息学报, 2017, 39(8): 1942–1949.

[9] 冷亚军, 陆青, 梁昌勇. 协同过滤推荐技术综述[J]. 模式识别与人工智能, 2014, 27(8): 720–734.

[10] 王成, 朱志刚, 张玉侠, 等. 基于用户的协同过滤算法的推荐效率和个性化改进[J]. 小型微型计算机系统, 2016, 37(3): 428–432.

[11] 王锦坤, 姜元春, 孙见山, 等. 考虑用户活跃度和项目流行度的基于项目最近邻的协同过滤算法[J]. 计算机科学, 2016, 43(12): 158–162.

[12] LEE J, LEE D, LEE Y C, et al. Improving the accuracy of top-N recommendation using a preference model[J]. Information Sciences, 2016, 348: 290–304.

[13] 郑伟, 李博涵, 王雅楠, 等. 融合偏好交互的组推荐算法模型[J]. 小型微型计算机系统, 2018, 39(2): 372–378.

[14] ZHANG J, LIN Y J, LIN M L, et al. An effective collaborative filtering algorithm based on user preference clustering[J]. Applied Intelligence, 2016, 45(2): 230–240.

[15] CHEN H, LI Z K, HU W. An improved collaborative recommendation algorithm based on optimized user similarity[J]. Journal of Supercomputing, 2016, 72(7): 2565–2578.

[16] WANG Y, DENG J Z, GAO J, et al. A hybrid user similarity model for collaborative filtering[J]. Information Sciences, 2017, 418–419: 102–118.

[17] 黄璐, 林川杰, 何军, 等. 融合主题模型和协同过滤的多样化移动应用推荐[J]. 软件学报, 2017, 28(3): 708–720.

[18] 于洪, 李俊华. 一种解决新项目冷启动问题的推荐算法[J]. 软件学报, 2015, 26(6): 1395–1408.

[19] 郑修猛, 陈福才, 吴奇, 等. 一种利用多群组智慧的协同推荐算法[J]. 西安交通大学学报, 2016, 50(10): 118–124.

[20] PARK Y, PARK S, JUNG W, et al. Reversed CF: a fast collaborative filtering algorithm using a *K*-nearest neighbor graph[J]. Expert Systems with Applications, 2015, 42(8): 4022–4028.

[21] YI M. Collaborative filtering[M]//SAMMUT C, WEBB G I. Encyclopedia of Machine Learning and Data Mining. 2nd ed. Boston, MA: Springer, 2017.

[22] MovieLens data sets[DB/OL]. [2019-04-20]. http://files.grouplens.org/datasets/hetrec2011/hetrec2011-movielens-2k-v2.zip.

（编辑：丁红艺）