计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(9):192−199 [doi: 10.15888/j.cnki.csa.008115]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

结合用户主观偏好与项目属性扩充的推荐算法①

钟耀亿, 丁晓剑, 杨　帆

(南京财经大学 信息工程学院, 南京 210046)
通讯作者: 丁晓剑, E-mail: wjswsl@163.com

摘　要: 协同过滤算法是推荐系统中使用广泛的一种算法, 然而传统协同过滤算法仅利用评分信息, 实际场景下会面临相似度计算准确率低, 推荐个性化程度不高的缺陷, 难以满足用户的需求. 针对协同过滤算法的不足, 结合用户主观偏好与项目属性扩充提出一种改进算法, 首先在项目相似度计算上做了两个改进: 引入标签相关度, 依据项目标签相关度来研究项目之间的相似度, 并根据项目历史评分用户的特征构造项目的扩充属性, 可用于从项目受众类型的角度衡量项目相似度; 其次考虑到用户存在主观偏好的情况, 使用支持向量机为每个用户训练标签偏好预测模型, 可用于项目预测评分的修正, 提高推荐的个性化程度和准确度. 基于 MovieLens 数据集的实验结果表明, 所提算法能更准确地计算项目间的相似度, 且能根据用户的个性化偏好得出更精确的预测评分.

关键词: 协同过滤; 相似度; 属性扩充; 个性化偏好; 支持向量机

引用格式: 钟耀亿,丁晓剑,杨帆.结合用户主观偏好与项目属性扩充的推荐算法.计算机系统应用,2021,30(9):192–199. http://www.c-s-a.org.cn/1003-3254/8115.html

Recommendation Algorithm Combined with User Preference and Item Attribute Extension

ZHONG Yao-Yi, DING Xiao-Jian, YANG Fan

(College of Information Engineering, Nanjing University of Finance & Economics, Nanjing 210046, China)

Abstract: Collaborative filtering algorithms are widely used in recommendationsystems. However, traditional collaborative filtering algorithms, which only use scoring information, have the defects of inaccurate similarity calculation and low personalization in actual scenarios and thus fail to meet user needs. For this reason, this study proposes an improved algorithm combined with user preferences and item attribute extension. Firstly, two improvements are made in the calculation of item similarity: Tag correlation is introduced to study the similarity between items; the extended attribute of items constructed according to the characteristics of the users who scored the item scan measure the item similarity in terms of item audience type. Secondly, considering the subjective preferences of users, a support vector machine is adopted to train the preference prediction model for each user, which can help to modify the item prediction score and improve the personalization and accuracy. Experimental results based on MovieLens dataset show that the proposed algorithm can calculate the similarity more accurately between items and get more accurate prediction scores according to users' personalized preferences.

Key words: collaborative filtering; similarity; attribute extension; personalized preference; Support Vector Machine (SVM)

① 基金项目:国家自然科学基金 (62002156);江苏省高等学校自然科学研究面上项目 (19KJB520035);江苏省研究生科研与实践创新计划 (KYCX20_1327)
Foundation item: National Natural Science Foundation of China (62002156); General Program of Natural Science Foundation of Higher Education, Jiangsu Province (19KJB520035); Graduate Research and Practice Innovation Plan of Jiangsu Province (KYCX20_1327)
收稿时间: 2020-12-15; 修改时间: 2021-01-18; 采用时间: 2021-02-08; csa 在线出版时间: 2021-09-02

1 概述

互联网的迅速发展使得网络上每天都产生数量惊人的信息, 其在为用户提供丰富的信息化服务的同时, 也让用户越发难以搜索到满足其个人偏好的有效信息, 进而让用户迷失在信息的海洋中, 这就是互联网时代的“信息过载”问题, 推荐系统技术的出现在一定程度上缓解了互联网数据爆炸式增长带给人们的信息过载问题[1−3], 如今, 推荐系统拥有广泛的实际应用场景, 如商品推荐, 影视推荐, 新闻推荐, 社交推荐等.

基于项目的协同过滤算法因简单易行, 目前是推荐系统广泛使用的一种技术, 其基于“物以类聚”的核心思想, 根据用户对项目的评价信息来计算项目相似性, 再利用相似性信息进行后续的推荐步骤, 但是其在实际应用中表现出了一些缺陷, 首先是该算法非常依赖用户评分矩阵的质量, 只有高密度的评分矩阵才能保证项目相似度计算的准确度, 然而用户评分矩阵往往是稀疏的, 因此计算出的项目相似度就会不准确[4,5], 同时, 基于项目的协同过滤算法也并未深入考虑用户的主观偏好情况, 其往往会向单个用户推荐大众喜好的流行项目[6,7], 因此推荐的个性化程度不高, 不能很好地满足用户需求.

本文针对基于项目的协同过滤算法相似度计算不准确和推荐缺乏个性化这两个问题进一步地做了改进工作, 考虑到传统基于项目的协同过滤算法只利用用户评分数据计算相似度, 比较单一, 因此在原有基于评分的项目相似度计算方法基础上, 额外增加了两个维度的项目相似度计算, 可以更全面地评估项目间的相似度, 此外考虑到用户主观偏好挖掘对推荐算法的重要性, 本文使用支持向量机构建用户标签偏好预测模型, 并用于评分预测公式的修正, 可以提供更准确的评分预测.

2 相关工作

2.1 基于项目的协同过滤算法

以下是基于项目的协同过滤算法的主要流程[8]:

1) 评分矩阵构建

基于项目的协同过滤推荐需要利用用户对项目的评价数据. 用户评价数据可以表示为一个评分矩阵 $R_{M\times N}$, 其中 M 表示用户的数量, N 表示项目的数量, $R_{i,j}$ 表示用户 i 对项目 j 的数值化评分.

2) 项目相似性计算

项目相似性主要根据评分矩阵来计算, 常用的相似性计算方法有皮尔逊相关系数, 余弦相似度, Jaccard 相似度, 这 3 种计算方式分别如式 (1)~式 (3) 所示:

$$SIM(i,j)_{\mathrm{PCC}}=\frac{\sum\limits_{u\in U_{i,j}}(R_{u,i}-\bar{R}_i)\times(R_{u,j}-\bar{R}_j)}{\sqrt{\sum\limits_{u\in U_{i,j}}(R_{u,i}-\bar{R}_i)^2}\times\sqrt{\sum\limits_{u\in U_{i,j}}(R_{u,j}-\bar{R}_j)^2}} \tag{1}$$

$$SIM(i,j)_{\mathrm{COS}}=\frac{R_i\cdot R_j}{\|R_i\|\times\left\|R_j\right\|} \tag{2}$$

$$SIM(i,j)_{\mathrm{Jaccard}}=\frac{\left|U_i\cap U_j\right|}{\left|U_i\cup U_j\right|} \tag{3}$$

其中, $R_{u,i}$ 和 $R_{u,j}$ 分别表示用户 u 对项目 i 和项目 j 的评分, $U_{i,j}$ 表示对项目 i 和项目 j 都评分过的用户集合, $\bar{R}_i$ 和 $\bar{R}_j$ 分别表示项目 i 和项目 j 的平均评分, R_i 和 R_j 分别表示项目 i 和项目 j 的用户评分向量, $\|R_i\|$ 和 $\left\|R_j\right\|$ 分别表示项目 i 和项目 j 评分向量的模, U_i 和 U_j 分别表示对项目 i 和项目 j 有过评价的用户集合. 根据项目间的相似度值, 就可以得到项目相似性矩阵 $S_{N\times N}$, $S_{i,j}$ 表示项目 i 和项目 j 的相似度.

3) 项目邻居选择

给定项目 i, 它的邻居是指与其相似性最高的 k 的项目, 可以从项目相似性矩阵中得到, k 是可选的值, 一般根据实际情况调整.

4) 项目评分预测

对用户未评分的项目, 根据式 (4) 来预测评分:

$$PRED(u,i)=\bar{R}_i+\frac{\sum\limits_{j\in N_i\cap I_u}S_{i,j}*(R_{u,j}-\bar{R}_j)}{\sum\limits_{j\in N_i\cap I_u}S_{i,j}} \tag{4}$$

式 (4) 中, N_i 表示项目 i 的邻居, I_u 表示用户 u 已评价的项目集合, $S_{i,j}$ 表示项目 i 和项目 j 的相似性, $R_{u,j}$ 表示用户 u 对项目 j 的评分, $\bar{R}_i$ 和 $\bar{R}_j$ 为项目 i 和项目 j 的平均评分.

2.2 支持向量机

支持向量机作为一种新兴的机器学习算法, 以其自身在二类分类学习问题上表现出较好的泛化和推广性能, 近年来得到了人工智能和机器学习领域研究者的广泛关注. 基于统计学习理论, 支持向量机的主要目标是借助于核方法来最小化结构风险, 并最终得到支持向量[9].

支持向量机算法的基本理念是, 假设给定一个特征空间上的训练样本数据集 $\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$, 其中, $x_i\in R^n$, $y_i\in\{+1,-1\}$, $i=1,2,\cdots,n$. x_i 为第 i 个样本

数据的特征向量, y_i为x_i对应的类标记, 当$y_i = +1$时, 称x_i为正例; 而当$y_i = -1$时, 称x_i为负例, 支持向量机的目标是在特征空间中找到一个最优的分离超平面, 其能够尽可能正确地将正负实例分到其各自所对应的类中去, 不仅如此, 支持向量机算法还确保两种实例间分类间隔距离的最大化, 以此来降低结构化风险, 获得较优的分类效果.

在本文的研究中, 支持向量机将被用于学习并预测用户对给定项目标签的偏好, 而给定一个项目标签实例, 其到分离超平面的距离将被认为是用户对该项目标签的偏好程度.

3 标签相关度与用户偏好预测

传统基于项目的协同过滤算法依赖用户的评分来计算项目之间的相似度, 受到用户评分矩阵稀疏的影响, 相似度计算的结果往往不够准确, 因此推荐效果有很大的局限性. 在推荐系统平台中, 除了有用户的评分数据可用, 有时项目本身也有一些描述性的属性信息, 例如标签信息, 那么就可以合理地利用这些信息来进一步地挖掘项目之间的联系以及用户对项目的主观偏好.

3.1 标签相关度

标签可以用于概括性地描述项目的内在特征, 比较常见的根据标签信息表征项目的方法是采用独热编码 (one-hot encoding) 来构造项目的标签向量[10], 其将所有的项目标签考虑进来, 如果一个项目有一个给定的标签, 则该项目标签向量中相对应的值将为 1, 否则为 0, 之后可以基于余弦相似度公式, 使用标签向量计算项目之间的相似度, 该方法具有简单易行的优点, 但由于忽略了标签之间的关系, 只对两个标签向量中相对应的标签进行了比较, 因此损失了一些有价值的信息.

在电影推荐中, 一部电影通常有多个标签, 有些标签经常同时出现在一部电影中, 这说明标签之间存在一定的关联, 假如电影 1 有动作标签, 电影 2 有冒险标签, 并且假设动作标签和冒险标签经常同时出现, 如果用标签向量的方式表征电影, 并用余弦相似度计算电影 1 和电影 2 在标签之间的相似度, 那么将得到值为零的相似度, 这意味着系统会认为这两部电影在用户看来是完全不同的, 不幸的是, 对于一个喜欢动作和冒险电影的用户来说, 如果用户曾经对电影 1 打过评分, 但由于电影 1 和电影 2 的相似度为零, 那么电影 2 就没有机会被推荐给那个用户, 但是如果换种方式使用项目标签数据, 具体来说, 可以统计所有不同标签的共现次数, 那么就可以计算出标签之间的相关性, 这样就可以用标签相关性代替标签向量来计算项目之间的标签相似度了, 在前面提到的情况中, 可以利用动作标签和冒险标签之间的相关性来计算两部电影的相似度, 这样得到的电影相似度就不会为零, 而用户可能得到满意的推荐.

3.2 标签相关度计算

$$COR\left(T_i,T_j\right)=\frac{\sum_{n=1}^{k} I_{(T_i \text{ and } T_j)}(m_n)}{\sum_{n=1}^{k} I_{(T_i)}(m_n)} \tag{5}$$

本文根据不同标签的共现次数来计算标签间的相关性, 具体如式 (5) 所示, 其中$COR\left(T_i,T_j\right)$代表标签T_i和标签T_j的相关度, k为数据集中的项目数目, m_n为训练集中的第n个项目, $I_{(T_i \text{ and } T_j)}$和$I_{(T_i)}$是指示函数, 前者判断当前项目是否同时含有标签$T_i$和标签$T_j$, 后者判断当前项目是否含有标签$T_i$, 若为真则函数值为 1, 否则为 0.

3.3 基于标签的项目相似性计算

$$SIMT\left(m_i,m_j\right)=\frac{\sum_{t_{m_i}=1}^{k_{m_i}}\sum_{t_{m_j}=1}^{k_{m_j}} COR\left(T_{t_{m_i}},T_{t_{m_j}}\right)}{k_{m_i}*k_{m_j}} \tag{6}$$

如式 (6) 所示, 可以计算出项目间基于标签的相似性, 其中, $T_{t_{m_i}}$和$T_{t_{m_j}}$分别表示项目m_i的第t_{m_i}个标签和项目m_j的第t_{m_j}个标签, k_{m_i}和k_{m_j}分别表示项目m_i和项目m_j的标签数. 式 (6) 通过式 (5) 对两个项目中的各个标签对计算相关度, 之后再求出两个项目中所有标签对间相关度的均值得出两个项目的标签相似性.

3.4 用户偏好预测

如前文所述, 基于项目的协同过滤算法仅利用用户矩阵来进行推荐, 两个项目的相似性很大程度取决于有多少用户共同评价过, 这就使传统的协同过滤算法偏向于给单个用户推荐大众流行的项目, 但并未探究用户对特定项目标签的主观偏好, 因此给品味独特的用户的推荐就不够个性化, 考虑到传统算法的这一不足, 本文使用支持向量机构建一种可以预测单个用户对项目标签偏好的模型, 以提高推荐的个性化程度和准确度.

由于要训练预测模型, 因此要根据用户历史评分构建数据集, 设用户u_i的历史评价项目集合为$M=\{m_1,m_2,\cdots,m_K\}$, 项目集中任意一个项目m_k包含标签的集合为$T=\{t_1,t_2,\cdots,t_l\}$, l表示项目m_k包含的标签总数,

用户u_i对项目m_k的评分为$R_{i,k}$, 首先需要确定用户对项目的偏好程度, 可以依据评分设定一个用户偏好阈值ϑ, 若$R_{i,k} \geq \vartheta$, 则认为用户u_i对项目m_k有正向偏好, 若$R_{\mathrm{i,k}} < \vartheta$, 则认为用户$u_i$对项目$m_k$有负向偏好, 其次, 需要将用户对项目的偏好映射到用户对标签的偏好上, 本文的做法是构建项目m_k的标签特征向量$x_k = \{x_{k1}, x_{k2}, \cdots, x_{kL}\}$, 将其对应的用户偏好记为$y_k$, L 表示所有标签的总数, x_{ki}表示项目m_k属于标签t_i的程度, 具体计算如式 (7) 所示, 项目标签特征向量记录项目属于各个标签的程度, 若用户对项目m_k有正向偏好, 则将用户对该标签特征向量的偏好y_k设为 1, 否则将y_k设为–1.

$$x_{ki} = \begin{cases} 1, & t_i \in T \\ \dfrac{\sum_{j=1}^{l} COR(t_j, t_i)}{l}, & t_j \in T, t_i \notin T \end{cases} \tag{7}$$

项目标签特征向量包含了所有标签的权值, 这么做的原因和 3.1 节中所述例子类似, 即若仅考虑用户已评价过的标签, 则学习到的预测模型只会在这个用户历史评价标签范围内做出有效预测, 对一个不在该范围内但和用户历史评价标签高度相关的标签就会一直预测为负向偏好, 这显然是不合理的.

对任意一个用户u_i, 其标签偏好数据集为$\{(x_1, y_1), (x_2, y_2), \cdots, (x_K, y_K)\}$, 其中, K 为用户已评价项目的总数, 这样就可以对每个用户训练一个标签偏好预测模型, 预测模型的获得需要求解如式 (8) 所示的优化问题[11]:

$$\min \frac{1}{2}\|\omega\|^2 \ \text{s.t.} \ y_i(\omega \cdot x_i + b) \geq 1, i = 1, 2, \cdots, K \tag{8}$$

式 (8) 中, $\omega \cdot x_i + b$为需要求解的预测模型, 在实际情况中, 一些样本数据往往是非线性可分的, 支持向量机通常会先使用一些映射函数将样本的特征从低维空间映射到高维空间中, 然后再进行后续的优化步骤, 最终的预测模型可由式 (9) 表示.

$$f(x) = \sum_{j=1}^{\mu} \alpha_j y_j \varphi(x) \cdot \varphi\left(x_j\right) + b \tag{9}$$

式 (9) 中, x_j是支持向量, μ为支持向量的个数, α_j是支持向量x_j所对应的拉格朗日乘子, 经映射函数映射后的点积运算$k\left(x, x_j\right) = \varphi(x) \cdot \varphi\left(x_j\right)$被称为核函数.

在预测阶段, 将项目的标签特征向量输入到预测模型$f(x)$中, 若所得结果为正数, 则说明用户可能会对该项目的标签感兴趣, 否则就表示用户可能不感兴趣, 同时结果的大小也反映了用户对项目标签正偏好或负偏好的程度.

4 项目属性扩充

4.1 动机

目前有些研究工作考虑了项目各种属性间的相似度, 并将其与评分相似度相结合, 在一定程度上进一步提高项目相似度计算的准确度[12,13], 然而项目属性信息只能从项目的角度单方面地描述项目的特征, 在电影推荐中, 常识思维认为动画标签的电影是面向低龄用户的, 但是对为有些动画标签电影打过较高评分的用户群体的特征进行分析, 就会发现这些用户不一定是低龄用户, 那么这些动画电影也不一定是面向低龄用户的, 其就不应被和面向低龄用户动画电影等同看待, 更进一步说, 虽然两个项目具有相同的属性信息, 但用户对它们的偏好也有可能是不同的, 具有相同属性信息的项目可能会受到不同用户群体的偏好, 而具有不同属性信息的项目也可能会受到同一用户群体的偏好, 由此可以看出, 分析项目的受众特征信息可以比直接分析项目自身属性信息能更准确地判断项目会受哪类用户偏好, 所以, 不能仅仅根据项目的属性信息来评估两个项目之间的相似性, 因为这可能会导致有偏的推荐, 此外还应该考虑项目的受众类型是否相似.

有研究工作利用了用户特征来提高用户之间相似度计算的精确度[14,15], 但是很少有研究试图利用用户特征来提高项目之间相似度计算的精确度, 本文提出利用受众特征来构造项目的扩充属性, 进而可以从项目受众类型的角度来丰富项目的属性, 是对仅利用项目自身属性计算项目相似度的有效补充, 最终可以更全面地比较项目之间的相似度.

4.2 一个启发性的例子

在给出项目扩充属性的计算方式之前, 将列举一个关于电影推荐的启发性例子.

假设有表 1 所示用户电影评分 (0 分表示没有评分) 和表 2 所示用户特征信息, 若表 1 中所列都不是热门电影, 那么本文推测一部电影的受众在某些方面是相似的, 也就是说如果用户有共同的品味, 那么在其他方面应该也有一定的相似性. 在这个例子中, 从直观上来看, 对电影 1、2 和 5 这 3 部电影有正面评价的用户有两个方面比较相似, 其一是用户的年龄, 其二是用户的职业; 再如电影 3 和电影 4, 通过用户评分和用户特征信息可以看出电影 3 更受女性用户偏好, 电影 4 更受男性用户偏好.

表 1 用户电影评分

用户	电影1	电影2	电影3	电影4	电影5
用户1	5	5	0	5	5
用户2	5	5	4	3	5
用户3	4	4	4	0	0
用户4	2	0	5	0	0
用户5	1	0	0	5	0

表 2 用户特征信息

用户	性别	年龄	职业
用户1	男	小于25	学生
用户2	女	小于25	学生
用户3	女	小于25	作家
用户4	女	25至34	律师
用户5	男	35至44	工程师

通过对数据集中每个项目的用户特征信息进行统计分析, 就可以得出每个项目的受众特征, 然后利用这些受众特征构造项目的扩充属性来表示该项目的受众类型, 便可以进一步从受众类型的角度进行项目相似度计算.

4.3 项目属性扩充

定义用户的特征集合为$\{f_1,f_2,\cdots,f_k\}$, 用户u的特征可以表示为$\{uf_1,uf_2,\cdots,uf_k\}$, 若该用户具备特征$f_i$, 则$uf_i$的值为 1, 否则为 0, 定义项目$m_i$的扩充属性集合为$A_{m_i}=\{Attr_1,Attr_2,\cdots,Attr_k\}$, 项目$m_i$关于用户特征$f_i$的扩充属性值, 即$Attr_i$的大小, 可通过式 (10) 来计算, 式 (11) 对式 (10) 的计算结果进行了归一化处理:

$$Attr_i=\frac{\sum\limits_{u\in Rated} I(uf_i==1)\times R_u}{\sum\limits_{u\in Rated} R_u} \tag{10}$$

$$Normalized\_Attr_i=\frac{Attr_i}{\sum\limits_j Attr_j} \tag{11}$$

式 (10) 中, $Rated$表示对项目有过评价的用户集合, $I(uf_i==1)$是一个指示函数, 若用户u具备特征f_i, 则函数值为 1, 否则为 0, R_u为用户u对该项目的评分. 项目的各个扩充属性值表现为对项目有过评分的用户的特征加权评分 (权值即前文所提指示函数的值) 与项目历史评分总和的比值, 若一个用户对一个项目有较高的评分, 则该用户的特征信息对该项目的扩充属性值就会有较高的贡献, 最终, 项目的扩充属性会记录对项目有较高评分的用户群体的特征信息.

4.4 基于受众类型的项目相似性计算

项目的扩充属性标识项目的受众类型信息, 因此, 如果不同项目的受众类型是相似的, 那么这些项目也可以被认为是相似的, 可用于推荐, 因为它们受到了具有相似特征用户群体的偏好.

通过项目扩充属性计算项目相似性的方法如式 (12):

$$SIMATTR(m_i,m_j)=2-\frac{2}{1+\exp\left(-\sum\limits_k\left|A_{m_i,k}-A_{m_j,k}\right|\right)} \tag{12}$$

式 (12) 中, $A_{m_i,k}$和$A_{m_j,k}$分别表示项目m_i和项目m_j的第k个扩充属性值的大小, 当两个项目的各个扩充属性值基本相同时, 根据式 (12) 计算出的项目相似值就会接近 1, 也就表明两个项目具有相似的受众群体, 当两个项目的各个扩充属性值基本不同时, 根据式 (12) 计算出的项目相似值就会接近 0, 表明两个项目具有不相似的受众群体.

5 项目相似度计算与用户评分预测

5.1 项目相似度计算

在基于评分的项目相似度计算基础上, 本文结合第 3 节和第 4 节描述的标签相似度和扩充属性相似度提出综合相似度. 综合相似度考虑项目在 3 个维度上的相似性, 采用相乘的方式得出, 如式 (13) 所示:

$$SIMCOMP(m_i,m_j)=SIMATTR(m_i,m_j)*SIMT(m_i,m_j)*SIM(m_i,m_j)_{\mathrm{PCC}} \tag{13}$$

5.2 用户评分预测

对用户u未评价过的项目m_i, 并找出k个与项目m_i最相似的项目, 再根据式 (14) 计算用户u对项目m_i的预测评分, 式 (14) 对相似项目集合中的项目计算出经过相似度加权的评分均值, 加上待预测项目的历史评分均值, 最终得到预测评分, 其中$\boldsymbol{N}_i$表示项目m_i的k近邻, $\boldsymbol{M}_U$表示用户u已经评价过的项目集合, R_{u,m_k}表示用户u对项目m_k的评分, $\bar{R}_{m_i}$和$\bar{R}_{m_k}$为项目m_i和m_k的平均分.

$$PRED(u,m_i)=\frac{\sum\limits_{m_k\in \boldsymbol{N}_i\cap \boldsymbol{M}_u} SIMCOMP(m_i,m_k)*(R_{u,m_k}-\bar{R}_{m_k})}{\sum\limits_{m_k\in \boldsymbol{N}_i\cap \boldsymbol{M}_u} SIMCOMP(m_i,m_k)}+\bar{R}_{m_i} \tag{14}$$

虽然式 (14) 使用了项目综合相似度来预测用户评分, 会一定程度提高预测准确度, 但未能直接表达出用

户对项目的主观偏好, 因此结合 3.4 节所述用户标签偏好预测模型, 提出一种用户标签偏好修正因子, 如式 (15) 所示, 能根据用户主观偏好情况, 调整预测评分计算.

$$\beta(f_u(x),x_i)=\frac{1}{1+e^{-f_u(x_i)}} \tag{15}$$

式 (15) 中, $f_u(x)$为用户u的标签偏好预测模型, x_i为待预测项目m_i的标签特征向量, $f_u(x_i)$为用户u对项目m_i的标签偏好度, 若其值为正且越大, 则说明用户对项目标签正向偏好程度越大, β值会趋于 1, 否则β值趋于 0. 将式 (15) 引入到式 (14) 中即得用户偏好修正的预测评分计算法, 如式 (16) 所示.

$$PRED(u,m_i)=\bar{R}_{m_i}+\beta(f_u(x),x_i)*\frac{\sum_{m_k\in N_i\cap M_u}SIM\_COMP(m_i,m_k)*(R_{u,m_k}-\bar{R}_{m_k})}{\sum_{m_k\in N_i\cap M_u}SIM\_COMP(m_i,m_k)} \tag{16}$$

式 (16) 中, 最后一个分项即为协同过滤算法评分预测公式中根据待预测项目与用户历史偏好项目的相似性计算出的预测偏差, 修正因子可以调节预测偏差对最终预测评分计算的作用, 若用户偏好该项目的标签, 则最终预测评分就会受到预测偏差的影响, 否则就会削弱预测偏差带来的影响, 即使待预测项目与用户历史偏好项目的相似度很高, 但若偏好预测模型预测出用户的偏好度很低, 则预测评分也不会很高.

5.3 本文算法步骤

输入: 用户对项目的评分矩阵$R_{M\times N}$, 用户的特征矩阵$F_{M\times K}$, 项目类别矩阵$G_{N\times L}$, 近邻数k, 待预测项目m, 待推荐用户u.

输出: 用户u对项目m的预测评分$\widehat{R}_{u,m}$.

步骤 1. 根据项目类别信息, 根据式 (5) 计算出项目标签的关联度.

步骤 2. 根据式 (6) 计算出数据集中项目间的标签相似度.

步骤 3. 根据项目标签信息和评分信息构建用户标签偏好数据集, 使用支持向量机对数据集中每个用户训练标签偏好预测模型.

步骤 4. 根据用户特征, 对数据集中每个项目按式 (10) 和式 (11) 计算出项目的扩充属性.

步骤 5. 根据式 (12) 计算出数据集中项目间的扩充属性相似性.

步骤 6. 根据式 (13) 计算出项目间的综合相似度.

步骤 7. 对于待预测项目m, 根据步骤 6 得出的项目综合相似度找出与项目m最相似的k个项目作为邻居, 并在邻居中去除掉用户已经评价过的项目, 通过式 (16) 计算出用户u对项目m的预测评分.

6 实验结果与分析

6.1 实验数据集

本文采用在推荐系统研究领域中具有较大知名度的 MovieLens-100 k 数据集进行实验分析[16], 该数据集包含了由明尼苏达大学 GroupLens 研究组从 MovieLens 电影评分网站上收集的 943 个用户对 1652 部电影的 100 000 条评分数据、943 条用户个人信息以及 1652 条电影信息. 本文使用用户个人信息中的性别、年龄和职业信息来构造项目的扩充属性, 并利用电影信息中的标签信息来计算项目标签的关联度以及训练用户标签偏好预测模型. 数据集被随机地分成 80% 的训练数据集和 20% 的测试数据集.

6.2 评估指标

为了评估算法预测的用户评分与实际用户评分之间的差异, 本文采用了均方根误差 (*RMSE*) 作为算法性能评估的指标. *RMSE* 的计算如式 (17) 所示.

$$RMSE=\sqrt{\frac{1}{|TD|}\sum_{u,i\in TD}\left(R_{u,i}-\widehat{R}_{u,i}\right)^2} \tag{17}$$

式中, TD为测试数据集, $R_{u,i}$为测试数据集中用户u对项目i的实际评分, $\widehat{R}_{u,i}$为算法预测出的测试数据集中用户u对项目i的评分.

6.3 用户偏好阈值ϑ的影响

本文根据用户历史评分数据学习并预测用户的标签偏好, 因此合理的用户偏好阈值ϑ的选择就很关键, 其直接影响到预测模型的工作性能, 因此就几种阈值ϑ的选择对算法性能的影响做了实验验证. 由于数据集采用了 1 分至 5 分的评价标准, 所以采用 2 分、3 分和 4 分作为用户正向偏好和负向偏好的分界阈值进行实验, 实验结果如图 1 所示.

由图 1 实验结果可以看出当ϑ取 3 时, 本文算法的评分预测性能表现得最好, 故后续实验取ϑ为 3.

6.4 用户标签偏好修正因子β的有效性

为了验证用户标签偏好修正因子β能够有效地根据用户偏好修正传统评分预测公式的计算结果, 故分

别采用式 (14) 和式 (16) 作为评分预测公式进行了对比实验, 实验结果如图 2 所示.

由图 2 实验结果可以看出, 用户标签偏好修正因子β确实能在一定程度上修正预测评分的误差, 说明了用户偏好的挖掘对推荐算法的重要性.

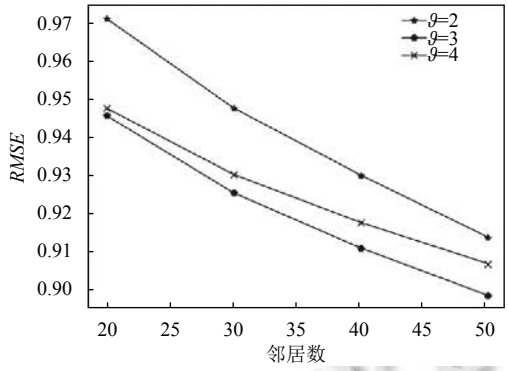

图 1 阈值ϑ对算法性能的影响

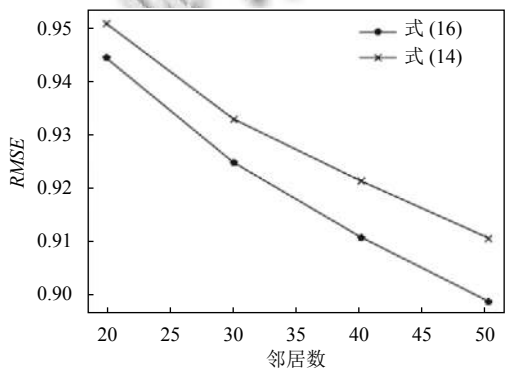

图 2 用户标签偏好修正因子β的有效性

6.5 扩充属性粒度的影响

本文假设扩充属性的粒度对项目相似度计算有一定的影响, 即用户特征划分地越细, 则项目相似度的计算准确度就越高. 为了验证这一假设, 本文根据用户的年龄特征和职业相似性, 通过适当简化原始数据集中的用户特征构造了一个粒度稍低的用户特征集用于实验, 其中, 使用原始数据集中用户特征的算法记为 TP-IAE-1, 使用本文简化用户特征的算法记为 TP-IAE-2. 扩充属性粒度对本文算法性能影响的实验结果如图 3 所示.

由图 3 实验结果可以看出, 两种算法预测误差率变化趋势一致, 但使用原始用户特征的 TP-IAE-1 算法的预测误差率低于使用简化用户特征的 TP-IAE-2 算法, 很显然, 这是因为原始用户特征具有相对更高的粒度, 因此项目受众特征更加丰富, 可以为项目构造更精细的扩充属性, 项目受众类型相似度计算也更加准确.

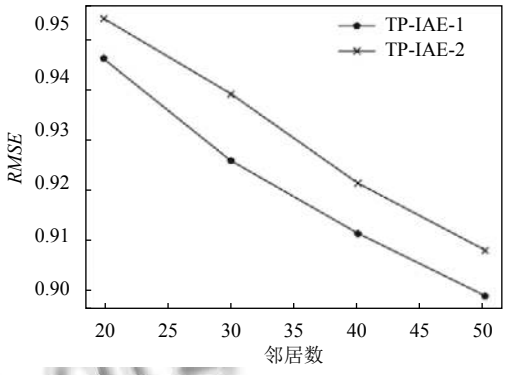

图 3 扩充属性粒度对本文算法性能的影响

6.6 对比试验

为了验证本文算法对传统基于项目的协同过滤算法做出了有效的改进, 以及相较于其他相关算法的性能改善, 故将本文算法 TP-IAE 和 TP-IAE-β (在评分预测阶段, TP-IAE 用式 (14), TP-IAE-β用式 (16) 进行) 与基于项目的协同过滤算法 (IBCF)、文献 [17] 提出的基于内容和标签权重的混合推荐算法 (TW-ContentItem)、文献 [18] 提出的结合项目属性偏好的混合协同过滤算法 (HCF) 以及文献 [19] 提出的基于用户偏好矩阵填充的改进混合推荐算法 (UPR) 做了对比实验, 实验结果如图 4 所示.

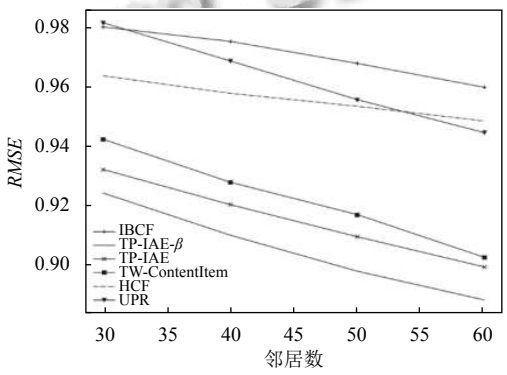

图 4 对比实验结果

从图 4 中可以看出, 本文所提算法和其他 4 种对比算法的预测评分误差率都随着邻居数目的增加而下降, 4 种算法的用户预测评分误差率变化趋势也基本相同, 但在邻居数的变化范围内, 本文算法的预测评分误差率相比其他 4 种算法降低了 2% 至 13%, 显示出良好的推荐效果. 与传统基于项目协同过滤方法相比, 尽

管 HCF 算法在预测效果上有了很大的提高, 因为其不再仅依赖用户的评分信息, 但是, 该方法只是从项目的角度额外地比较项目相似度, 而本文算法不仅考虑了项目的标签属性信息, 另外还考虑了项目的受众特征, 用来判断给定的两个项目是否会得到同一用户群体的偏好, 可以进一步地提高项目相似度计算的准确性, 与 UPR 和 TW-ContentItem 算法的对比结果也显示出本文算法预测效果的提升, 且经用户标签偏好修正因子 β对评分预测公式进行修正后, 预测评分的误差率能够进一步地下降.

7 结束语

本文针对传统基于项目的协同过滤算法面临相似度计算不准确和推荐缺乏个性化这两个问题做了相关的改进工作, 提出的方法结合项目的受众特征信息和项目的标签信息对传统基于项目的协同过滤算法的相似度计算方式做出了有效改进, 不仅可以避免项目相似度计算方法过于片面单一, 而且可以在评分数据稀疏的实际场景下提高项目相似性计算的准确度, 提出的方法还考虑了用户的个性化偏好程度, 以进一步提高推荐效果, 实验结果表明, 本文算法相较传统基于项目的协同过滤算法可以显著地降低预测评分的误差率. 基于项目的协同过滤算法的核心是项目相似性计算, 因此下一步的工作是研究从更多的维度来计算项目的相似性以提高推荐算法的性能.

参考文献

1 Chen JM, Tang Y, Li JG, *et al*. Survey of personalized recommendation algorithms. Journal of South China Normal University (Natural Science Edition), 2014, (5): 8–15.

2 翁小兰, 王志坚. 协同过滤推荐算法研究进展. 计算机工程与应用, 2018, 54(1): 25–31. [doi: 10.3778/j.issn.1002-8331.1710-0081]

3 Su XY, Khoshgoftaar TM. A survey of collaborative filtering techniques. Advances in Artificial Intelligence, 2009, 2009: 421425.

4 Liu HF, Hu Z, Mian A, *et al*. A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-Based Systems, 2014, 56: 156–166. [doi: 10.1016/j.knosys.2013.11.006]

5 Anand D, Bharadwaj KK. Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities. Expert Systems with Applications, 2011, 38(5): 5101–5109. [doi: 10.1016/j.eswa.2010.09.141]

6 张旭, 孙福振, 方春, 等. 引入兴趣稳定性的时间敏感协同过滤算法. 计算机工程与应用, 2018, 54(11): 161–165, 197. [doi: 10.3778/j.issn.1002-8331.1701-0332]

7 Abdollahpouri H. Popularity bias in ranking and recommendation. Proceedings of 2019 AAAI/ACM Conference on AI, Ethics, and Society. Honolulu, HI, USA. 2019. 529–530.

8 于洪, 李俊华. 一种解决新项目冷启动问题的推荐算法. 软件学报, 2015, 26(6): 1395–1408. [doi: 10.13328/j.cnki.jos.004587]

9 祁亨年. 支持向量机及其应用研究综述. 计算机工程, 2004, 30(10): 6–9. [doi: 10.3969/j.issn.1000-3428.2004.10.003]

10 Pham DH, Le AC. Exploiting multiple word embeddings and one-hot character vectors for aspect-based sentiment analysis. International Journal of Approximate Reasoning, 2018, 103: 1–10. [doi: 10.1016/j.ijar.2018.08.003]

11 Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L, *et al*. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 2020, 408: 189–215. [doi: 10.1016/j.neucom.2019.10.118]

12 Zhu L, Hu QH, Zhao L, *et al*. Collaborative filtering algorithm based on rating preference and item attributes. Computer Science, 2020, 47(4): 67–73.

13 王卫平, 王金辉. 基于 Tag 和协同过滤的混合推荐方法. 计算机工程, 2011, 37(14): 34–35, 38. [doi: 10.3969/j.issn.1000-3428.2011.14.009]

14 孙龙菲, 黄梦醒. 综合用户特征和项目属性的协作过滤推荐算法. 计算机应用研究, 2014, 31(2): 384–387. [doi: 10.3969/j.issn.1001-3695.2014.02.015]

15 刘发升, 洪营. 基于用户特征属性和云模型的协同过滤推荐算法. 计算机工程与科学, 2014, 36(6): 1172–1176. [doi: 10.3969/j.issn.1007-130X.2014.06.028]

16 Harper FM, Konstan JA. The MovieLens datasets: History and context. ACM Transactions on Interactive Intelligent Systems, 2015, 5(4): 19.

17 刘宇, 朱文浩. 基于内容和标签权重的混合推荐算法. 计算机与数字工程, 2020, 48(4): 773–777. [doi: 10.3969/j.issn.1672-9722.2020.04.008]

18 于波, 陈庚午, 王爱玲, 等. 一种结合项目属性的混合推荐算法. 计算机系统应用, 2017, 26(1): 147–151. [doi: 10.15888/j.cnki.csa.005490]

19 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. 计算机工程与设计, 2020, 41(10): 2784–2790.