doi:10.14132/j.cnki.1673-5439.2021.01.013

基于用户特征和评分的精准推荐策略研究

傅金京,李玲娟

(南京邮电大学 计算机学院,江苏 南京 210023)

摘要:个性化推荐系统是帮助用户发现内容,克服信息过载的重要工具。为了提高推荐算法的准确率和效率,综合协同过滤推荐算法和 K-means 聚类算法,设计了一种基于用户特征和评分的精准推荐策略。该策略一方面针对新用户冷启动问题,引入 K-means 聚类算法对全体用户特征进行聚类,将新用户所属类中其他用户喜好的物品中的 Top N 个推荐给新用户;另一方面根据物品数和用户数的大小关系,或者不同推荐算法所得 F1 值的大小关系,来决定选择将哪种推荐算法产生的结果推荐给老用户。在 Movielens 和 FilmTrust 数据集上的实验结果表明,这种基于用户特征和评分的精准推荐策略能够有效地针对新用户和老用户做出准确的最佳推荐。

关键词:协同过滤推荐;用户冷启动;K-means 聚类算法

中图分类号:TP311 **文献标志码**:A **文章编号**:1673-5439(2021)01-0107-08

Accurate recommendation strategy based on user characteristics and ratings

FU Jinjing, LI Lingjuan

(School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: A personalized recommendation system is an important tool to help users discover the content and overcome the information overload. To improve the accuracy and the efficiency of the recommendation algorithm, a collaborative filtering recommendation algorithm and a K-means clustering algorithm are combined to design an accurate recommendation strategy based on user characteristics and ratings. Aiming at the cold start problem of new users, K-means clustering algorithm is adopted to cluster all user characteristics, and the Top N items preferred by other users in the class to which the new user belongs are recommended to the new users. According to the size relationship between the number of items and the number of users, or the size relationship between the F1 values obtained by different recommendation algorithms, results with the appropriate recommendation algorithm are chosen for the existing users to solve the problem, that is, a fixed recommendation algorithm is not used properly for existing users. Experimental results on Movielens and FilmTrust datasets show that the strategy is effective in making accurate and best recommendations for both new and existing users.

Keywords: collaborative filtering recommendation algorithm; user cold-start; K-means clustering algorithm

收稿日期:2020-06-16;**修回日期**:2020-07-25 **本刊网址**:http://nyzr.njupt.edu.cn

基金项目:国家自然科学基金(61572260,61872196)资助项目

作者简介:傅金京,女,硕士研究生;李玲娟(通信作者),女,博士,教授,lilj@njupt.edu.cn

引用本文:傅金京,李玲娟.基于用户特征和评分的精准推荐策略研究[J].南京邮电大学学报(自然科学版),2021,41(1):107-114.

个性化推荐系统可以收集并分析用户在使用互联网过程中所产生的用户个人信息、浏览历史、购买记录以及评价等信息,然后基于这些信息从海量数据中挖掘出用户独特的需求和兴趣倾向,再将用户感兴趣的信息或商品推荐给用户[1-2]。近年来,科研人员提出了许多适用于不同场景的推荐算法,例如基于内容的推荐[3]、基于内存的协同过滤推荐[4]、基于关联规则的推荐[5]等。其中应用最广的是基于内存的协同过滤推荐算法,此类算法又有基于用户的协同过滤推荐算法 UserCF 和基于物品的协同过滤推荐算法 ItemCF。前者适用于用户数少的场合,后者适用于物品数明显小于用户数的场合。因此一个推荐系统只采用固定的一种推荐算法是不合适的,需要根据具体情况灵活选用合适的推荐算法来产生更有针对性的推荐结果。

推荐系统通过评估目标物品被用户喜欢的程度来决定是否将其列入推荐列表,而要完成用户偏好的评估,需要一定数据量的支持。但是,在实际应用中,推荐系统会有大量的新用户或者新物品,导致了数据的稀疏性很大,此时推荐系统没有足够的信息进行偏好的评估与分析,这种现象称为冷启动[6-7]。

为了解决以上问题,本文设计了一种基于用户特征和评分的精准推荐策略(Accurate Recommendation Strategy based on User Characteristics and Ratings,UCRAR)。该策略一方面采用 K-means 聚类算法对全体用户特征进行聚类,将新用户所属类中其他用户喜好物品的 Top N 个推荐给新用户,另一方面根据物品数和用户数的大小关系,或者不同推荐算法所得 F1 值的大小关系,来决定选择用哪种推荐算法产生的结果推荐给老用户。这种策略引入用户特征解决了用户冷启动问题,并且能够根据具体情况灵活选用合适的推荐算法来产生更有针对性的推荐结果。

1 相关知识

1.1 UserCF 与 ItemCF

协同过滤(Collaborative Filtering,CF)是个性化推荐系统中应用最为广泛的技术之一。该算法表明具有相似兴趣的用户可能喜欢相似的物品,或者用户可能对相似的物品表现出相似的偏好,这是一种具有集体智慧的决策方法[8]。

(1) 基于用户的协同过滤推荐算法 UserCF

UserCF 算法的基本思想是:使用用户相似度计算方法,为目标用户找到最近邻居用户集;然后,提取最近邻居用户集中用户所喜欢的物品,并除去目标用户历史记录中感兴趣的物品;最后根据目标用户对这些物品感兴趣的程度排序获得最终结果,选择并推荐给用户。

在基于用户的协同过滤算法中,最重要的部分是计算用户的相似度,常用的用户相似度计算方法有皮尔逊相关系数[9]、欧几里得距离[10]、余弦相似度等[11]。

皮尔逊相关系数计算方法如式(1)所示。

$$w_{uv}=\frac{\sum_{i\in I}(r_{ui}-\bar{r}_u)\cdot(r_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in I}(r_{ui}-\bar{r}_u)^2\sum_{i\in I}(r_{vi}-\bar{r}_v)^2}} \tag{1}$$

其中,i 表示物品,I 表示用户 u 和用户 v 共同评分过的物品集合,r_{ui} 为用户 u 对物品 i 的评分,$\bar{r}_u$ 为用户 u 对他评过分的所有物品的平均评分。

余弦相似度的计算方法如式(2) 所示。

$$w_{uv}=\frac{\sum_{i\in I}r_{ui}\cdot r_{vi}}{\sqrt{\sum_{i\in I}r_{ui}^2}\sqrt{\sum_{i\in I}r_{vi}^2}} \tag{2}$$

在 UserCF 算法中,另一个重要的部分是预测用户对未评分物品的评分。计算方法如式(3)所示。

$$\hat{r}_{ui}=\bar{r}_u+\frac{\sum_{v\in S(u,K)\cap N(i)}w_{uv}(r_{vi}-\bar{r}_v)}{\sum_{v\in S(u,K)\cap N(i)}|w_{uv}|} \tag{3}$$

其中,$S(u,K)$ 为和用户 u 兴趣最相似的 K 个用户的集合,$N(i)$ 为对物品 i 评过分的用户集合。

(2) 基于物品的协同过滤推荐算法 ItemCF

ItemCF 算法的基本思想是:通过使用物品相似度计算方法,结合目标用户的历史行为,为目标用户找到与其曾经喜好过的物品最相似的其他物品;然后,根据目标用户对这些物品感兴趣的程度排序,选择合适的物品推荐给用户。

在 ItemCF 算法中,物品相似度计算公式可以采用式(1)或式(2)进行调整运算,也可以运用关联规则方法计算,如式(4)所示。

$$w_{ij}=\frac{|N(i)\cap N(j)|}{\sqrt{|N(i)||N(j)|}} \tag{4}$$

其中,$|N(i)|$ 为喜欢物品 i 的用户数,$|N(i)\cap N(j)|$ 为同时喜欢物品 i 和物品 j 的用户数。

用户偏好计算方法如式(5)所示。

$$p_{uj}=\sum_{i\in N(u)\cap S(j,K)}w_{ji}r_{ui} \tag{5}$$

其中,$N(u)$为用户喜欢物品的集合,$S(j,K)$为和物品 j 最相似的 K 个物品的集合,w_{ji}为物品 j 和物品 i 的相似度。

1.2 K-means 聚类

聚类算法是机器学习中非监督学习的重要研究领域。聚类算法的基本目标是将相似的数据样本分组为一个类。传统的聚类算法可分为:划分聚类、层次聚类、基于密度的聚类、基于模型的聚类和基于网格的聚类[12-13]。划分聚类中最常用的是 K-means 聚类算法[14]。

K-means 聚类算法的基本思想是,首先任意选取 K 个数据作为初始聚类中心;然后进入迭代过程,每次迭代包含两个步骤:(1) 分配步骤,即将其余各个数据划归到距离它最近的那个聚类中心所处的簇类中;(2) 更新步骤,即重新计算每个簇类中所有数据的平均值,找到新的聚类中心。当聚类中心不再变化时,聚类完成。该算法具体步骤[15]如下:

(1) 给定大小为 n 的数据集,令 $I=1$,随机选择 K 个初始聚类中心 $Z_j(I)$,$j=1,2,\cdots,K$。

(2) 计算每个数据对象与聚类中心的距离 $D[x_i,Z_j(I)]$,$i=1,2,\cdots,n$;$j=1,2,\cdots,K$,若满足 $D[x_i,Z_j(I)]=\min\{D[x_i,Z_j(I)],i=1,2,\cdots,n\}$,那么 $x_i\in w_k$。

(3) 计算 K 个新聚类中心

$$Z_j(I+1)=\frac{1}{n}\sum_{i=1}^{n_j}x_i^{(j)},j=1,2,\cdots,K \tag{6}$$

以及误差平方和准则函数 J_c 的值

$$J_c(I+1)=\sum_{j=1}^{K}\sum_{k=1}^{n_j}\|x_k^{(j)}-Z_j(I+1)\|^2 \tag{7}$$

(4) 判断。如果 $|J_c(I+1)-J_c(I)|<\xi$,算法结束;反之,$I=I+1$,则重新返回第(2)步执行。

1.3 Top N 推荐评价指标

Top N 推荐常用的评价指标有准确率(Precision)、召回率(Recall)和 F1 值。其中,F1 值是综合了前两者指标的评估指标。

(1) 准确率(Precision)

准确率(Precision)计算方法如式(8)所示。

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

其中,$R(u)$为根据用户在训练集上的行为给用户做出的推荐列表,$T(u)$为用户在测试集上的行为列表。

(2) 召回率(Recall)

召回率(Recall)计算方法如式(9)所示。

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

(3) F1 值

F1 值计算方法如式(10)所示。

$$\text{F1}=\frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}} \tag{10}$$

2 基于用户特征和评分的精准推荐策略的设计与分析

与其他推荐算法相比,协同过滤推荐算法的优势在于它只依赖用户的行为数据,不依赖于具体用户及标的物相关的数据就可以做推荐。但是,如何在有新用户、新物品等情况下保证推荐精度,以及如何根据具体情况选择合适的算法来提高推荐精度,都是需要解决的问题。本文设计的基于用户特征和评分的精准推荐策略 UCRAR 总体上包括:新用户冷启动问题的解决方法、推荐算法及结果的选择策略。

2.1 新用户冷启动问题的解决办法

本文主要引入用户特征、K-means 聚类算法来解决用户冷启动问题。研究发现要利用 K-means 聚类算法来寻找与新用户特征最为相似的老用户,需先对所有用户特征数据进行特征归一化处理,然后再对新老用户特征进行聚类,接着依次计算新用户所属类中其他用户与新用户的相似度,把它作为用户可能喜好物品的权重,最后从中选取 Top N 个推荐给新用户。

对用户特征数据进行归一化处理的过程如下:首先提取用户特征信息 $\mathbf{Ftr}=[\text{age},\text{gender},\text{occupation}]$,构建用户特征矩阵 $\boldsymbol{U}\times\mathbf{Ftr}$;然后使用 LabelEncoder 方法对分类型特征值进行编码,即将离散型的数据转换成 0 到 $n-1$ 之间的数,这里 n 是一个列表的不同取值的个数,可以认为是某个特征的所有不同取值的个数;接着,对离散型特征进行 one-hot 编码,其方法是用 N 位状态寄存器编码 N 个状态,每个状态都有独立的寄存器位,保证每个样本中的每个特征只有一位处于状态 1,其他都是 0;最后,将用户的所有特征拼接成一个用户特征向量,例如对于“年龄为 18 岁,性别为男,职业是学生”的用户 $\mathbf{Ftr}_1=[18,\text{M},\text{student}]$,经过如上处理后得到该用户特征向量为[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1]。

2.2 推荐算法及结果的选择策略

鉴于 UserCF 算法和 ItemCF 算法的使用特性,即前者适用于用户数少的场合而后者适用于物品数明显小于用户数的场合,本文把用户特征作为算法的输入,生成针对用户灵活选用合适推荐算法的多个判断条件,最终来产生更有针对性的推荐结果。

在 UCRAR 策略中,首先比较用户数和物品数的大小关系,若用户数远大于物品数,则直接选用 ItemCF 算法;若物品数远大于用户数,则直接选用 UserCF 算法;若用户数和物品数较为接近,则系统把这两个推荐算法都作用于该用户并生成推荐列表集,每个列表和用户真实评价列表比较,并依据评价指标 F1 值决定当前用户所适用的推荐算法,若两者的推荐评价指标 F1 值结果不分上下,则将两者的推荐列表集结合在一起并根据权重与评分相乘结果排序,选择值最高的 N 个物品推荐给用户。

2.3 基于用户特征和评分的精准推荐策略总体流程

UCRAR 策略的总体流程如图 1 所示。

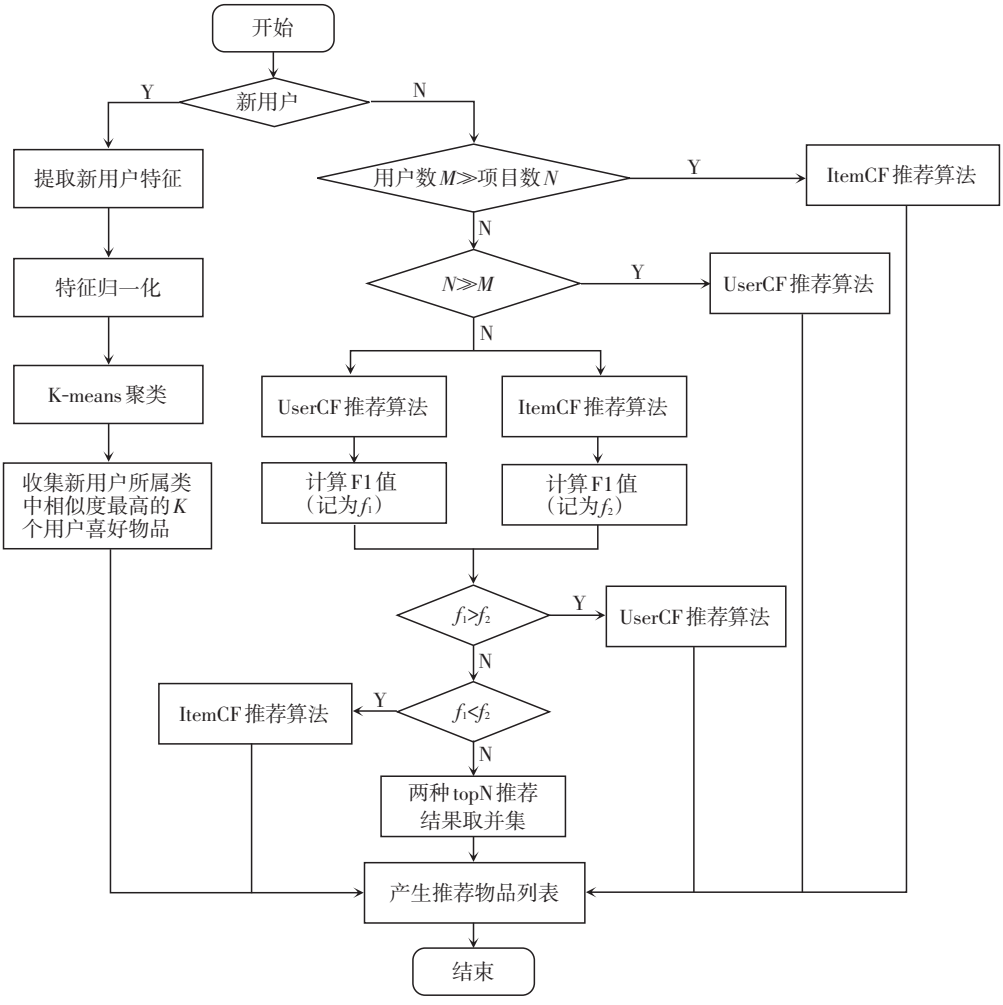


图 1 UCRAR 策略的总体流程

总输入是用户信息,总输出是推荐物品列表。

在对任意一个用户进行推荐之前,首先判断该用户是否为新用户,然后采取不同的推荐策略,直到获得最终推荐结果。

针对新用户做推荐的具体步骤如下:

Step1:提取新用户的用户特征信息 $\mathbf{Ftr}_{\mathbf{new}}$ = [age,gender,occupation]。

Step2:使用 LabelEncoder 方法和 one-hot 编码对新老用户特征进行特征归一化处理,利用式(6)和式(7)得到最终的 K 个聚类中心 $Z_j(I)$,$j=1,2,\cdots,K$。此步骤还可以得到一个关于每个数据样本所属最终聚类中心的列表 label_pred。

Step3:从列表 label_pred 中找到与新用户属于同一个聚类中心的其他用户。计算这些用户与新用户的相似度,选择相似度最高的前 K 个用户。此步骤可生成由与新用户相似度最高的 K 个用户及对

应相似度组成的字典nerb。

Step4:收集这 K 个用户所有喜好的物品,然后把字典 nerb 中对应用户的相似度作为该用户喜好物品的权重,据此计算新用户对所有推荐物品的喜好程度,从中选择喜好程度最高的 N 个物品,产生推荐物品列表,推荐给用户。

该部分伪代码如下。

算法 1:针对新用户做推荐的算法

input:新用户特征 $\mathbf{Ftr}_{new}$,老用户特征集 $U_{old}\times\mathbf{Ftr}$,用户-物品评分矩阵 $\boldsymbol{R}$

output:新用户推荐物品列表 $itemList_{new}$

for each $U_i \in U_{old}$ do
  extract **Ftr** to U_{old}
end for
add $\mathbf{Ftr}_{new}$ into $U_{old}\times\mathbf{Ftr}$
set Ftr_Nlz through LabelEncoder and One-hot Encoding
initial clustering center $Z_j(I)$, $j=1,2,\cdots,K$
repeat
  for each $Ftr\_Nlz_i \in Ftr\_Nlz$
    for each clustering center $Z_j(I) \in Z(I)$
      calculate $D[Ftr\_Nlz_i, Z_j(I)]$
      if $D[Ftr\_Nlz_i, Z_j(I)]$ min
        $Ftr\_Nlz_i \in w_k$
update all clustering center

$$Z_j(I+1) = \frac{1}{n}\sum_{i=1}^{n_j} Ftr\_Nlz_i^{(j)}, j = 1,2,\cdots,K$$

calculate sum of squared errors

$$J_c(I+1) = \sum_{j=1}^{K}\sum_{k=1}^{n_j} \| Ftr\_Nlz_k^{(j)} - Z_j(I+1) \|^2$$

if $|J_c(I+1)-J_c(I)|<\xi$
  get label_pred
  end repeat
else
  repeat
for i in range(len(label_pred)−1)
  if $label\_pred_i = label\_pred_{-1}$
    calculate $sim(Ftr\_Nlz_{new}, Ftr\_Nlz_i)$
  end if
end for
set nerb through comparing $sim(Ftr\_Nlz_{new}, Ftr\_Nlz_i)$
for each $nerb_i \in nerb$
  for $movie_j \in R(nerb_i)$
    $rank_j += nerb_i$('sim')
  end for
end for
set $itemList_{new}$ through comparing $rank_j$

针对老用户做推荐的具体步骤如下。

Step1:比较用户数 M 和物品数 N 的大小关系。

Step2:若用户数远大于物品数,则直接选用 ItemCF 算法做出推荐,获得推荐物品列表,推荐结束。

Step3:若物品数远大于用户数,则直接选用 UserCF 算法做出推荐,获得推荐物品列表,推荐结束。

Step4:若用户数和物品数较为接近,则系统把 UserCF 算法和 ItemCF 算法都作用于目标用户并生成推荐物品列表。

Step5:将这两个列表与用户真实评价物品列表进行比对,并以推荐评价指标 F1 值的高低抉择当前用户所适用的推荐算法。选用 F1 值较大的推荐结果,产生推荐物品列表;若两者的推荐评价指标 F1 值结果相近,则将两者的推荐物品列表集结合在一起,并根据权重与评分相乘结果排序,选择值最高的 N 个物品推荐给用户,推荐结束。

该部分伪代码如下。

算法 2:针对老用户做推荐的算法

input:目标用户 id,用户-物品评分矩阵 $\boldsymbol{R}$

output:老用户推荐物品列表 $itemList_{old}$

if $M \gg N$
  set $itemList_{old}$ through ItemCF
else if $M \ll N$
  set $itemList_{old}$ through UserCF
else
  set F1 as f_1 through UserCF
  set F1 as f_2 through ItemCF
  if $f_1 > f_2$
    set $itemList_{old}$ through UserCF
  else if $f_1 < f_2$
    set $itemList_{old}$ through ItemCF
  else
    set rec_item through UserCF and ItemCF
  end if
  set $itemList_{old}$ through comparing rec_item('sim')
end if

2.4　UCRAR 策略复杂度分析

UCRAR 策略在针对新用户做推荐时,引入 K-means 聚类算法对全体用户特征进行聚类,该阶段的时间复杂度为 $O(nki)$,其中 n 为用户数,k 为簇集数,i 为 K-means 聚类算法迭代次数。然后,将新用户所属类中其他用户喜好的物品中的 Top N 个推荐给新用户,该阶段的时间复杂度为 $O(mN)$,其中 m 为新用户所属类中的用户数,N 为推荐物品数,并且新用户所属类中的用户数远小于整个用户集数量,即 $m \ll n$。UCRAR 策略在针对老用户做推荐时,主要是根据物品数和用户数的大小关系,或者不同推荐算法所得 F1 值的大小关系,来决定选择将哪种推荐算法产生的结果推荐给老用户,其时间复杂度近

似为 $O(n^2)$,n 为用户数。综上所述,UCRAR 策略具有较低的时间复杂度,在一定程度上提高了推荐系统的实时响应能力。

3 实验及结果分析

为了验证 UCRAR 策略的性能,本文分别基于 Movielens 数据集、FilmTrust 数据集进行了实验。为保证算法的准确性,对上述两个数据集分别做了 50 次测试,求取相应测试结果的准确率(Precision)、召回率(Recall)和 F1 值的平均值,并同经典的 UserCF 算法和 ItemCF 算法的推荐结果进行对比。

3.1 实验数据集

Movielens 数据集[16]是由明尼苏达大学 GroupLens 研究组提供的。它是一个关于电影评分的数据集,里面包含用户对电影的评分信息、用户信息、电影信息和标签信息。Movielens 数据集有多个版本,本文采用的是 Movielens-100K 数据集,它包含 943 个用户对 1 652 部电影的 100 000 个评分,稀疏度为 93.7%。

FilmTrust 数据集[17]是在 2011 年 6 月从 FilmTrust 网站上抓取下来的一个关于电影评分和用户间信任度评级的数据集。该数据集由 ratings.txt 和 trust.txt 两部分组成,包含了 1 508 个用户对 2 071 部电影的35 497 个评分,以及关于 1 642 个用户间信任度评级的1 853条数据,稀疏度为 98.86%。

3.2 实验结果及分析

(1) 冷启动状态下的推荐

冷启动状态下的推荐结果如表 1 所示,其中的用户特征是指[年龄,性别,职业]。

表 1 Movielens 数据集上冷启动推荐结果

目标用户特征	推荐物品	喜欢推荐物品的老用户	老用户特征
[22,M,engineer]	100, 174, 172, 258,405,50,181, 12,56	216,844,487,561, 69, 435, 105, 627, 714,70	22-27, M,engineer
[43,F,admin]	286, 258, 269, 321,50,294,301, 257,121,237	89,238,79,529,72, 873, 34, 799, 720,716	38-49, F,admin
[26,M,executive]	258,121,7,100, 405, 597, 117, 151,546,288	181, 862, 54, 871, 653, 724, 84, 213, 546,374	25-33, M,executive
[16,F,student]	288, 294, 682, 258, 313, 815, 121,275,237,111	434,904,341,646, 849,281,618,482, 588,787	15-18, F,student
[30,M,educator]	50,181,127,475, 1,187,288,173, 318,286	791, 315, 686, 51, 64, 794, 903, 242, 243,569	28-34, M,educator

(2) 用户数和物品数较为接近时的推荐结果

本文将基于用户的协同过滤推荐算法和基于物品的协同过滤推荐算法所获得的推荐物品列表与用户真实评价物品列表进行比对,并以 F1 值的高低抉择当前用户所适用的推荐算法。

实验中首先选择了两个用户,对每个用户分别用 UserCF 算法和 ItemCF 算法实施推荐,对比结果如图 2 所示,其中 N 为推荐物品数量。

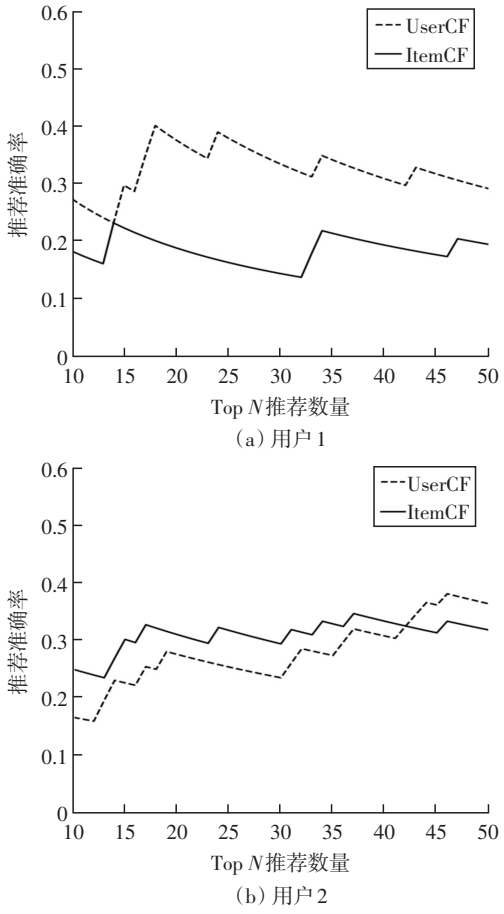
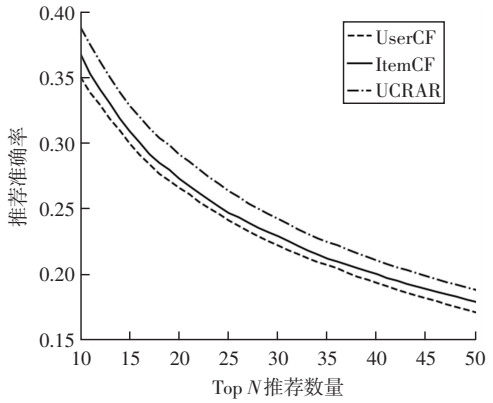


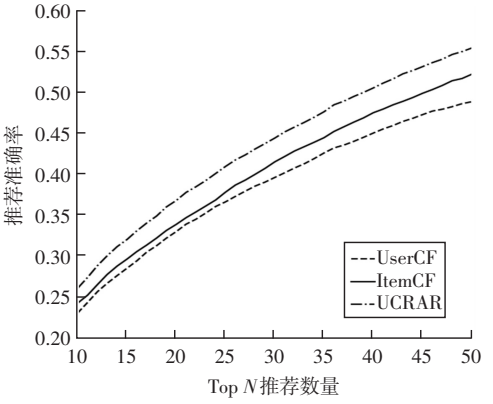
图 2 两个用户在两种推荐方法上的 F1 值对比

可以看出:用两种推荐算法对这两个用户推荐时所取得的 F1 值并不一致。对于用户 1,UserCF 算法的 F1 值总体上优于 ItemCF 算法;而对于用户 2,ItemCF 算法的 F1 值总体上优于 UserCF 算法。

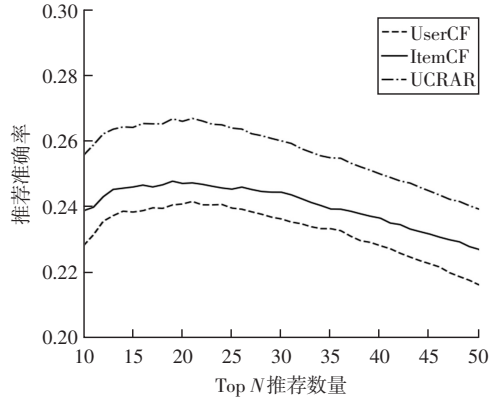
进一步从全体用户出发,对比全体用户分别使用 UserCF 算法、ItemCF 算法和本文的 UCRAR 策略的推荐效果,Movielens 数据集上的实验结果如图 3 所示。FilmTrust 数据集上的实验结果见表 2 至表 4。其中 N 为推荐物品数量。



(a) 准确率



(b) 召回率



(c) F1值

图 3　Movielens 数据集上的对比

表 2　FilmTrust 数据集上的准确率对比

N	10	20	30	40	50
UserCF	0.394 51	0.232 57	0.160 75	0.121 28	0.097 53
ItemCF	0.401 09	0.236 71	0.162 18	0.123 49	0.099 43
UCRAR	0.402 99	0.238 94	0.163 63	0.124 27	0.100 01

表 3　FilmTrust 数据集上的召回率对比

N	10	20	30	40	50
UserCF	0.720 48	0.832 45	0.864 64	0.868 15	0.870 59
ItemCF	0.726 88	0.852 60	0.896 04	0.910 33	0.914 98
UCRAR	0.746 44	0.870 45	0.908 20	0.916 54	0.919 57

表 4　FilmTrust 数据集上的 F1 值对比

N	10	20	30	40	50
UserCF	0.448 58	0.330 68	0.251 78	0.200 32	0.166 59
ItemCF	0.454 68	0.336 81	0.254 70	0.204 31	0.170 02
UCRAR	0.460 13	0.340 78	0.257 09	0.205 59	0.170 98

综合图 2 和图 3 以及表 2 至表 4 中的结果可以看出,对于不同特征的用户使用 UCRAR 策略所取得的准确率、召回率和 F1 值均好于只用 UserCF 算法或 ItemCF 算法对用户进行推荐的方式。

4　结束语

本文基于 K-means 聚类算法和协同过滤推荐算法设计了一种基于用户特征和评分的精准推荐策略。该策略通过引入用户特征,使用 K-means 聚类算法对全体用户特征进行聚类,将新用户所属类中其他用户喜好的物品中的 Top N 个推荐给新用户,解决了新用户冷启动问题;又根据物品数和用户数的大小关系,或者不同推荐算法所得 F1 值的大小关系,来决定选择将哪种推荐算法产生的结果推荐给老用户,弥补了单一协同过滤推荐算法在实际应用上的诸多不足。最后,将针对新老用户做出不同推荐的方法结合在一起,扩展了算法的使用范围,增强了算法的稳定性。在真实数据集上的实验结果表明,本文提出的基于用户特征和评分的精准推荐策略有着较高的准确度。

参考文献:

[1] 李俊.基于聚类的推荐算法研究与应用[D].南京:南京邮电大学,2018.
LI Jun. Research and application of recommendation algorithm based on clustering[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2018. (in Chinese)

[2] BOURHIM S, BENHIBA L, IDRISSI M A J. Investigating algorithmic variations of an RS Graph-based collaborative filtering approach[C]// Proceedings of the ArabWIC 6th Annual International Conference Research Track. 2019: 1-6.

[3] 何佳知.基于内容和协同过滤的混合算法在推荐系统中的应用研究[D].上海:东华大学,2016.
HE Jiazhi. The research of content-based and collaborative filtering algorithm in recommendation system [D]. Shanghai: Donghua University, 2016. (in Chinese)

[4] RAMEZANI M, MORADI P, AKHLAGHIAN F. A pattern mining approach to enhance the accuracy of collaborative filtering in sparse data domains[J]. Physica A: Statistical

Mechanics and its Applications,2014,408:72-84.

[5] KIRAN R U, KITSUREGAWA M. Towards addressing the coverage problem in association rule-based recommender systems[M]// Lecture Notes in Computer Science. Berlin: Springer, 2013:418-425.

[6] 乔雨.面向冷启动问题的推荐算法研究[D].南京:南京邮电大学,2018.
QIAO Yu. Research on recommendation algorithms against cold-start problem[D]. Nanjing: Nanjing University of Posts and Telecommunications,2018.(in Chinese)

[7] LI W J, XUE H. A clustering algorithm based on weighted distance and user preference of incorporating time factors [C]// Proceedings of the 6th International Conference on Information Technology.2018:1-6.

[8] LAVIE T, SELA M, OPPENHEIM I, et al. User attitudes towards news content personalization [J]. International Journal of Human-Computer Studies, 2010, 68 (8): 483-495.

[9] ZHANG F, ZHOU W T, SUN L L, et al. Improvement of pearson similarity coeffcient based on item frequency [C]// International Conference on Wavelet Analysis and Pattern Recognition.2017.

[10] KHOSHNESHIN M, STREET W N. Collaborative filtering via euclidean embedding [C]// Proceedings of the 4th ACM Conference on Recommender Systems.2010:87-94.

[11] MOHANA H, SURIAKALA M. Integrated cosine and tuned cosine similarity measure to alleviate data sparsity issues for personalized recommendation [C]// International Conference on Computational System & Information Technology for Sustainable Solutions.2018.

[12] MONATH N, ZAHEER M, SILVA D, et al. Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space [C]// Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.2019:714-722.

[13] KRISDHAMARA W P, PHARMASETIAWAN B, MUTIJARSA K. Improvement of collaborative filtering recommendation system to sparsity problem using combination of clustering and opinion mining methods[C]// International Seminar on Application for Technology of Information and Com-munication.2019.

[14] KAMALI H M, SASAN A. MUCH-SWIFT: a high-throughput multi-core HW/SW co-design K-means clustering architecture[C]// Proceedings of the 2018 on Great Lakes Symposium on VLSI.2018.

[15] 王振武,徐慧.数据挖掘算法原理与实现[M].北京:清华大学出版社,2015.

[16] HARPER F M, KONSTAN J A. The Movielens Datasets: History and Context[M]. New York: ACM Press,2015.

[17] GOLBECK J. Trust and nuanced profile similarity in online social networks[J]. ACM Transactions on the Web,2009, 3(4):1-33.