融合类别信息和用户兴趣度的协同过滤推荐算法

何 明 肖 润 刘伟世 孙 望
(北京工业大学计算机学院 北京 100124)

摘 要 协同过滤直接根据用户的行为记录去预测其可能感兴趣的项目,是现今最成功、应用最广泛的推荐技术。推荐的准确度受相似性度量方法效果的影响。传统的相似性度量方法主要关注用户共同评分项之间的相似度,忽视了评分项目中的类别信息,在面对数据稀疏性问题时存在一定的不足。针对上述问题,提出基于分类信息的评分矩阵填充方法,结合用户兴趣相似度计算方法并充分考虑到评分项目的类别信息,使得兴趣度的度量更加符合推荐系统应用的实际情况。实验结果表明,该算法可以弥补传统相似性度量方法的不足,缓解评分数据稀疏对协同过滤算法的影响,能够提高推荐的准确性、多样性和新颖性。

关键词 协同过滤,推荐系统,兴趣度,相似性计算
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.08.039

Collaborative Filtering Recommendation Algorithm Combing Category Information and User Interests

HE Ming XIAO Run LIU Wei-shi SUN Wang
(College of Computer Science, Beijing University of Technology, Beijing 100124, China)

Abstract Collaborative filtering is the most successful and widely used information technology to make personalized prediction by exploiting the historical behaviors of users. The accuracy of the recommendation depends on effectiveness of the similarity measure. The methods of traditional similarity measure, which mainly concern with the similarity of the common ratings but ignore the category information in the rated items, are suffering from data sparsity problem. To address this issue, we proposed a ratings matrix filling method which is based on classification information by combining with user interest similarity calculation method and consider the category information fully to make the measure of interest more realistic. The experimental results show that the proposed algorithm can relieve the influence of the sparsity of user-item ratings on collaborative filtering algorithm and improve recommendation accuracy, diversity, and novelty.

Keywords Collaborative filtering, Recommendation systems, Interest, Similarity computation

1 引言

随着互联网的不断发展,“信息过载”问题日益突出。搜索引擎虽然在很大程度上缓解了过载问题,但仍不能满足日益丰富的个性化需求。推荐系统的产生,旨在帮助人们找到其感兴趣的信息,为不同用户提供不同的服务,带来由“人找信息”到“信息找人”的转变。传统推荐系统挖掘用户与项目(user-item)之间的二元关系,为用户推荐可能满足需求的项目。由于协同过滤算法[1]不同于以往的推荐技术,即不需要依靠推荐对象的特征抽取来判断用户的兴趣,且能够很好地解决用户兴趣转移问题,推荐的个性化程度高,因此随后大量的推荐实例应用都使用了协同过滤思想。

协同过滤的基本思想是利用“人群的智慧”对信息进行过滤筛选,其基本假设是具有相同或者相似兴趣偏好的用户对项目的评分也是相似的。协同过滤思想使用统计技术寻找与目标用户有相似兴趣偏好的邻居用户,并根据邻居用户的评分信息来预测目标用户对物品项的评分值,选择预测评分值最高的前 N 项物品推荐给目标用户。对于何为“相似”的用户的问题,文献[2]认为其是指“兴趣和口味相似”的用户,相似的用户在对项目进行评分时往往会给出相近的评分,这也是利用协同过滤思想进行评分预测的主要依据。

与其他推荐算法相比,协同过滤算法具有以下两个方面的优势:

(1)对推荐对象无特殊要求,对于复杂、抽象的资源也能实现推荐。

(2)只需要显式或者隐式的用户历史评价数据,不需要有关用户本身的属性知识,且不会给用户的推荐体验带来任何负面影响。

到稿日期:2016-07-05 返修日期:2016-10-31 本文受国家自然科学基金项目(91646201,91546111),北京市教委科研计划一般项目(KM201710005023)资助。

何 明(1975-),男,博士,副教授,主要研究方向为推荐系统、数据挖掘、机器学习,E-mail:heming@bjut.edu.cn;肖 润(1990-),男,硕士生,主要研究方向为推荐系统、数据挖掘;刘伟世(1989-),男,硕士生,主要研究方向为机器学习;孙 望(1990-),男,硕士生,主要研究方向为信息检索。

正是由于这些显著的优势，协同过滤推荐算法在实际应用中使用得非常广泛。在协同过滤算法中，确定目标用户的最近邻是整个算法的关键，因此如何通过评分矩阵来计算项目之间的相似度，是协同过滤算法的核心问题。传统的计算相似度的方法有余弦相似度、相关相似性以及修正的余弦相似度算法[3]。由协同过滤思想可以看出，用户评分数据越多，协同过滤推荐算法的推荐质量越高。但是在实际的推荐系统中，用户的评分数据往往非常稀疏，传统的相似度方法难以准确地度量用户间的相似性，从而导致推荐准确率较低[4]。尤其是近几年来随着电子商务系统规模的不断扩大，用户和商品的数目都急剧增加，用户评分的项目往往只占项目总数的 1%，导致数据极度稀疏，传统的相似度度量方法的准确度不断下降。为此，文献[5]提出了基于项目评分预测的协同过滤算法，一定程度上弥补了数据稀疏的不足；文献[6]融合协同过滤推荐算法和基于内容的推荐算法，一定程度上缓解了由新用户和新项目带来的影响；文献[7]和文献[11]将用户兴趣相似性与传统相似性方法相结合来计算用户的相似度，一定程度上提高了推荐的准确性。

为了缓解数据稀疏性导致的推荐准确性下降的问题，本文充分利用了各个项目的分类信息：首先，使用基于分类信息的领域最近邻方式对评分项并集中的未评分项目进行估值填充；同时，本文不再使用单一的相似度度量方法，而是将用户兴趣相似性与传统相似性方法相结合来计算用户的相似度，并且对传统的用户兴趣相似性度量方法进行了改进，提出一种基于用户兴趣度的协同过滤推荐算法（UICF）。该方法考虑到了各个类别的物品数量的差异，能更加准确地刻画用户对不同项目类别的真实偏好程度，使得用户相似性度量更加准确，推荐准确率更高。

本文第 2 节具体介绍了传统的协同过滤算法以及几种相似性度量方法；第 3 节提出一种基于用户兴趣的协同过滤推荐算法和基于类别信息对未评分项进行评分预测的方法；第 4 节为实验及其结果分析，比较了几种协同过滤推荐算法推荐的准确性；最后总结全文。

2　传统的协同过滤推荐算法

2.1　传统的协同过滤算法的推荐步骤

传统的基于用户的协同过滤算法通常经过 3 个步骤[8]对目标用户产生推荐。

（1）收集可以代表用户兴趣的信息，如用户评分等，并建立相应的模型，如建立一个基于用户-项目评分的矩阵 $R(m,n)$，如表 1 所列，其中 m 表示用户数，n 表示项目数，$R_{i,j}$ 表示用户 i 对项目 j 的评分值。

表 1　用户-项目评分矩阵 $R(m,n)$

	$item_1$	…	$item_j$	…	$item_n$
$User_1$	$R_{1,1}$	…	$R_{i,j}$	…	$R_{1,n}$
⋮	⋮	⋮	⋮	⋮	⋮
$User_i$	$R_{i,1}$	…	$R_{i,j}$	…	$R_{i,n}$
⋮	⋮	⋮	⋮	⋮	⋮
$User_m$	$R_{m,1}$	…	$R_{m,j}$	…	$R_{m,n}$

（2）计算目标用户与其他用户之间的相似度，找出与目标用户最相似的“最近邻”集合作为目标用户的最近邻。

（3）根据最近邻集合中的实际评分矩阵预测目标用户对未评分项目的评分。选取预测评分最高的 N 个项目作为推荐结果反馈给目标用户。

上述就是经典的 top-N 推荐算法[15]。其中，准确计算用户间的相似度是整个推荐过程的核心。

2.2　传统相似度计算方法

对于目标用户 u，计算其与其他用户的相似度 $sim(u,v)$，得到 K 个递减排列的邻居集合 $N=\{N_1,N_2,\cdots,N_K\}$，$u\notin N$。通常有 3 种计算相似度的方法：余弦相似性、相关相似性和修正的余弦相似性。

（1）余弦相似性：每个用户的所有评分数据被看作是 n 维项目空间中的一个向量，n 为总的项目数量，如果用户没有对某个项目进行评分，则默认设置为零。两个用户间的相似度可以看作是两个用户评分向量之间夹角的余弦值，余弦值越大，说明用户间的相似度越高。假设两个用户 x 和 y 的评分向量分别为 u 和 v，则 x 和 y 之间的余弦相似度为：

$$sim(x,y)=\cos(u,v)=\frac{\vec{u}\times\vec{v}}{\|\vec{u}\|\ \|\vec{v}\|} \tag{1}$$

（2）相关相似度：相关相似度又称为 Person 相关性。假设用户 u 和用户 v 的共同评分项集合为 R_{uv}，R_{ui} 和 R_{vi} 分别表示用户 u 和用户 v 对项目 i 的评分。则用户 u 和用户 v 的相关相似性为：

$$sim(u,v)=\frac{\sum\limits_{i\in R_{uv}}(R_{u,i}-\overline{R_u})\times(R_{v,i}-\overline{R_v})}{\sqrt{\sum\limits_{i\in R_{uv}}(R_{u,i}-\overline{R_u})^2}\times\sqrt{\sum\limits_{i\in R_{uv}}(R_{v,i}-\overline{R_v})^2}} \tag{2}$$

（3）修正的余弦相似度：余弦相似性度量方法忽略了不同用户的评分尺度问题。修正的余弦相似性度量方法减去了用户对项目的平均评分。设 R_w 表示用户 u 与用户 v 共同评分过的集合，R_u 和 R_v 分别表示用户 u 和用户 v 的评分集合。则用户 u 和用户 v 的修正余弦相似度为：

$$sim(u,v)=\frac{\sum\limits_{i\in R_{uv}}(R_{u,i}-\overline{R_u})\times(R_{v,i}-\overline{R_v})}{\sqrt{\sum\limits_{i\in R_u}(R_{u,i}-\overline{R_u})^2}\times\sqrt{\sum\limits_{i\in R_v}(R_{v,i}-\overline{R_v})^2}} \tag{3}$$

2.3　评分预测

通常采用 K 近邻[13]方法进行评分预测，即选择与目标用户最相似的 K 个用户作为最近邻集合来进行计算。假设集合 U 表示目标用户 u 的最近邻集合，则用户 u 对未评分项 i 的预测评分值可用下式表示：

$$P(u,i)=\overline{R_u}+\frac{\sum\limits_{u_k\in U}sim(u,u_k)\times(R_{u_k,i}-\overline{R_{u_k}})}{\sum\limits_{u_k\in U}sim(u,u_k)} \tag{4}$$

其中，$R_{u_k,i}$ 表示 u_k 对项目 i 的非空评分，$\overline{R_{u_k}}$ 表示 u_k 在与 u 的共同评分项集合中的平均评分，$\overline{R_u}$ 则表示 u 在所有的项目上的平均评分。

3　基于用户兴趣度的协同过滤算法

3.1　基于类别的评分预测

在 2.2 节所提到的相似性计算方法中，余弦相似性计算方法是一种通过距离测算的方式来计算用户间相似性的方法；相关相似性方法是通过统计学的方式来计算用户相似性的一种方法。传统的相似性计算方法都存在一定的弊端[7]，尤其在数据稀疏的情况下，用户评分数据极端稀疏，传统的相

似度度量方法难以发挥作用。在余弦相似性度量方法中，将用户没有评分的项目的评分值均设置为 0，设用户 i 对项目 j 的评分为 $r_{i,j}$，则在构造用户评分数据矩阵 $R(m,n)$ 时，任意用户 i 对任意项目 j 的评分 $R_{i,j}$ 为：

$$R_{i,j}=\begin{cases} r_{i,j}, & \text{if user } i \text{ rated item } j \\ 0, & \text{if user } i \text{ not rated item } j \end{cases}$$

这种做法虽然可以提高计算效率，但在数据稀疏的情况下，上述方法所得评分矩阵的可信度并不高。事实上，用户对未评分项目的评分并不完全相同。

另外，修正的余弦相似度和相关相似度都考虑到了用户间的共同评分项，如果用户间的共同评分项很少，即使用户在小集合上的评分非常相似，也不能保证他们之间的相似性很高。

综上所述，传统的 3 种相似性度量方法在数据极端稀疏的情况下并不能有效地度量用户间的相似性，进而会导致推荐质量下降。

为了解决用户评分数据的稀疏性问题，传统的方法是将用户未评分项设置成一个固定的缺省值（一般设置为评分域的中间值，如 5 分制评分系统中设置为 3），或者设置为用户的评价平均值[9-10]。然而，用户对未评分项目的评分不可能完全相同，因此设置固定缺省值的方法并不能从根本上解决用户评分数据极端稀疏的问题[14]。

为了有效解决用户评分数据极度稀疏的问题，本文采用评分项并集来计算用户间的相似性。同时，采用基于类别信息的“领域最近邻”方法对评分项并集进行估值填充，使得最近邻的选取更加准确。

本文基于项目所属类别来计算用户的最近邻集合，即领域最近邻。设用户 u 和用户 v 的评分项并集为 U_{uv}，为了预测用户 u 对 U_{uv} 中未评分项 i 的评分值，需选取用户-项目评分矩阵 $R(m,n)$ 中属于项目 i 所在类别 C_a 的所有项目评分数据组成领域评分矩阵 R_i，然后基于 R_i 计算用户 u 的领域最近邻集合，进而基于得到的领域最近邻集合对未评分项 i 进行评分值预测。

项目类别对应着用户的兴趣领域，用户对不同类别的项目所表现出来的兴趣度有一定差异，在某些情况下差异可能非常大，因此基于全部用户评分数据寻找最近邻并不是很合适，而根据未评分项所在类别数据的矩阵寻找最近邻将更加准确。基于上述原因，本文通过领域最近邻对缺省值进行预测。

在实际的环境中，所有的项目都被划分到若干个类别中。以 MovieLens 数据集[12]为例，其中包含 943 位用户对 1682 部电影的 100000 条评分数据，将电影分为 19 个类别。设目标用户 u 和用户 v 的评分项集合为 I_u 和 I_v，I_u 和 I_v 中各个项目所属类别的集合分别为 C_u 和 C_v，设 $C_t=C_u\cap C_v$，则存在两种情况，如表 2 所列。

表 2 用户评分项类别分布

	Action		Comedy		Horror		
	I_1	I_2	I_3	I_4	I_5	I_6	I_7
u	2	4			1	5	3
v		3	4	5		4	3

(1)如果存在 $C_i\in(C_u\cup C_v)$，且 $C_i\notin C_t$，即存在用户在评分项并集中对某个类别中的所有项目都没有做出评分，如在表 2 中，u 在 comedy 分类中的项目 I_2 和 I_3 没有评分，而 v 在此分类中的项目有评分，则本文将用户在 comedy 分类中的平均评分作为 u 对 I_2 和 I_3 项目的填充评分，即对 C_i 中的未评分项目的评分值设置为该分类的平均评分值。

(2)对于分类 C_t 中的未评分项，如在表 2 中，用户 u 和用户 v 在 horror 分类及 action 分类中的项目都有评分，则针对用户 v 对于项目 I_1 和项目 I_5 的预测评分值，本文使用领域最近邻方法进行计算，公式如下：

$$P_i=\overline{R_u}+\frac{\sum\limits_{u_k\in U_c} sim(u,u_k)\times(R_{u_k,i}-\overline{R_{u_k}})}{\sum\limits_{u_k\in U_c} sim(u,u_k)} \tag{5}$$

其中，U_c 为用户 u 在类别 c 中的最近邻集合，$sim(u,u_k)$ 表示用户 u 和用户 u_k 的相似性，$R_{u_k,i}$ 表示用户 u_k 对项目 i 的评分值，$\overline{R_u}$ 和 R_{u_k} 分别表示用户 u 和用户 u_k 在分类 c 中的平均评分。

使用上述方法即可将两个用户评分项并集中的未评分项进行估值填充，之后就可使用 2.2 节中的相似性度量方法计算两个用户间的相似度。

3.2 基于兴趣度的相似性度量方法

传统的协同过滤算法仅考虑了用户对项目单一的评分相似度，基于用户兴趣度的协同过滤算法考虑了项目所属的类型信息，根据用户评分所反映出的用户对项目类别的喜好相似度来改进用户间的相似度度量，在一定程度上提高了算法的推荐质量。这种基于用户间多相似度的协同过滤推荐算法在实际应用中越来越普遍[11]。

在实际环境中，用户对项目类别的喜好程度为推荐提供了重要的依据，一个类别中的项目被用户评价的次数越多，表明用户对这个类别的项目越感兴趣。当两个用户具有相同的兴趣爱好时，也可以认为他们之间具有较高的相似度。由用户-项目评分矩阵可计算出用户-项目类别评分数目矩阵 N：

$$N=\begin{bmatrix} N_{11} & \cdots & N_{1k} \\ \vdots & \ddots & \vdots \\ N_{s1} & \cdots & N_{sk} \end{bmatrix}$$

其中，s 行代表用户数目，k 列代表项目类别数目，N_{sk} 表示用户 s 对 k 类项目的评价数目。

用户 x 对 a 类项目的兴趣度可表示为：

$$I_{xa}=\frac{N_{xa}}{N_x} \tag{6}$$

其中，N_{xa} 表示用户 x 对 a 类项目的评价总数；N_x 表示用户 x 的评价总数。用户对某一类别的评价数越高，表明该用户对这一类别的兴趣度越高。

从而两个用户的兴趣相似度可以用余弦相似度来计算，即：

$$sim(x,y)=\frac{\sum\limits_{a=1}^{k} I_{xa}I_{ya}}{\sqrt{\sum\limits_{a=1}^{k} I_{xa}^2}\sqrt{\sum\limits_{a=1}^{k} I_{ya}^2}} \tag{7}$$

其中，k 为项目的类别数，I_{xa} 表示用户 x 对 a 类项目的兴趣度。

因此，结合用户评分相似度和用户兴趣相似度便得到用

户间的整体相似度,即:

$$sim(x,y)=(1-w)\times sim_r(u,v)+w\times sim_I(u,v) \quad (8)$$

其中,$sim_r(u,v)$表示用户 u 和用户 v 的项目评分相似度,$sim_I(u,v)$表示用户 u 和用户 v 的兴趣相似度,$0<w<1$。

但是,式(6)中的类别兴趣度的度量方法中并没有考虑到各个类别中的项目数量的因素,各个类别中的项目数量的差异可能非常大。以第 4 节表 4 中数据集 M1 为例,表 3 列出 M1 中各个类型的电影数目统计数据。

表 3　MovieLens 数据集各个类别中的电影数目

unknown	Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	Mystery	…
2	251	135	42	122	505	109	50	725	22	24	92	56	61	…

表 3 中,类别中的电影数目存在较大的差异,19 个类别中,有 14 个类别的电影数目多于 50,数量小于 100 的类别有 10 类,最高的数值为 725,最低的数值为 2。各个类别中的电影数目差异很大。

此外,在实际的应用中,类别中项目数量的差异对类别兴趣度的度量也有一定影响。以 MovieLens 数据集为例,假设用户 x 对动作片和恐怖片都进行了 20 次评分,采用式(6)得出用户 x 对动作片和恐怖片的喜好程度相同。但实际情况是,动作片的影片数量远远大于恐怖片,恐怖片相对于动作片来说是相对小众的电影类别。事实上,用户 x 对动作片和恐怕片的评分数量相同隐含着一个重要信息,即用户 x 是一个恐怖片爱好者。

因此,本文对式(7)中的兴趣的度量方法进行了改进,使其在实际应用中能更加准确地计算用户的兴趣度,计算公式如下:

$$sim(x,y)=\frac{\sum_{a=1}^{k}T_{xa}T_{ya}}{\sqrt{\sum_{a=1}^{k}T_{xa}^2}\sqrt{\sum_{a=1}^{k}T_{ya}^2}} \quad (9)$$

其中,T_{xa} 为改进后的用户 x 对 a 类项目兴趣度的度量:

$$T_{xa}=\frac{N_{xa}}{G_a} \quad (10)$$

其中,G_a 表示属于 a 类别的项目数量,N_{xa} 表示用户 x 对 a 类项目的评价总数。

式(9)充分考虑了各个类别中项目数量的差异以及各个类别受欢迎的程度。式(10)在对项目兴趣度的度量方法中加入了对项目数量的考量,使其更加贴近实际的推荐情况。

同时,如果 G_a 的数值过低,即类别中项目数过少,将导致 T_{xa} 数值过高,进而导致用户兴趣度计算不合实际。以 MovieLens数据集为例,类别为 Fantasy 的电影数目为 22,如果使用式(10)计算用户兴趣度,会得到一个很高的兴趣度数值,这显然是不切实际的。针对这种情况,本文设置一个阈值 M,如果 G_a 大于 M,则采用式(10)计算用户对某项目的兴趣度;如果 G_a 小于 M,则采用式(6)计算用户对项目的兴趣度。用户兴趣相似度的度量方法为:

$$sim_I(x,y)=\begin{cases}\dfrac{\sum_{a=1}^{k}I_{xa}I_{ya}}{\sqrt{\sum_{a=1}^{k}I_{xa}^2}\sqrt{\sum_{a=1}^{k}I_{ya}^2}}, & G_a\leqslant M\\ \dfrac{\sum_{a=1}^{k}T_{xa}T_{ya}}{\sqrt{\sum_{a=1}^{k}T_{xa}^2}\sqrt{\sum_{a=1}^{k}T_{ya}^2}}, & G_a>M\end{cases} \quad (11)$$

推荐系统往往倾向于向用户推荐流行的、热门的产品,这一现象被称为推荐系统中的流行偏置现象。流行偏置现象导致马太效应的产生,即推荐系统倾向于推荐热门的产品,导致热门商品的热门程度越来越高,被推荐的概率也越来越大,而冷门产品则越来越不被用户关注。

图 1 将 Movielens 数据集中的产品按照流行度降序排列。从图 1 中可以发现,不同产品的流行度差别很大,最热门的产品被评分近 600 次,而大量产品被评分的次数不足 10 次。

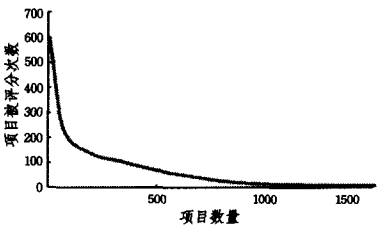

图 1　MovieLens 数据集的产品流行度分布

由图 1 可知,MovieLens 中的电影流行度存在很大差异,导致推荐算法面临流行偏置问题的挑战。本文对兴趣度度量方法进行了改进,通过计算用户兴趣相似度,同时结合评分相似度,最终求得用户间的整体相似度。该方法充分考虑了各个项目所在分类的受欢迎程度,使得对兴趣度的度量更加合理,一定程度上缓解了流行偏置问题,使得推荐结果更具多样性和新颖性。

3.3　推荐过程

结合用户兴趣度,由式(8)和式(11)可以计算用户间的相似度,进而可以发现目标用户的最近邻集合,再根据式(4)可以得出目标用户对未评分项的预测评分,预测评分最高的 N 项作为 top-N 推荐结果集合。对于目标用户 u,要为其推荐合适的项目集合 P,即选取用户 u 对未评分项目预测评分最大的 N 个项目,推荐流程如图 2 所示。

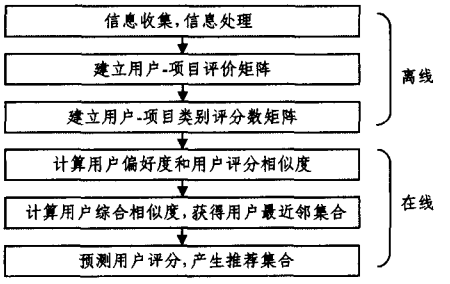

图 2　基于用户兴趣度的协同过滤算法的流程

算法 1　基于用户兴趣度的协同过滤算法(UICF)

输入:用户-项目评分矩阵,项目属性矩阵,最近邻用户数 K,推荐集合数目 N

输出:目标用户 u 的 top-N 推荐集合

Step 1　建立修正的用户-项目评分矩阵

对于目标用户 u 与其他用户的评分项并集 U，使用本文 3.1 节中的领域最近邻方法填充并集 U 中的未评分项，得到修正的用户-项目评分矩阵。

Step 2 建立用户-项目类别评分数目矩阵

根据用户-评分矩阵和项目属性矩阵，计算出目标用户 u 在各个类别中的项目评分数目。

Step 3 计算目标用户与其他非目标用户的整体相似性

基于修正的用户-项目评分矩阵 R′，利用式(3)即修正的余弦相似度计算出基于用户评分的相似度，利用式(10)计算出基于用户兴趣度的相似度，选取一个权重因子 w，根据式(8)计算出用户 u 和其他用户的最终相似度矩阵 sim。

Step 4 进行评分预测，生成推荐结果。

根据相似度矩阵，获取目标用户的邻居集合 S，并根据邻居集合 S 由式(4)计算出目标用户 u 对项目 i 的预测评分。将预测评分值由高到低排列，取出最大的 N 个项目作为 top-N 推荐结果集。

假设有 m 个用户和 n 个物品，需要查找 k 个近邻用户。Step 1 只在项目 i 所属类别 C 中寻找最近邻集合，因此时间复杂度为 $O(m\times n_{C_i})\approx O(m)<O(m\times n)$，其中 n_{C_i} 为对项目 i 做出评分的所有用户在类别 C 中的评分项总数；Step 2 为离线计算，时间复杂度为 $O(l)<O(m\times n)$；Step 3 在修正的用户-项目评分矩阵中计算目标用户与 m 个基本用户的相似度，时间复杂度为 $O(m)$；Step 4 对 m 个相似度进行排序，查找出最近邻居，时间复杂度为 $O(m\log m)$，通过 k 个最近邻居生成推荐，时间复杂度为 $O(k\times m)$，由于 k 为远小于 m 的常数，因此时间复杂度为 $O(m)$。综合分析算法 1 的整个过程，为用户 u 产生推荐结果集的时间复杂度为 $O(m)$，总的时间复杂度为 $O(m^2)$。

4 实验比较和分析

本实验的目的主要包括以下几个方面：

(1)基于本文提出的结合用户兴趣度的多维度相似性计算方法，发现在 MovieLens 数据集上使 MAE 最小的阈值 w。

(2)比较本文提出的结合用户兴趣度的多维度相似性计算方法与传统方法的推荐精度，验证结合用户兴趣度的多维度相似性计算方法有效提高了推荐精度。

(3)观察本文提出的兴趣度度量方法对推荐多样性和新颖性的影响，验证该方法可以提高推荐的多样性和新颖性。

4.1 实验设置

本文选择 Apache Mahout 作为基础推荐引擎，该开源项目已经集成了协同过滤、聚类和分类等多种推荐算法。本文在 Mahout[16] 的基础上开发部署，并选择 MovieLens[17] 作为实验的测试数据集。MovieLens 是明尼苏达大学 GroupLens 小组搜集的电影评价数据，通过用户对电影的评分(1～5)进行电影推荐。MovieLens 提供了几种不同数量级的数据集，实验选取了其中的两种，并分别将这两种不同规模的数据集随机地划分成训练集和测试集两部分，其中训练集占整个数据集的 80%，测试集占 20%。数据集的具体描述细节如表 4 所列。

表 4 MovieLens 数据集信息描述

数据集	用户数	项目数	评分数	稀疏性/%
M1	943	1682	100000	93.70
M2	6040	3706	1000209	95.53

4.2 评估标准

由于推荐系统的应用目标不同，有多种评价方式对推荐算法在当前目标下的表现进行评估，以帮助系统决策者选择适合当前环境的任务和推荐算法。对推荐效果的评价主要分为 3 种类型：准确性评价、多样性评价和新颖性评价。

(1)准确性评价

对推荐结果准确性的评价标准主要有两类：统计精度度量方法和决策支持精度度量方法。本文实验采用统计精度度量方法中广泛应用的平均绝对误差 MAE(Mean Absolute Error)作为评价标准，它通过预测出的用户评分和实际用户的评分之间的偏差来度量预测的准确性。MAE 越小，表示推荐精度越高。

设预测的用户评分集合表示为 $\{p_1,p_2,p_3,\cdots,p_n\}$，与预测评分对应的实际用户评分集合为 $\{q_1,q_2,q_3,\cdots,q_n\}$，则 MAE 可通过式(12)求得：

$$MAE=\frac{\sum_{i=1}^{N}|p_i-q_i|}{N} \tag{12}$$

(2)多样性评价

准确率是评价推荐算法优劣的重要指标。但若推荐算法只能为用户推荐同一种类型的产品，即使其准确率很高，也会影响用户的体验。因此，推荐丰富多彩的产品是推荐算法关注的另一个重要方面，推荐结果的多样性和新颖性也是评价算法推荐效果的重要指标。

推荐结果对系统产品的覆盖程度在一定程度上也反映了推荐结果的多样性，它度量了一个推荐算法可以把多大比例的产品推荐给至少一个用户。针对 Top-N 推荐问题，算法的覆盖度(COV)可以使用式(13)进行计算：

$$COV(N)=\frac{N_d}{N} \tag{13}$$

其中，N_d 是在 Top-N 推荐中推荐算法推荐给至少一个用户的产品总数。COV 越大，说明推荐算法的多样性越好。

(3)新颖性评价

此外，推荐产品的热门程度可在一定程度上反映推荐结果的新颖性，越不热门的产品越能让用户觉得新颖。因此，可以使用推荐结果对长尾产品的覆盖量(Coverage in Long-tail，CIL)评价推荐算法的新颖性，计算方法如式(14)所示：

$$CIL(N)=\frac{NL_c}{N} \tag{14}$$

其中，NL_c 是长尾产品集合和 N_d 的交集。CIL 越大，则说明推荐结果中的长尾产品越多，推荐结果的新颖性越好。

4.3 实验结果及分析

为了验证和比较本文所提方法的有效性，分别在数据集 M1 和 M2 下进行了以下几组实验。

(1)实验 1：该组实验主要用于阈值 w 的选取。针对 3.2 节中提出的基于兴趣度的相似度度量方法，并考虑到式(8)中

阈值 w 的取值对推荐精度的影响，本实验中 w 在0.1～0.9之间取值，每次间隔 0.1，依次观察 w 的变化对推荐精度的影响，最后选出最佳的 w 取值。本实验选取表 4 中的 M1 和 M2 作为实验数据，实验结果如图 3 所示。可以看出，当 w 取值为 0.8 时，推荐系统的 MAE 最小，推荐精度最高。因此，本文取 w 值为 0.8。

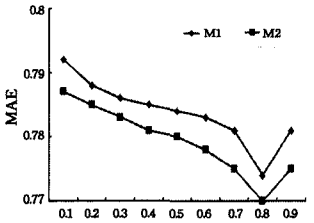

图 3　阈值 w 对 MAE 的影响

(2)实验 2：该组实验主要比较余弦 Cosine，Pearson 以及矩阵分解方法(Matrix Factorization，MF)[18]与 UICF 方法在协同过滤推荐中的推荐准确度。近邻数量由 5 个递增到 30 个，间隔为 5，MF 方法的因子维度为 10。实验结果如图 4 和图 5 所示，分别表示在数据集 M1 和 M2 上的实验结果。注意到，当邻居数量为 5 时，文献[11]方法的 MAE 值略小于 UICF 方法，这主要是由于文献[11]基于用户间的多相似度进行评分预测，使得在邻居数目少的情况下 UICF 方法的 MAE 略高(0.018 和 0.002)；当邻居数量为 5 时，MF 的 MAE 值也略小于 UICF 方法；而在其他情况下，UICF 方法的 MAE 均小于其他 3 种方法，说明本文提出的 UICF 方法能提高推荐准确性。

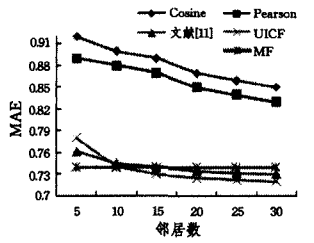

图 4　推荐精度比较(M1)

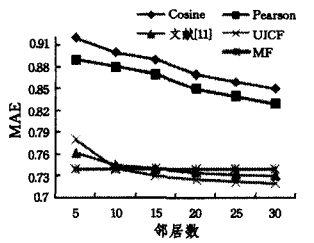

图 5　推荐精度比较(M2)

(3)实验 3：该组实验主要验证 UICF 方法的推荐多样性。实验设最近邻数为 5，通过计算 top-N 结果集，根据式(13)对比 UICF 方法、余弦方法、Pearson 方法、文献[11]方法以及 MF 的覆盖度差异，N 在 5～30 之间取值，每次间隔为 5。实验结果如图 6 和图 7 所示，分别表示在数据集 M1 和 M2 上的实验结果。由图 6 和图 7 可知，本文提出的 UICF 方法在取不同近邻数目时的 COV 值均高于传统方法和文献[11]推荐算法的 COV，提高了推荐多样性。

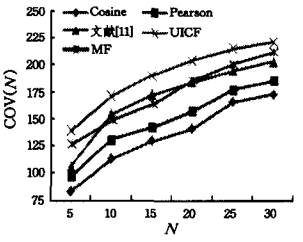

图 6　覆盖率 COV 比较(M1)

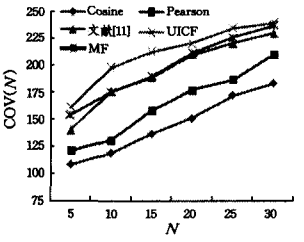

图 7　覆盖率 COV 比较(M2)

(4)实验 4：该实验用于验证 UICF 方法的推荐新颖性。设最近邻数为 5，通过计算 top-N 结果集，根据式(14)对比 UICF 方法、余弦方法、Pearson 方法、文献[11]的方法以及 MF 的覆盖度差异，N 在 5～30 之间取值，每次间隔为 5。实验结果如图 8 和图 9 所示，分别表示在数据集 M1 和 M2 上的实验结果。由图 8 和图 9 可知，本文提出的 UICF 方法在取不同近邻数目时的 CIL 值均高于传统方法和文献[11]推荐算法的 CIL，提升了推荐新颖性。

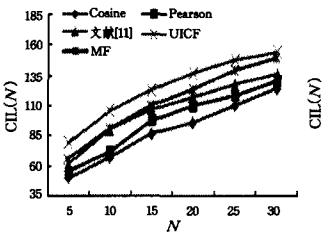

图 8　覆盖量 CIL 比较(M1)

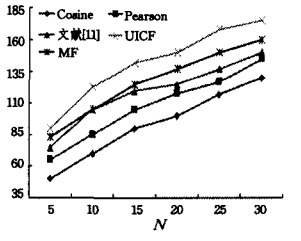

图 9　覆盖量 CIL 比较(M2)

结束语　协同过滤推荐算法的核心是计算用户之间的相似度。本文充分考虑了用户评分项的类别信息，使用类别信息填充用户评分项并集中的未评分项，并且在计算用户间的兴趣相似度时融合了对项目分类信息的考虑，使得相似度计算更加符合实际情况，一定程度上缓解了数据稀疏问题。通过实验可知，该算法有效地解决了传统相似性度量算法由于数据稀疏而引起的推荐结果不准确的问题，同时在一定程度上提升了推荐准确性、多样性和新颖性。下一步的工作是对算法进行优化和改进，并考虑利用用户画像等数据计算用户相似性，进一步提高推荐系统的推荐质量。

参 考 文 献

[1] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems: A survey of the state-of-art and possible extensions[J]. Transactions on Knowledge and Data Engineering, 2005, 17(6): 734-749.

[2] XU H L, WU X, LI X D, et al. Comparison study of Internet recommendation system [J]. Journal of Software, 2009, 20(2): 350-362. (in Chinese)
许海玲，吴潇，李晓东，等. 互联网推荐系统比较研究[J]. 软件学报，2009，20(2)：350-362.

[3] AHN H J. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem[J]. Information Sciences, 2008, 178(1): 37-51.

(下转第 269 页)

[5] ETTINGER A, RESNIK P, CARPUAT M. Retrofitting sense-specific word vectors using parallel text[C]// Proceedings of NAACL-HLT. 2016: 1378-1383.

[6] AKKAYA C, WIEBE J, MIHALCEA R. Iterative Constrained Clustering for Subjectivity Word Sense Disambiguation[C]// Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. 2014: 269-278.

[7] KLAPAFTIS I P, MANANDHAR S. Word sense induction & disambiguation using hierarchical random graphs[C]// Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. 2010: 745-755.

[8] TANG G B, YU D, XUN E D. An Unsupervised Word Sense Disambiguation Method Based on Sememe Vector in HowNet [J]. Journal of Chinese Information Processing, 2015, 29(6): 23-29. (in Chinese)
唐共波，于东，荀恩东. 基于知网义原词向量表示的无监督词义消歧方法[J]. 中文信息学报，2015，29(6)：23-29.

[9] QIAN T, JI D H, DAI W H. A Hypergraph Model for Word Sense Induction [J]. Journal of Sichuan University (Engineering Science Edition), 2016, 48(1): 152-157. (in Chinese)
钱涛，姬东鸿，戴文华. 一个基于超图的词义归纳模型[J]. 四川大学学报(工程科学版)，2016，48(1)：152-157.

[10] VAN DE CRUYS T, POIBEAU T, KORHONEN A. Latent vector weighting for word meaning in context[C]// Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011: 1012-1022.

[11] LAU J H, COOK P, MCCARTHY D, et al. Word sense induction for novel sense detection[C]// Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. 2012: 591-601.

[12] HUANG Y, SHI X, SU J, et al. Unsupervised word sense induction using rival penalized competitive learning [J]. Engineering Applications of Artificial Intelligence, 2015, 41: 166-174.

(上接第 235 页)

[4] DAI J L. Study on the sparsity problem of collaborative filtering algorithm [D]. Chongqing: Chongqing University, 2013. (in Chinese)
代金龙. 协同过滤算法中数据稀疏性问题研究[D]. 重庆：重庆大学，2013.

[5] DENG A L, ZHU Y Y, SHI B L. A collaborative filtering recommendation algorithm based on item ratingprediction [J]. Journal of Software, 2003, 14(9): 1621-1628. (in Chinese)
邓爱林，朱扬勇，施伯乐. 基于项目评分预测的协同过滤推荐算法[J]. 软件学报，2003，14(9)：1621-1628.

[6] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions [J]. IEEE transactions on knowledge and data engineering, 2005, 17(6): 734-749.

[7] JI X S, LIU Y B, LUO L M. Similarity measurement based on interest in collaborative filtering [J]. Journal of Computer Applications, 2010, 30(10): 2618-2620. (in Chinese)
嵇晓声，刘宴兵，罗来明. 协同过滤中基于用户兴趣度的相似性度量方法[J]. 计算机应用，2010，30(10)：2618-2620.

[8] CLEGER-TAMAYO S, FERNáNDEZ-LUNA J M, HUETE J F. Top-N news recommendations in digital newspapers [J]. Knowledge-Based Systems, 2012, 27(6): 180-189.

[9] ZHANG X S. Research on collaborative filtering recommendation algorithms for data sparsity [D]. Hefei: University of Science & Technology China, 2011. (in Chinese)
张学胜. 面向数据稀疏的协同过滤推荐算法研究[D]. 合肥：中国科学技术大学，2011.

[10] YU X. Research on recommendation methodsbased on collaborative filtering techniques[D]. Tianjin: Tianjin University, 2009. (in Chinese)
郁雪. 基于协同过滤技术的推荐方法研究[D]. 天津：天津大学，2009.

[11] FAN B, CHENG J J. Collaborative filtering recommendation algorithm based on user's multi-similarity[J]. Computer Science, 2012, 39(1): 23-26. (in Chinese)
范波，程久军. 用户间多相似度协同过滤推荐算法[J]. 计算机科学，2012，39(1)：23-26.

[12] MILLER B N, ALBERT I, LAM S K, et al. MovieLens unplugged: experiences with an occasionally connected recommender system[C]// Proceedings of the 8th International Conference on Intelligent User Interfaces. ACM, 2003: 263-266.

[13] LUO X, OUYANG Y X, XIONG Z, et al. The effect of similarity support in K-Nearest-Neighborhood based collaborative filtering [J]. Chinese Journal of Computers, 2010, 33(8): 1437-1445. (in Chinese)
罗辛，欧阳元新，熊璋，等. 通过相似度支持度优化基于 K 近邻的协同过滤算法[J]. 计算机学报，2010，33(8)：1437-1445.

[14] STECK H. Evaluation of recommendations: rating-prediction and ranking[C]// Proceedings of the 7th ACM Conference on Recommender Systems. ACM, 2013: 213-220.

[15] SHI Y, LARSON M, HANJALIC A. Unifying rating-oriented and ranking-oriented collaborative filtering for improved recommendation[J]. Information Sciences, 2013, 229(6): 29-39.

[16] Apache. Mahout[EB/OL]. [2016-06-03]. http://mahout.apache.org.

[17] MovieLens datasets [EB/OL]. [2016-06-16]. http://grouplens.org/datasets/movielens.

[18] KOREN Y, BELL R M, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. IEEE Computer, 2009, 42(8): 30-37.