ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2018/12(02)-0197-11
doi: 10.3778/j.issn.1673-9418.1611020

E-mail: fcst@vip.163.com
http://www.ceaj.org
Tel: +86-10-89056056

联合用户兴趣矩阵及全局偏好的推荐算法*

张以文[1,2],艾晓飞[2+],崔光明[2],钱付兰[1,2]
1. 安徽大学 计算智能与信号处理教育部重点实验室,合肥 230031
2. 安徽大学 计算机科学与技术学院,合肥 230601

Recommendation Algorithm with User's Interest Matrix and Global Preference*

ZHANG Yiwen[1,2], AI Xiaofei[2+], CUI Guangming[2], QIAN Fulan[1,2]
1. Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui University, Hefei 230031, China
2. School of Computer Science and Technology, Anhui University, Hefei 230601, China
+ Corresponding author: E-mail: aixiaofei_liu@qq.com

ZHANG Yiwen, AI Xiaofei, CUI Guangming, et al. Recommendation algorithm with user's interest matrix and global preference. Journal of Frontiers of Computer Science and Technology, 2018, 12(2): 197-207.

Abstract: How to recommend the most interested information to users accurately from a large amount of disordered information has become an important research subject in service recommendation system. This paper propose a recommendation algorithm based on the combination of the user's interest matrix and the global preferences, which is used for personalized service recommendation. This paper firstly introduces the interest tag mechanism to form the user interest chain, and to form the user interest matrix by filling unevaluated service and complementing evaluated service on the user service rating set. Then, this paper calculates the partial similarity by the Euclidean distance of the user's interest matrix. Lastly, this paper combines the global preference similarity based on the user cognitive difference and the global behavior difference. The algorithm can effectively integrate the user's preference information, also reduce the sparsity of the data set, and improve the recommendation accuracy. A large number of experiments on the real MovieLens 1M data set show that the proposed algorithm significantly improves the recommenda-

* The National Natural Science Foundation of China under Grant No. 61175046 (国家自然科学基金); the National Key Technology R&D Program of China under Grant No. 2015BAK24B01 (国家科技支撑计划); the Key Project of Nature Science Research for Universities of Anhui Province under Grant No. KJ2016A03 (安徽省高校自然科学基金重点项目).
Received 2016-10, Accepted 2016-12.
CNKI网络优先出版: 2016-12-07, http://www.cnki.net/kcms/detail/11.5602.TP.20161207.0922.016.html

tion accuracy compared with the current representative recommendation algorithms.

Key words: collaborative filtering; interest chain; interest matrix; global preference; similarity

摘 要: 如何从大量无序的信息中向用户准确推荐其最感兴趣的信息,是推荐系统研究领域的重要课题。为此提出一种融合用户兴趣矩阵及全局偏好的推荐算法,用于个性化服务推荐。首先,引入兴趣标签机制形成用户兴趣链,对用户服务评分集合中未评价服务进行填充,对已评价服务进行互补,从而形成用户兴趣矩阵;其次,采用兴趣矩阵的欧几里德距离进行局部相似度计算;最后,联合用户认知差异和全局行为差异形成全局偏好相似度。算法在有效融入了用户的个性化偏好信息的同时,减少了数据集稀疏性,提高了推荐的准确性。在真实的MovieLens 1M数据集上进行的大量实验表明,与当前具有代表性的推荐算法相比,算法显著提高了推荐精度。

关键词: 协同过滤;兴趣链;兴趣矩阵;全局偏好;相似度

文献标志码: A **中图分类号:** TP311

1 引言

随着科技的进步、时代的发展,大量的相似服务和新用户出现在推荐系统中。传统的推荐算法已经很难从如此庞大的候选服务集中选出满足用户个性化偏好的服务。如何从用户的历史行为中更深层次地挖掘出用户的个性化偏好,做出更精确的推荐,已成为研究的热点问题。协同过滤作为应用最为广泛、研究最为深入的推荐算法,已得到广泛应用[1-2],其主要分为基于用户的协同过滤推荐算法和基于项目的协同过滤推荐算法。基于用户的协同过滤推荐算法的核心在于计算用户间的相似度,从而求得目标用户的邻居用户集;基于项目的协同过滤推荐算法的核心在于计算项目之间的相似度,从而求得目标项目的邻居项目集,因此相似度计算是关键。目前常用的相似度计算方法主要有皮尔逊相似度、余弦相似度、欧几里德距离相似度等。

然而,传统的相似度计算方法存在各自的缺陷,例如现有方法往往仅考虑两个用户共同调用的服务集合,未考虑用户的个性化偏好信息等。一般而言,两个用户的服务评分集合中会存在未重合部分,即只有一方调用过的服务集合,而传统的相似度计算方法忽略了这一部分的隐藏偏好。如何深层次地挖掘用户的个性化偏好信息,对于推荐系统推荐精度的提高至关重要。用户个性化偏好信息中比较重要的两种信息分别是用户认知差异和用户行为差异,其中用户认知差异主要反映用户对一系列服务的认同情况,行为差异主要反映用户的不理性打分情况。因此,如何运用好用户的个性化隐藏偏好,对推荐算法的精确度至关重要。

基于上述问题,本文提出了一种融合用户兴趣矩阵及全局偏好的推荐算法,主要贡献如下:

(1)在建立服务兴趣标签机制的基础上,提取出用户兴趣矩阵模型。具体而言,对用户服务评分集合中未评价服务进行填充,对已评价服务进行互补,得到了每位用户的兴趣矩阵,并取每两位用户的兴趣矩阵欧几里德距离作为局部相似度。本文模型可有效地减少数据集稀疏性,同时融入了用户的个性化偏好信息。

(2)引入用户认知差异和用户行为差异机制。认知差异采用余弦相似度计算,行为差异采用用户服务评分集合的平均值和方差来反映,并结合二者作为全局偏好相似度,能够更深层次地挖掘用户的个性化偏好信息。

(3)综合局部相似度和全局偏好相似度形成最终的用户相似度,进行邻居用户选择和评分预测。在MovieLens 1M真实数据集上进行实验,结果表明本文的推荐算法具有更高的精确度和稳定性,同时实现起来简单、快速。

本文组织结构如下:第2章介绍了推荐系统中的相关工作;第3章是问题建模;第4章重点描述了本

文算法框架及其实现过程；第5章给出了实验结果与分析；最后对全文工作进行了总结与展望。

2 相关工作

协同过滤推荐相对成熟并已被广泛应用于各种推荐场景。除此之外，目前流行的推荐技术还包括基于内容的推荐[3-5]以及混合推荐[6]。基于内容的推荐主要是根据用户以往的历史使用记录来推荐相似的服务，它简单易于实现，但效果一般，普适性不高。混合推荐算法主要是使用不同方法结合协同过滤推荐算法和基于内容的推荐算法。

Sarwar等人首次提出基于项目相似性的协同过滤推荐算法[7]，使用Person和余弦作为相似度计算依据。近年来，很多改进算法相继提出。邓爱林等人提出了基于项目评分预测的协同过滤推荐算法[8]，该算法同样基于项目评分填充的思想，运用基于项目的协同过滤算法进行评分填充，再利用基于用户的协同过滤算法产生推荐。Tan等人提出了基于项目分类的协同过滤算法[9]，算法利用项目标签对项目进行聚类，产生多个子数据集，然后在子数据集中进行协同过滤推荐。以上算法虽然实现简单，但并没有融入用户个性化偏好信息，推荐精度较差。

现在也有很多工作倾向于去挖掘出用户更多的个性化偏好。例如：Fletcher等人提出了一种基于用户个性化偏好的推荐系统[10]，为每个用户都定义了一个满意区间，将用户评分映射到该区间内，用映射之后的评分值代替原有评分值进行相似度计算。Ahn提出了一种新的相似度计算方法以缓和冷启动问题[11]，引入了一种混合相似度计算方法，包括Proximity、Impact、Popularity三部分，Proximity考虑了用户评分积极性和消极性的影响，Impact考虑了用户的认知差异，Popularity考虑了用户的行为差异，较好地融入了用户个性化信息。Liu等人提出联合用户信息、服务信息以及服务潜在相关性的服务推荐算法[12]，算法从用户、服务以及服务之间的关系中挖掘用户个性化偏好信息。上述算法较好地融入了用户的个性化偏好信息，但算法实现时间复杂度过高，有的需要额外的训练数据，而且都不能在缓和数据集稀疏性问题的同时实现用户的个性化推荐。为此，本文提出了兴趣矩阵填充互补数据集以减少数据集稀疏性，同时融入了用户个性化信息，最后联合用户全局偏好信息产生更为有效的个性化推荐。

3 问题建模

假设用户集合表示为 $U=\{u_1,u_2,\cdots,u_n\}$，服务集合表示为 $S=\{s_1,s_2,\cdots,s_m\}$，那么可定义 n 个用户对 m 个服务的评分矩阵为 $\boldsymbol{R}=[R_{u_i,s_j}]_{n\times m}$ $(i\in[1,n],j\in[1,m])$。其中 R_{u_i,s_j} 表示用户 u_i 对服务 s_j 的评分，服务的评分区间定义为 $[r_1,r_2]$，用户 u_i 的服务评分集合为 $\boldsymbol{US}_i=\{[s_1,r_1],[s_2,r_2],\cdots,[s_c,r_c]\}$。下面首先给出若干相关定义。

定义1（兴趣标签集合 $\boldsymbol{I}$）兴趣标签集合记为 $\boldsymbol{I}=\{i_1,i_2,\cdots,i_p\}$，其中：

（1）i_k（$k\in[1,p]$）表示第 k 个兴趣标签；

（2）p 表示集合的模长。

针对每一种推荐系统，相应的兴趣标签集合 $\boldsymbol{I}$ 都是不同的。具体而言，电影推荐系统的集合 $\boldsymbol{I}$ 通常包含恐怖、喜剧、爱情、动作等，而服装类推荐系统主要包含颜色种类、材质种类、大小等。一般由服务提供商提供，是规定好的固定集合。

定义2（服务兴趣链 $\boldsymbol{SI}$）服务兴趣链集合记为 $\boldsymbol{SI}=\{[i_1,1],[i_2,1],\cdots,[i_a,1]\}$，其中：

（1）$i_k\in\boldsymbol{I}$（$k\in[1,a]$）表示服务所包含的标签归属于兴趣标签集合 $\boldsymbol{I}$；

（2）$[i_k,1]$（$k\in[1,a]$）表示对兴趣标签 i_k 的偏好值为1；

（3）a 表示集合的模长；

（4）$\boldsymbol{SI}_j$ 表示服务 s_j 的服务兴趣链。

每一个服务都存在一个 $\boldsymbol{SI}$，描述了它们的属性、特征等信息，一般由服务运营商提供。

定义3（用户兴趣链 $\boldsymbol{UI}$）用户兴趣链集合记为 $\boldsymbol{UI}=\{[i_1,p_1],[i_2,p_2],\cdots,[i_b,p_b]\}$，其中：

（1）$i_k\in\boldsymbol{I}$（$k\in[1,b]$）表示用户感兴趣的标签归属于兴趣标签集合 $\boldsymbol{I}$；

（2）$[i_k,p_k]$（$k\in[1,b]$）表示对兴趣标签 i_k 的偏好值为 p_k；

(3)满足 $\sum_{k=1}^{b} p_k = 1$;

(4)b 表示集合的模长;

(5) UI_i 表示用户 u_i 的用户兴趣链。

每一个用户都存在一个 UI,描述了用户对他感兴趣的兴趣标签的偏好值集合,反映了该用户的个性化偏好。

4 基于兴趣矩阵和全局偏好的推荐算法

4.1 推荐算法框架

根据前文的描述,本文的推荐算法主要由两部分组成:

(1)生成兴趣矩阵相似度作为局部相似度;

(2)生成全局偏好相似度。

具体推荐流程如图1所示。

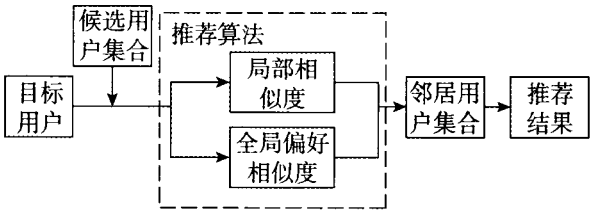

Fig.1 Recommendation algorithm process

图1 推荐算法流程

本文算法具体思路描述如下:

(1)首先基于目标用户得到候选用户集合,即调用过待预测项目的用户集合,并且对目标用户和候选用户集合中的所有用户求得用户兴趣链;其次基于用户的服务评分集合对于目标用户和候选用户集合中的每一位用户分别求出兴趣矩阵,同时对已评分服务进行融合,未评分服务进行填充;最后与目标用户分别求解得到兴趣矩阵的欧几里德距离作为用户之间的局部相似度 $Interest_{(u_i,u_j)}$。

(2)基于用户的服务评分集合,首先求得目标用户与候选用户集合中的每一位用户的余弦相似度作为认知差异;其次求解得到全局行为差异;最后综合认知差异和全局行为差异作为用户之间的全局偏好相似度 $sim(u_x,u_y)_{gi}$。

(3)将用户之间的局部相似度 $Interest_{(u_i,u_j)}$ 和全局偏好相似度 $sim(u_x,u_y)_{gi}$ 相加作为目标用户和候选用户集合中的每一位用户的最终相似度,进而解得邻居用户集合,最终得到预测评分,从而形成推荐。

4.2 兴趣矩阵的建立

4.2.1 用户兴趣链的提取

前文已经提到了兴趣标签通常都是由服务运营商所提供的,并且应用也较为广泛[13-14]。例如网易云音乐中的歌曲就提供了标签信息,可供用户选择;豆瓣电影也提供了诸多的标签信息。因此兴趣标签的提取也较为简单易行,兴趣标签的提取也就是为每个用户建立起用户兴趣链 UI,其生成过程用算法1描述。

算法1 用户兴趣链的生成

输入:待生成用户 u_i 的历史调用服务集合 US_i、服务兴趣链 SI 集合

输出:用户 u_i 的兴趣链 UI_i

```
Begin
    Initialize A=Map[user,Map[tip,interest]]()
    Initialize B=Map[tip,interest]()
    i=0
    For s_j in US_i :
        For int in SI_j :
            i+=1
            If(!B.contains(int))
               B+=(int → 1)
            Else
               B(int)+=1
            End if
        End
        For int in B:
           B(int)=B(int)/i
        End
        A+= (u_i →B)
    End
    Return B
End
```

针对于 US_i 里的每个服务 $s_j(j\in[1,m])$ 都会有相应的服务兴趣链 $SI_j(j\in[1,m])$,累计所有的服务兴趣链的各兴趣标签的偏好值可以得到用户 $u_i(i\in[1,n])$ 的用户兴趣链 UI_i。同时为了标准化,用每个兴趣标签偏好值占总偏好值的比例来代替兴趣标签偏好

值，得到最终用户兴趣链 $UI_i=\{[i_1,p_1],[i_2,p_2],\cdots,[i_b,p_b]\}$，这里 $p_k(k\in[1,b])$ 在[0,1]之间，同时满足 $\sum_{k=1}^{b}p_k=1$。例如用户 u_1 的历史调用服务集合 US_1 为 $\{s_1,s_2,s_3\}$，然后服务 s_1、s_2、s_3 的服务兴趣链 SI_1、SI_2、SI_3 分别为 {[a,1], [b,1], [c,1]}、{[b,1], [c,1], [d,1]}、{[a,1], [c,1], [d,1]}，那么可以得到用户 u_1 的用户兴趣链为 $UI_1=\{[a,2], [b,2], [c,3], [d,2]\}$，标准化为 {[a,0.222], [b,0.222], [c,0.333], [d,0.222]}。

4.2.2 用户兴趣链的提取

用户兴趣链建立完成之后，传统推荐算法在计算两个用户相似度时一般只考虑其共同调用服务部分，而忽略了只有一个用户调用过的服务集合，因此要用隐性评分对另一个用户相应的未调用服务集合进行填充。两个用户之间兴趣矩阵缺失值填充算法如算法2所示。

算法2 两个用户之间兴趣矩阵缺失值填充

输入：用户 u_i 的历史调用服务集合 US_i；用户 u_j 的历史调用服务集合 US_j；服务兴趣链 SI 集合；用户 u_i 的用户兴趣链 UI_i；用户 u_j 的用户兴趣链 UI_j

输出：填充后用户 u_i、u_j 的兴趣矩阵 P_i、P_j；用户 u_i、u_j 的填充部分 C_i、C_j

```
Begin
    Initialize sim= Array[service]()
    For s_j in US_i :
        If (US_j .contains (s_j) )
            sim+= s_j
End if
    End
    Initialize P_i = P_j =Map[service,rating]()
    Initialize C_i = C_j =Array[service]()
    P_i = US_i ; P_j = US_j ; I=J=0
    For s_j in P_i :
        If (! P_j .contains (s_j) )
            C_j += s_j
            For int in SI_j :
                J+= UI_j (int)
            End
            P_j += (s_j → J)
        End if
    End
    For s_j in P_j :
        If (! P_i .contains (s_j) )
            C_i += s_j
            For int in SI_j :
                I+= UI_i (int)
            End
            P_i += (s_j → I)
        End if
    End
    Return P_i 、P_j 、C_i 、C_j
End
```

其中，C_i 表示用户 u_i 需要评分填充的部分，即在用户 u_i 和 u_j 的服务评分集合之间，用户 u_j 评分过而 u_i 未评分过的项目。C_j 同理。I 和 J 分别表示属于用户 u_i 和 u_j 的隐性评分。算法实现机制主要是针对填充部分中的待填充服务 s_j。首先遍历 s_j 的服务兴趣链 SI_j，对于每一个兴趣标签，从该用户的用户兴趣链中找出该用户对其的偏好值，全部累加可得到该用户对待填充服务的隐性评分，用隐性评分填充该服务评分值。

4.2.3 兴趣矩阵的互补以及局部相似度的生成

获得了填充之后的用户兴趣矩阵之后，还要对用户调用过的项目进行互补，即加入隐性评分来使用户的评分更加个性化，从而更全面地反映用户的兴趣偏好。并取最后的兴趣矩阵的欧几里德距离作为局部相似度。用户间完全兴趣矩阵及局部相似度生成算法如算法3所示。

算法3 完全兴趣矩阵以及局部相似度的生成

输入：填充后的用户 u_i、u_j 的兴趣矩阵 P_i、P_j；用户 u_i、u_j 的填充部分 C_i、C_j

输出：完全用户兴趣矩阵 I_i、I_j 以及局部相似度 $Interest_{(u_i,u_j)}$

```
Begin
    Initialize I_i = I_j =Map[service,rating]()
    I_i = P_i ; I_j = P_j
    b=1−sim.length/ I_i .length
```

$Interest_{(u_i,u_j)}$=0.0
For s_j in I_i:
If (! C_i .contains (s_j))
I_i (rating)= I_i (rating)/ r_2
I_i (rating)=(I_i (rating)+J)/2
End if
End
For s_j in I_j:
If (! C_j .contains (s_j))
I_j (rating)= I_j (rating)/r_2
I_j (rating)= (I_j (rating)+J)/2
End if
End
For s_j in I_i:
If (sim.contains (s_j))
$Interest_{(u_i,u_j)}$ + = pow((I_i (service)−
I_j (service)),2)
Else
$Interest_{(u_i,u_j)}$ + = exp(b)*pow((I_i (service)−
I_j (service)),2)
End if
End
Return I_i 、I_j 、$Interest_{(u_i,u_j)}$
End

这里采用用户的评分即显性评分和隐性评分取平均值的方法作为用户最终的评分信息。又因为显性评分信息的区间是在$[r_1,r_2]$之间，所以这里除以r_2将显性评分标准化到[0,1]之间。同时在求解欧几里德距离时，也考虑到了共同调用服务部分所占比例的影响。具体措施为：在计算服务s_j时，若该服务是共同调用的服务，那么评分差值系数为1；相反若是单个用户调用的项目，评分差值的系数为exp(1−共同调用服务比例)。

4.3 全局偏好

4.3.1 基于余弦相似度的用户认知差异

传统的相似度计算方法往往只考虑了用户评分之间的差异性，而忽略了用户对服务集合的认知差异，这里引入了余弦相似度来反映用户之间的认知差异。余弦相似度主要利用两个用户之间的服务评分集合所形成的评分矩阵的夹角余弦值来反映它们的认知程度，也即是它们的偏离程度。例如用户u_1和u_2的评分矩阵的相同部分分别为[3,3,3]和[5,5,5]，那么它们的偏离程度为0，也就是他们的认知程度完全一样，他们认为这3样物品同样好。其计算方式如式(1)：

$$\cos_{(u_x,u_y)}=\frac{\sum_{i\in U}R_{(u_x,i)}R_{(u_y,i)}}{\sqrt{\sum_{i\in U}(R_{(u_x,i)})^2}\sqrt{\sum_{i\in U}(R_{(u_y,i)})^2}} \tag{1}$$

其中，$\cos_{(u_x,u_y)}\in[0,1]$；U表示两个用户的共同调用服务集合；$R_{(u_x,i)}$表示用户u_x对服务i的评分。这里$\cos_{(u_x,u_y)}$和用户相似度成正比。

4.3.2 基于全局行为的用户偏好

同时也注意到在实际运行的推荐系统中，肯定存在这样一群用户，他们拥有自己的打分习惯，不管这样物品有多么令他不满意，他还是会打一个不低的分数；相反的不管这样物品多么令他满意，他也只是会打一个不高的分数，这种用户全局行为的差异也要在推荐系统中反映出来。另外，一个用户也可能是一个极其随意的用户，根据他当时的心情随意地打分，这样的用户难以成为试验对象，他的评分集合方差势必极大，因此也要将这些用户和正常的用户区分开来。为此引入了全局行为误差[15]，其计算方式如式(2)：

$$sim_{(u_x,u_y)}^{\text{global}}=1-\frac{1}{1+\exp(-|\overline{R_x}-\overline{R_y}|\cdot|\overline{A_x}-\overline{A_y}|)} \tag{2}$$

其中，$\overline{R_x}$代表用户x的平均值；$\overline{A_x}$代表用户x的方差。这里两个用户之间平均值、方差悬殊越大，相似度越小。

4.3.3 全局偏好的生成

最后用户全局偏好相似度的生成应该基于用户认知程度差异和全局行为差异，这里采用的是将两者相加的方法，因为两者都是与用户相似度正相关，所以相加后也是与用户相似度正相关。同时也考虑到因为余弦相似度存在为0的情况，此时两个用户之间认知程度无法估计，所以为0，如果采用相乘的话，会导致整个用户全局偏好差异也为0，就抹杀了全局

行为差异。最后的用户全局偏好差异计算方式如下：

$$sim(u_x,u_y)=\cos_{(u_x,u_y)}+sim_{(u_x,u_y)}^{\text{global}} \tag{3}$$

5 实验结果与分析

5.1 数据集

本文实验采用的是GroupLens提供的MovieLens数据集作为各个对比算法的训练集以及测试集。它包含了6 040个用户对3 900部电影的1 000 209条评分信息，它是一个评分数据集，其中用户的评分区间在[1,5]之间。

同时为了模拟数据集的稀疏性问题，本文随机分割数据集，使得训练集和测试集的比例分别为5:5、6:4、7:3、8:2、9:1。同时在邻居用户数量为5、10、15、20、25时分别进行实验，得到最终结果。

5.2 评测指标

本文实验采用了已经获得的评分集合进行离线训练与评估，同时为保证从各个方面来验证本文算法的准确性，划分了多个训练集与测试集的比例，也在不同邻居用户数量的基础上进行全面实验。主要采用以下两种常用的评测指标。

(1)平均绝对误差(mean absolute error，MAE)[16]

$$MAE=\frac{1}{N}\sum_{n=1}^{N}|R_{u,r}-R_{u,r}^{p}| \tag{4}$$

(2)均方根误差(root mean squared error，RMSE)[17]

$$RMSE=\sqrt{\frac{1}{N}\sum_{n=1}^{N}(R_{u,r}-R_{u,r}^{p})^2} \tag{5}$$

其中，N代表测试集要测试的服务数量；$R_{u,r}$代表用户u对服务v的真实评分；$R_{u,r}^{p}$代表推荐系统所推荐的分数。

5.3 实验结果分析

以下实验主要是对本文的推荐算法(dem)与相似度计算方法为皮尔逊相似度(person)、余弦相似度(cos)、欧几里德相似度(em)3种传统的基于用户的协同过滤推荐算法进行比较，验证算法的精确度和稳定性。

(1)邻居用户数量的影响

为验证本文算法在精确度上的提升，在设定训练集与测试集不同比例的情况下，变化邻居用户数量从5到25，分别计算求得4种算法的MAE和RMSE，实验结果如图2~图11所示。

从上述实验结果图中可以看出，在相同的训练集和测试集比例下，对于不同邻居数量，本文算法的MAE和RMSE均明显低于其他算法，因此本文算法具有更好的精确度。

这是因为本文在用户显性评分中融入基于用户兴趣链得到的隐性评分，从而综合得到了用户最终的评分。基于此求得了用户兴趣矩阵，这样就有效减少了数据集稀疏度，并且使得用户的评分更加符合用户的个性化偏好。同时用户认知差异的引入使得那些认知相似但是评分差距较大的用户变得更加相似，全局行为差异的加入也剔除了那些随意打分或者打分习惯不好的用户。因此本文推荐算法的精确度有了较大提升。最后随着邻居用户数量的增加，4种算法误差逐步减小，并趋于稳定，这是样本容量增加的结果。

(2)训练集与测试集比例的影响

为测试本文算法能在多大程度上缓和数据集稀疏性问题，同时验证算法稳定性，本文在固定邻居用

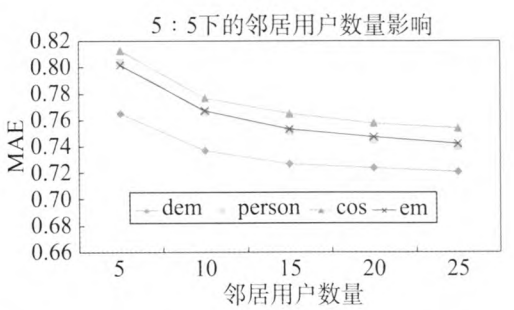

Fig.2 Change of MAE on train set:test set=5:5

图2 训练集和测试集比例为5:5时MAE变化

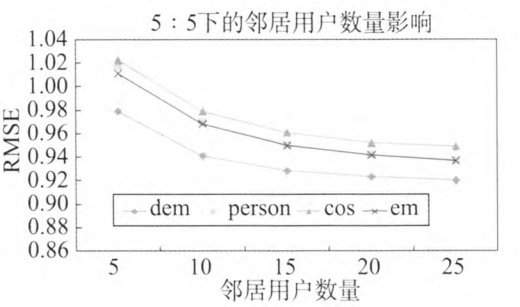

Fig.3 Change of RMSE on train set:test set=5:5

图3 训练集和测试集比例为5:5时RMSE变化

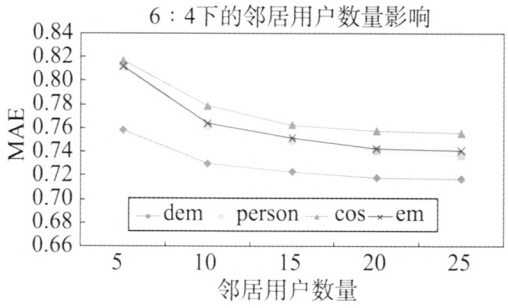

Fig.4 Change of MAE on train set∶test set=6∶4

图4 训练集和测试集比例为6∶4时MAE变化

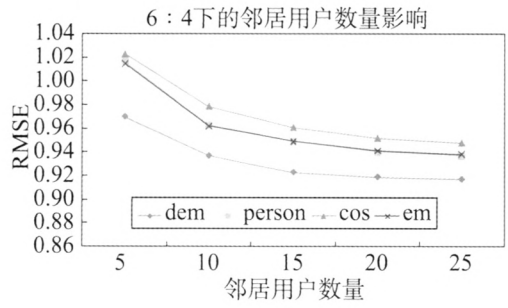

Fig.5 Change of RMSE on train set∶test set=6∶4

图5 训练集和测试集比例为6∶4时RMSE变化

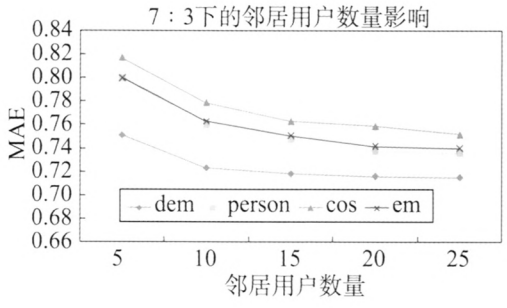

Fig.6 Change of MAE on train set∶test set=7∶3

图6 训练集和测试集比例为7∶3时MAE变化

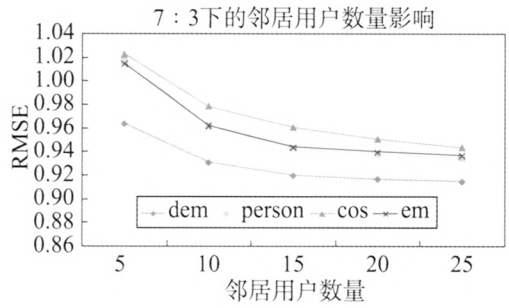

Fig.7 Change of RMSE on train set∶test set=7∶3

图7 训练集和测试集比例为7∶3时RMSE变化

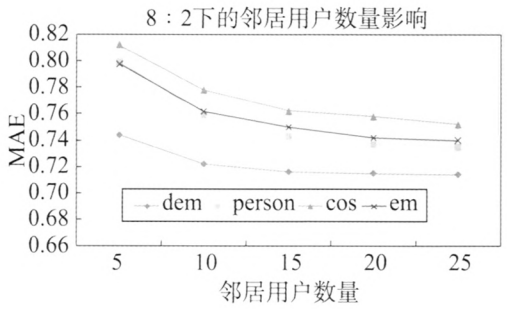

Fig.8 Change of MAE on train set∶test set=8∶2

图8 训练集和测试集比例为8∶2时MAE变化

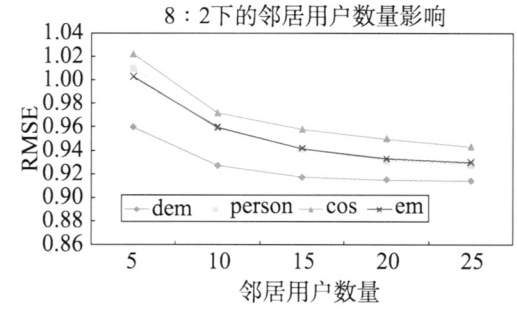

Fig.9 Change of RMSE on train set∶test set=8∶2

图9 训练集和测试集比例为8∶2时RMSE变化

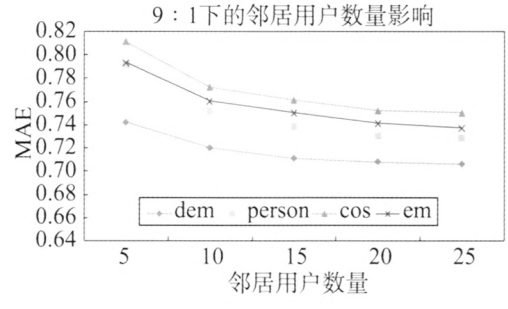

Fig.10 Change of MAE on train set∶test set=9∶1

图10 训练集和测试集比例为9∶1时MAE变化

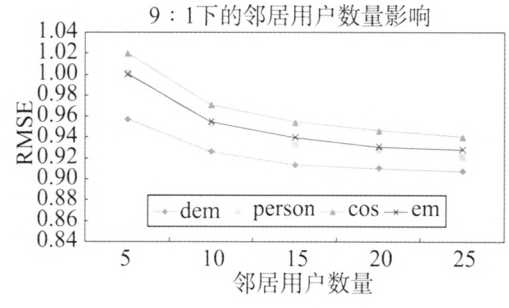

Fig.11 Change of RMSE on train set∶test set=9∶1

图11 训练集和测试集比例为9∶1时RMSE变化

Table 1 Effect of dataset sparsity on algorithm stability

表1 数据集稀疏性对算法稳定性的影响

邻居用户数	对比算法	50%		60%		70%		80%		90%	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
N=5	dem	**0.765 7**	**0.979 1**	**0.757 9**	**0.971 0**	**0.752 4**	**0.966 3**	**0.747 8**	**0.959 5**	**0.744 1**	**0.957 2**
	person	0.804 8	1.017 7	0.803 3	1.016 0	0.801 5	1.014 1	0.799 2	1.011 0	0.793 7	1.005 2
	cos	0.813 5	1.024 8	0.815 5	1.026 3	0.815 3	1.026 2	0.813 8	1.022 8	0.812 5	1.021 4
	em	0.802 2	1.012 2	0.802 1	1.011 0	0.801 4	1.009 6	0.797 5	1.003 6	0.795 1	1.001 8
N=10	dem	**0.736 9**	**0.940 9**	**0.730 7**	**0.9356**	**0.726 4**	**0.931 7**	**0.722 3**	**0.926 5**	**0.720 1**	**0.925 6**
	person	0.765 6	0.968 7	0.763 2	0.966 1	0.761 1	0.963 6	0.758 5	0.959 8	0.752 2	0.952 7
	cos	0.777 7	0.978 6	0.778 7	0.978 7	0.778 9	0.978 5	0.777 0	0.974 7	0.774 6	0.972 0
	em	0.766 2	0.966 7	0.765 5	0.964 4	0.764 9	0.962 6	0.763 1	0.959 1	0.760 0	0.955 8
N=15	dem	**0.727 9**	**0.928 8**	**0.722 3**	**0.924 0**	**0.718 0**	**0.920 1**	**0.714 6**	**0.916 3**	**0.712 1**	**0.914 5**
	person	0.751 8	0.952 2	0.748 7	0.948 2	0.746 6	0.945 7	0.743 8	0.941 7	0.738 0	0.934 9
	cos	0.764 9	0.962 2	0.765 2	0.961 8	0.765 4	0.961 3	0.763 5	0.957 7	0.761 1	0.955 1
	em	0.753 0	0.950 6	0.752 5	0.948 6	0.752 1	0.946 9	0.750 2	0.942 9	0.747 5	0.940 2
N=20	dem	**0.723 7**	**0.923 1**	**0.718 3**	**0.918 4**	**0.714 3**	**0.914 9**	**0.711 4**	**0.911 3**	**0.708 7**	**0.909 4**
	person	0.744 5	0.943 7	0.741 2	0.939 3	0.738 6	0.936 1	0.736 1	0.932 3	0.731 1	0.926 5
	cos	0.758 2	0.953 7	0.758 4	0.953 3	0.758 2	0.952 4	0.756 6	0.948 9	0.754 0	0.946 0
	em	0.746 0	0.942 4	0.745 6	0.940 4	0.745 4	0.938 6	0.743 3	0.934 7	0.741 3	0.932 5
N=25	dem	**0.721 6**	**0.920 1**	**0.716 2**	**0.915 4**	**0.712 5**	**0.912 3**	**0.709 4**	**0.908 4**	**0.707 0**	**0.906 5**
	person	0.740 4	0.938 9	0.736 9	0.934 2	0.733 7	0.930 4	0.731 2	0.926 5	0.726 3	0.921 0
	cos	0.753 8	0.948 4	0.754 0	0.947 8	0.753 7	0.946 9	0.752 4	0.943 8	0.749 6	0.940 7
	em	0.741 8	0.937 7	0.741 3	0.935 5	0.741 0	0.933 7	0.739 3	0.929 9	0.736 9	0.927 5

户数量情况下，分别求得在训练集占不同比例情况下4种算法的MAE和RMSE，实验结果如表1所示。

从表1中可知，在训练集所占比例相同的情况下，本文算法拥有最好的推荐精确度。同时随着训练集所占比例的增加，本文算法精确度稳定提升，而其他算法基本不变化甚至有所下降。

这是因为本文利用用户兴趣链生成了用户的隐性评分，对用户已评分服务进行糅合，未评分服务进行填充，有效降低了数据集稀疏性；并且使得用户评分更加真实有效，符合用户兴趣。同时全局行为差异的引入将那些不可靠用户剔除，有效地选取邻居用户。因此本文算法具有较好的稳定性，不会受到数据集稀疏性较大的影响，同时又具有较好的精确度。

综上所述，相对于传统的推荐算法而言，本文算法具有更高的精确性和稳定性，能够充分地从用户的服务评分集合中挖掘出用户个性化偏好信息。

6 总结

本文在利用用户兴趣链对用户服务评分集合进行填充互补，形成用户兴趣矩阵基础上，基于用户兴趣矩阵的欧几里德距离进行局部相似度计算，运用余弦相似度形成认知差异和全局行为差异构成全局偏好相似度，并联合局部相似度和全局偏好相似度形成最后的用户相似度得到邻居用户，产生推荐结果。相对于传统的协同过滤算法，本文推荐算法在缓和了数据集稀疏性的同时，可有效地进行用户个性化推荐。接下来的研究中，进一步验证本文算法在大数据环境下的有效性和稳定性。

References:

[1] Bobadilla J, Ortega F, Hernando A, et al. Recommender systems survey[J]. Knowledge-Based Systems, 2013, 46(1): 109-132.

[2] Jiang Shuhui, Qian Xueming, Shen Jialie, et al. Author topic model-based collaborative filtering for personalized POI

recommendations[J]. IEEE Transactions on Multimedia, 2015, 17(6): 907-918.

[3] Adomavicius G, Tuzhilin A. Context-aware recommender systems[M]//Ricci F, Rokach L, Shapira B, et al. Recommender Systems Handbook. Boston: Springer, 2011: 217-253.

[4] Zheng Yong, Mobasher B, Burke R D. The role of emotions in context-aware recommendation[C]//Proceedings of the 3rd Workshop on Human Decision Making in Recommender Systems in Conjunction with the 7th ACM Conference on Recommender Systems, Hong Kong, China, Oct 12, 2013. New York: ACM, 2013: 21-28.

[5] Lu Zhongqi, Dou Zhicheng, Lian Jianxun, et al. Content-based collaborative filtering for news topic recommendation [C]//Proceedings of the 29th Conference on Artificial Intelligence, Austin, Jan 25-30, 2015. Menlo Park: AAAI, 2015: 217-223.

[6] de Campos L M, Fernández-Luna J M, Huete J F, et al. Combining content-based and collaborative recommendations: a hybrid approach based on Bayesian networks[J]. International Journal of Approximate Reasoning, 2010, 51(7): 785-799.

[7] Sarwar B M, Karypis G, Konstan J A, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th International World Wide Web Conference, Hong Kong, China, May 1-5, 2001. New York: ACM, 2001: 285-295.

[8] Deng Ailin, Zhu Yangyong, Shi Baile. A collaborative filtering recommendation algorithm based on item rating prediction [J]. Journal of Software, 2003, 14(9): 1621-1628.

[9] Tan Hengsong, Ye Hongwu. A collaborative filtering recommendation algorithm based on item classification[C]//Proceedings of the 2009 Pacific-Asia Conference on Circuits, Communications and Systems, Chengdu, May 16-17, 2009. Washington: IEEE Computer Society, 2009: 694-697.

[10] Fletcher K K, Liu X F. A collaborative filtering method for personalized preference-based service recommendation[C]// Proceedings of the 2015 International Conference on Web Services, New York, Jun 27-Jul 2, 2015. Washington: IEEE Computer Society, 2015: 400-407.

[11] 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.

[12] Liu Xumin, Fulia I. Incorporating user, topic, and service related latent factors into Web service recommendation[C]// Proceedings of the 2015 International Conference on Web Services, New York, Jun 27-Jul 2, 2015. Washington: IEEE Computer Society, 2015: 185-192.

[13] Sun Hongfei, Wu Huijuan, Zhou Lanping. Research on personalized information recommendation model based on the label[J]. Information Science, 2013, 31(4): 24-27.

[14] Zheng Yong, Mobasher B, Burke R D. Context recommendation using multi-label classification[C]//Proceedings of the 2014 International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, Warsaw, Aug 11-14, 2014. Washington: IEEE Computer Society, 2014: 288-295.

[15] Liu Haifeng, Hu Zheng, Mian A, et al. A new user similarity model to improve the accuracy of collaborative filtering[J]. Knowledge-Based Systems, 2014, 56(3): 156-166.

[16] Tang Mingdong, Dai Xiaoling, Cao Buqing, et al. WSWalker: a random walk method for QoS-aware Web service recommendation[C]//Proceedings of the 2015 International Conference on Web Services, New York, Jun 27-Jul 2, 2015. Washington: IEEE Computer Society, 2015: 591-598.

[17] Zhou Zuojian, Wang Binbin, Guo Jie, et al. QoS-aware Web service recommendation using collaborative filtering with PGraph[C]//Proceedings of the 2015 International Conference on Web Services, New York, Jun 27-Jul 2, 2015. Washington: IEEE Computer Society, 2015: 392-399.

附中文参考文献:

[8] 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.

[13] 孙鸿飞, 武慧娟, 周兰萍. 基于标签的个性化信息推荐理论模型研究[J]. 情报科学, 2013, 31(4): 24-27.

ZHANG Yiwen was born in 1976. He received the Ph.D. degree in management science and engineering from Hefei University of Technology in 2013. Now he is an associate professor at School of Computer Science and Technology, Anhui University, and the member of CCF. His research interests include service computing, cloud computing and swarm intelligence, etc.

张以文(1976—),男,安徽马鞍山人,2013年于合肥工业大学获得博士学位,现为安徽大学计算机科学与技术学院副教授、硕士生导师,CCF会员,主要研究领域为服务计算,大数据,群体智能等。

AI Xiaofei was born in 1994. He is an M.S. candidate at School of Computer Science and Technology, Anhui University. His research interests include service computing and service recommendation, etc.
艾晓飞(1994—)，安徽马鞍山人，安徽大学计算机科学与技术学院硕士研究生，主要研究领域为服务计算，服务推荐等。

CUI Guangming was born in 1991. He is an M.S. candidate at School of Computer Science and Technology, Anhui University. His research interests include service computing and evolutionary computing, etc.
崔光明(1991—)，男，安徽阜阳人，安徽大学计算机科学与技术学院硕士研究生，主要研究领域为服务计算，进化计算等。

QIAN Fulan was born in 1978. She received the Ph.D. degree in computer application from Anhui University in 2016. Now she is a lecturer at School of Computer Science and Technology, Anhui University, and the member of CCF. Her research interests include quotient space theory, social network and recommender system, etc.
钱付兰(1978—)，女，安徽蚌埠人，2016年于安徽大学获得博士学位，现为安徽大学计算机科学与技术学院讲师，CCF会员，主要研究领域为商空间，社交网络，推荐系统等。