计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2018.20170252
55(7):1508-1524,2018

一种基于特征的协同聚类模型

张 飞[1] 张立波[2] 罗铁坚[1] 武延军[2]
[1](中国科学院大学 北京 101408)
[2](中国科学院软件研究所 北京 100190)
(tjluo@ucas.ac.cn)

A Feature-Based Co-Clustering Model

Zhang Fei[1], Zhang Libo[2], Luo Tiejian[1], and Wu Yanjun[2]
[1](*University of Chinese Academy of Sciences*, *Beijing* 101408)
[2](*Institute of Software*, *Chinese Academy of Sciences*, *Beijing* 100190)

Abstract Recommendation system can effectively solve the personalized recommendation problem for users. As one of the most commonly used algorithm in recommendation system, collaborative filtering needs to take all the items into account, while a specific user may be only interested in the items in some certain domains. It's more natural to make recommendation for a user via the correlated domains than the entire items, therefore, users and items can be grouped according to their interests or characteristics, and then the recommendations can be made with the user-item subgroups. Based on this idea, we propose a novel co-clustering method based on the features of users and items to find the meaningful subgroups. The proposed method includes two main modules: a feature representation model to explore the interests of the users and the characteristics of the items, and a graph model constructed in accordance with these features for coming up with the final clustering results which are used for making recommendation. In this paper, we introduce the framework of our method and give an effective solution to get the features and the clustering results. Finally, by comparing with a variety of newest algorithms on three open datasets, we verify that the proposed method can significantly improve the accuracy of recommender system.

Key words recommender system; clustering model; collaborative filtering; feature representation; graph model

摘 要 推荐系统能够有效解决用户的个性化推荐问题,其中,协同过滤是近年来的主流方法.协同过滤算法具有一定的局限性,因为需要在全部的物品中为用户进行推荐,而单个用户往往只对某些领域的物品感兴趣.为了解决这个问题,提出了一种新的协同聚类模型,先将用户和物品根据兴趣或特征进行聚类分组,然后在每个分组上进行相应的推荐.该模型主要包含2个模块:1)特征表示模块,用以发掘用户的兴趣和物品的特征;2)根据该特征构建的图模型,用来求解最终的聚类分组.通过在3种公开数据集上与其他算法进行性能比较,验证了这种协同聚类模型能够显著提高推荐系统预测与推荐的准确度.

收稿日期:2017-04-10;**修回日期**:2017-09-05
基金项目:中国科学院系统优化基金项目(Y42901VED2,Y42901VEB1,Y42901VEB2)
This work was supported by the System Optimization Project of Chinese Academy of Sciences (Y42901VED2, Y42901VEB1, Y42901VEB2).
通信作者:张立波(zsmj@hotmail.com)

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

关键词　推荐系统；聚类模型；协同过滤；特征表示；图模型

中图法分类号　TP391

互联网的蓬勃发展促使信息总量呈现指数上升，这使得互联网用户从庞杂的信息中获取有用的信息变得越来越困难. 推荐系统能够根据用户的特点和兴趣有针对性地为他们推荐有价值的信息. 因此推荐系统的作用越来越突出，已经被应用到商品推荐等诸多领域中.

协同过滤(collaborative filtering, CF)[1-3]算法是推荐系统中一种被广泛采用的算法. 与传统基于内容(content-based)[4]的推荐算法不同，协同过滤算法不需要用户的个人资料以及物品本身的信息，仅仅利用用户和物品之间的交互关系(如点击、浏览、评分等)就能做出准确的推荐. 在用户越来越重视个人信息安全的形势下，协同过滤算法相比基于内容的推荐算法具有更大的优势.

传统的协同过滤算法在取得一定性能的同时仍然存在一些问题. 例如数据的高稀疏性使得算法的推荐性能难以提高，对此研究人员提出一种基于聚类的协同过滤算法，通过对用户和物品进行聚类，将有相同爱好的用户和相同特征的物品分为一组，然后在各个分组上分别利用协同过滤算法进行推荐. 但是，大部分聚类方法都只允许用户或者物品属于一个分组，而现实中一个用户可能有多种爱好并且一个物品可能具备多种特征，因此只允许用户和物品属于一个分组具有很大的局限性.

基于上述考虑，本文提出了一种基于特征的协同聚类模型，它充分利用用户与物品之间的交互信息来挖掘它们的潜在特征，并根据这些特征将他们分配到合适的多个分组中. 例如某用户同时喜欢喜剧电影与动作电影，则应该将该用户分配到喜剧和动作这 2 个分组中去. 这种聚类方式充分利用了用户和物品的特征，使得聚类的结果更加可信. 最后将本文算法得到的聚类结果使用传统的协同过滤算法进行推荐，通过与多种最新算法在 3 种公开数据集上的比较，验证了本文提出的模型能够显著提高预测与推荐的准确度.

本文的主要贡献有 2 个方面：

1) 提出一种快速提取用户以及物品特征的方法以及给出该方法的求解过程；

2) 提出一种图模型将用户评分信息与特征信息结合，用来计算最终的聚类结果.

1　相关工作

Breese 等人[5]将协同过滤推荐算法分为 2 种类型：基于内存(memory-based)的协同过滤算法与基于模型(model-based)的协同过滤算法.

基于内存的协同过滤算法主要是通过寻找用户[6]或者物品[7]之间的相似性来进行推荐，虽然这种推荐系统实现简单并且很有效，但是它还存在很多问题，比如无法有效处理大规模数据以及高稀疏性的数据. 为了解决基于内存的协同过滤算法的这种问题，许多基于模型的协同过滤算法被提出，比如贝叶斯模型[8]、基于回归的模型[9]、潜在语义模型[10]、聚类模型[11-12]和矩阵分解模型[13]. 在这些模型中，矩阵分解模型被广泛使用，比如著名的 SVD (singular value decomposition)[14]模型、NMF(non-negative matrix factorization)[15]模型、MMMF (maximum margin matrix factorization)[16]、NPCA (nonparametric probabilistic principal component analysis)[17]等，它们在处理大规模数据时有着杰出的表现，但是推荐系统数据高稀疏性的特点仍旧限制了这些算法性能.

为了克服数据的高稀疏性等问题，基于聚类的协同过滤算法被提出. 聚类的协同过滤算法首先将用户与物品进行适当的聚类分组，然后在每个分组上分别使用协同聚类算法进行推荐. 由于互联网的发展，越来越多的算法在对用户物品进行聚类时，加入了很多额外的信息来增强聚类性能[18-22]，比如产品分类信息[23]、用户信任网络[24]等. 但是这些额外的信息有时候并不能增加聚类的准确度，比如某用户购买了某件商品，他的好友们可能不会对这件商品感兴趣，因此本文的聚类过程仅仅使用了隐式反馈信息，如用户物品评分. 这种只使用隐式反馈信息的聚类算法一般分为 3 种：基于用户的聚类、基于物品的聚类以及双侧聚类.

基于用户的聚类算法是根据用户的相似度将最相似的若干用户分为一组，在为具体的用户进行推荐时只考虑用户所在的分组. 例如 Sarwar 等人[25]就是将全部的用户根据相似度进行聚类，并将每一个聚类结果中的用户视为邻居；Xue 等人[26]则是通过训练数据将用户进行聚类，为每个用户挑选出最

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

好的若干邻居并学习出预测矫正信息,然后利用这些矫正信息以及邻居来为具体的用户进行推荐;相反,基于物品的聚类算法则是根据物品之间的相似度来进行聚类,例如 O'Connor 等人[27]提出的方法.

除了上述的一些单侧聚类算法外,还有一些基于双侧聚类的聚类算法,这与本文的主要方法一致,Ungar 等人[28]通过使用吉比斯采样与 k-means 方法对用户与物品进行聚类,该方法在模拟数据上有一定的效果,但是在真实数据上却不乐观;George 等人[29]则是对用户物品同时聚类,然后根据聚类分组中的平均打分以及用户自身的矫正量来进行最后的推荐;除了以上这些单类聚类算法外,Xu 等人[30]还提出了一种多类协同聚类算法,它与本文核心思想一致:允许用户和物品同时属于多个分组.但是它的聚类原理比较片面,仅仅考虑用户对物品的评分而忽略了用户自身的特点与物品自身的特点.

2 基于特征的协同聚类模型

本文提出的基于特征的协同聚类算法主要框架及其在推荐系统中的作用如图 1 所示,主要包括 2 个模块:1)一个特征表示模型,用来挖掘用户的兴趣以及物品的特征;2)一个图模型,利用挖掘出来的特征以及用户的评分信息计算最终的聚类结果.在推荐系统中,可以先利用本文算法完成数据的聚类工作,再将聚类的结果应用到推荐模型中,从而提高最终的推荐性能.

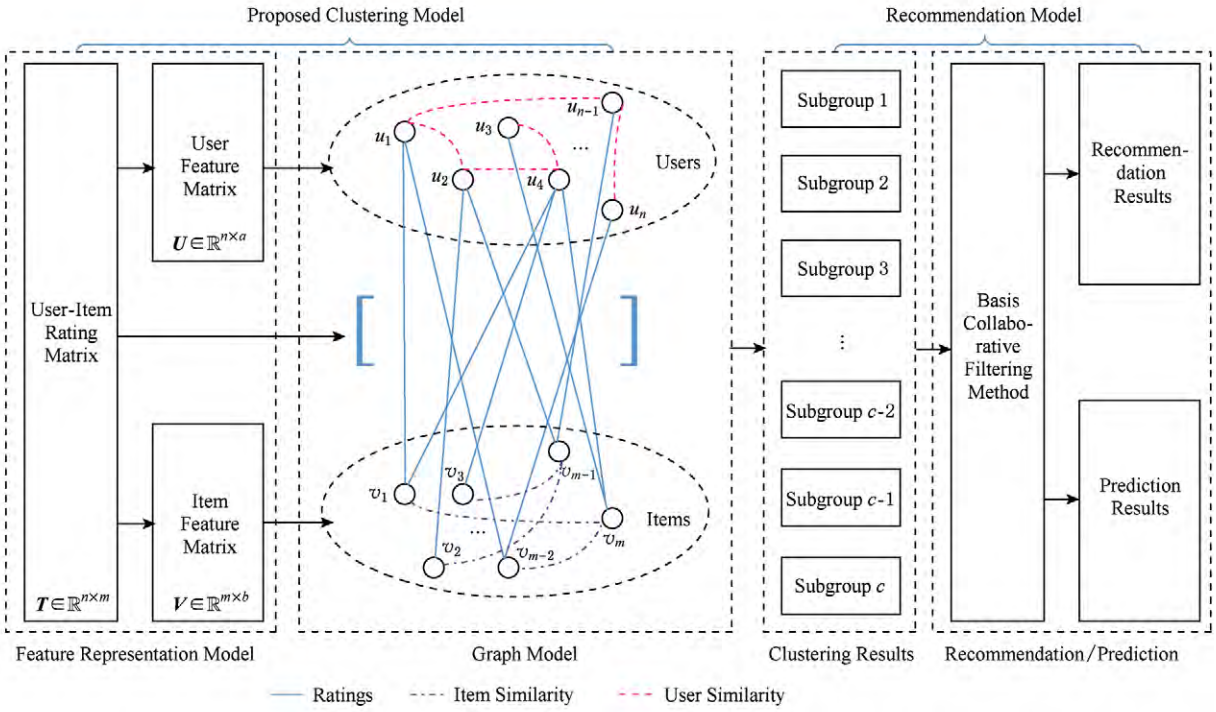

Fig. 1 The framework of the proposed method and its role in the recommender system

图 1 本文算法的框架及其在推荐系统中的作用

2.1 相似度的计算

通过图 1 可以发现在本文模型中需要计算用户相似度与物品相似度,在推荐系统中,著名的基于相似度的推荐算法如 Item-Based[7],User-Based[6]等所采用的相似度算法主要是余弦相似度、Pearson 相关系数以及约束 Pearson 相关系数等,它们的计算方法如表 1 所示.

在表 1 中,s 和 t 表示用户集合中的任意 2 位用户且 $s\neq t$,I_{st} 表示 s 和 t 共同评分过的物品集合,$\overline{T}_s$ 和 $\overline{T}_t$ 分别表示用户 s 和 t 在 I_{st} 上的平均打分,T_{med} 表示推荐系统所采用的评分制中值.

从表 1 可以看出,当 $I_{st}=\varnothing$ 时,也即用户 s 和 t 之间不存在共同打分过的物品,那么这 3 种相似度算法都无法计算 s 和 t 的相似度;另一方面当 2 个用户共同打分过的物品较少甚至只有 1 个时,仅仅利用这些少量的打分信息来判断用户之间的相似度显然是不够准确的.而在推荐系统中,其数据的高稀疏性是其最大的特点,2 个用户之间共同打分过的物品集合 I_{st} 往往都比较少甚至为空,因此,采用上述方法计算得到的相似度不够可靠,为了解决这个问题,本文采用一种新的特征表示模型来挖掘用户

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

以及物品的特征信息，然后用这些特征信息来计算相似度，我们将在 2.2 节讨论如何提取特征.

Table 1　Measures of Similarity[31]

表 1　相似度的计算方法[31]

Measures	Expression
Cosine Similarity	$sim(s,t)=\dfrac{\sum_{i\in I_{st}} T_{si}T_{ti}}{\sqrt{\sum_{i\in I_{st}} T_{si}^2}\sqrt{\sum_{i\in I_{st}} T_{ti}^2}}$
Pearson Correlation Coefficient	$sim(s,t)=\dfrac{\sum_{i\in I_{st}} (T_{si}-\overline{T}_s)(T_{ti}-\overline{T}_t)}{\sqrt{\sum_{i\in I_{st}} (T_{si}-\overline{T}_s)^2}\sqrt{\sum_{i\in I_{st}} (T_{ti}-\overline{T}_t)^2}}$
Constrained Pearson Correlation Coefficient	$sim(s,t)=\dfrac{\sum_{i\in I_{st}} (T_{si}-T_{\mathrm{med}})(T_{ti}-T_{\mathrm{med}})}{\sqrt{\sum_{i\in I_{st}} (T_{si}-T_{\mathrm{med}})^2}\sqrt{\sum_{i\in I_{st}} (T_{ti}-T_{\mathrm{med}})^2}}$

2.2　特征信息的提取

在本节我们将介绍图 1 中的特征表示模块，该部分用来提取用户与物品的特征以便计算它们的相似度.

设 n 个用户对 m 个物品的评分矩阵为 $\boldsymbol{T}\in \mathrm{R}^{n\times m}$，用户的特征信息矩阵为 $\boldsymbol{U}\in \mathrm{R}^{n\times a}$，其中第 u 行的 a 维行向量 $\boldsymbol{U}_u$ 表示第 u 个用户的特征信息向量，a 为一个较小的指定的特征信息维度；同样地，用矩阵 $\boldsymbol{V}\in \mathrm{R}^{m\times b}$ 表示全部物品的特征信息矩阵，b 表示物品特征信息的维度. 我们将通过二次回归模型用第 u 个用户的特征信息 $\boldsymbol{U}_u$ 以及第 v 个物品的特征信息 $\boldsymbol{V}_v$ 去拟合评分 T_{uv}.

在传统的二次回归模型中，对于特征向量 $\boldsymbol{x}$，其对应的监督值 y 通过表达式去拟合：

$$\hat{y}=z+\sum_{i=1}^{l} w_i x_i+\sum_{i=1}^{l-1}\sum_{j=i+1}^{l} W_{ij}x_i x_j, \tag{1}$$

其中，z 表示常数项系数，w 表示一次项系数，W 表示二次项系数，l 则表示向量 $\boldsymbol{x}$ 的维度. 则在本文的模型中，显然有 $\boldsymbol{x}=(\boldsymbol{U}_u,\boldsymbol{V}_v)$，$y=T_{uv}$，于是得到：

$$\hat{T}_{uv}=z+\sum_{i=1}^{a} w_i U_{ui}+\sum_{j=1}^{b} w_{j+a}V_{vj}+\sum_{i=1}^{a-1}\sum_{j=i+1}^{a} W_{ij}U_{ui}U_{uj}+\sum_{i=1}^{b-1}\sum_{j=i+1}^{b} W_{i+a,j+a}V_{vi}V_{vj}+\sum_{i=1}^{a}\sum_{j=1}^{b} W_{i,j+a}U_{ui}V_{vj}, \tag{2}$$

我们令：

$$p_u=\sum_{i=1}^{a} w_i U_{ui}+\sum_{i=1}^{a-1}\sum_{j=i+1}^{a} W_{ij}U_{ui}U_{uj}, \tag{3}$$

$$q_v=\sum_{j=1}^{b} w_{j+a}V_{vj}+\sum_{i=1}^{b-1}\sum_{j=i+1}^{b} W_{i+a,j+a}V_{vi}V_{vj}, \tag{4}$$

可以观察到，p_u 只和用户 u 有关，q_v 只和物品 v 有关，于是我们得到最终的二次回归模型：

$$\hat{T}_{uv}=z+p_u+q_v+\sum_{i=1}^{a}\sum_{j=1}^{b} W_{ij}U_{ui}V_{vj}, \tag{5}$$

注意到此处我们重新定义 $\boldsymbol{W}\in \mathrm{R}^{a\times b}$. 式(5)表示用户 u 对物品 v 的评分可近似由 1 个全局常数 z、1 个与用户有关的常数 p_u、1 个与物品有关的常数 q_v、以及他们的特征元素之间的两两乘积的加权和组成. 所以，我们可以令 z 为整个数据集评分的均值，令 p_u 为用户 u 的全部评分的均值，令 q_v 为物品 v 被评分的均值，即：

$$z=\Big(\sum_{u,v}\delta_{uv}T_{uv}\Big)\Big/\Big(\sum_{u,v}\delta_{uv}\Big), \tag{6}$$

$$p_u=\Big(\sum_{v}\delta_{uv}T_{uv}\Big)\Big/\Big(\sum_{v}\delta_{uv}\Big), \tag{7}$$

$$q_v=\Big(\sum_{u}\delta_{uv}T_{uv}\Big)\Big/\Big(\sum_{u}\delta_{uv}\Big), \tag{8}$$

其中，当 $T_{uv}=0$ 时表示用户 u 对物品 v 的评分不存在，此时 $\delta_{uv}=0$，否则 $\delta_{uv}=1$.

根据式(5)我们可以得到模型的拟合误差：

$$L=\frac{1}{2}\sum_{u,v}\delta_{uv}(T_{uv}-\hat{T}_{uv})^2=\frac{1}{2}\sum_{u,v}\delta_{uv}\Big(T_{uv}-z-p_u-q_v-\sum_{i=1}^{a}\sum_{j=1}^{b} W_{ij}U_{ui}V_{vj}\Big)^2, \tag{9}$$

优化式(9)使得 L 最小即可得到我们所需的特征信息矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{V}$. 我们利用随机梯度下降法求解，并设学习率为 η，我们得到参数更新公式：

$$W_{ij}=W_{ij}-\eta\sum_{u,v}[\delta_{uv}(\hat{T}_{uv}-T_{uv})U_{ui}V_{vj}], \tag{10}$$

$$U_{ui}=U_{ui}-\eta\sum_{v}\Big[\delta_{uv}(\hat{T}_{uv}-T_{uv})\sum_{j=1}^{b} W_{ij}V_{vj}\Big], \tag{11}$$

$$V_{vj}=V_{vj}-\eta\sum_{u}\Big[\delta_{uv}(\hat{T}_{uv}-T_{uv})\sum_{i=1}^{a} W_{ij}U_{ui}\Big]. \tag{12}$$

下面给出本文提取特征的算法的伪代码，如算法 1 所示：

算法 1. 特征提取算法.

输入：评分矩阵 $\boldsymbol{T}$、参数 n,m,a,b,η；

输出：矩阵 $\boldsymbol{U},\boldsymbol{V}$.

① 随机初始化 $\boldsymbol{W},\boldsymbol{U},\boldsymbol{V}$；
② 根据式(6)计算 z；
③ for $u=1$ to n do
④　根据式(7)计算 p_u；
⑤ end for
⑥ for $v=1$ to m do
⑦　根据式(8)计算 q_v；
⑧ end for

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

⑨ repeat

⑩ 根据式(5)更新矩阵$\hat{T}$;

⑪ 根据式(10)更新矩阵W;

⑫ 根据式(11)更新矩阵U;

⑬ 根据式(12)更新矩阵V;

⑭ until convergence.

在步骤⑫中,为防止更新U的值所需要的W是本轮更新后的,可以利用临时变量记住上一轮更新后的W,同理步骤⑬更新V也可以采用同样的策略.

最终得到U,V矩阵即可表示为用户和物品的特征信息矩阵,然后利用表1中的相似度算法就能够得到用户以及物品的特征信息相似度.

2.3 图模型的构建

在本节我们将介绍图1中的图模型模块,将2.2节计算得到的相似度与用户的评分信息结合构建出一个无向图,利用它来计算最终的聚类结果.

在我们的图模型中,所有的顶点由用户和物品组成,边则分别由3部分构成:用户相似度、物品相似度以及用户对物品的评分.另外,为了强调高评分在图中的重要性,有变换函数:

$$f(T_{uv})=(1+\mathrm{e}^{-\tau(T_{uv}-T_{\mathrm{med}})})^{-1}, \tag{13}$$

其中,T_{med}表示评分制的中值,τ为超参数.该函数的作用是将中值映射为权重值0.5,当评分高于中值时,表示用户对该物品有喜欢的倾向,因此在聚类中应该有较高权重,但同时弱化高评分之间的差距,比如4分与5分的权重差距显然应该小于4分与3分之间的差距,同理,评分小于中值时应该具有较低的权重,且同样需要弱化低评分之间的权重差距.

因此,该图模型的构建过程如下:

1) n个用户与m个物品组成$n+m$个节点,用户节点编号为$1,2,\cdots,n$,物品节点表示为编号$n+1,n+2,\cdots,n+m$,第v个物品编号为$n+v$;

2) 用户u与物品v之间边的权值定义为$W_{u(n+v)}=f(T_{uv})$;

3) 用户i与用户j之间边的权值定义为$W_{ij}=\alpha\times sim(\boldsymbol{U}_i,\boldsymbol{U}_j)$,其中$\alpha$表示放缩因子,其目的是为了均衡用户评分所转化的权值与根据相似度得到的权值之间的差异,函数$sim(\boldsymbol{U}_i,\boldsymbol{U}_j)$表示用户$i$与用户$j$之间的特征信息相似度,可以采用表1中的相似度算法计算,本文采用的是余弦相似度算法;

4) 物品i与物品j之间边的权值定义为$W_{(n+i)(n+j)}=\beta\times sim(\boldsymbol{V}_i,\boldsymbol{V}_j)$,函数$sim(\boldsymbol{V}_i,\boldsymbol{V}_j)$表示物品$i$与物品$j$之间的特征信息相似度,参数$\beta$也是放缩因子,为了平衡物品相似度与评分权值之间的差异.

通过上述步骤构建好图模型后,便得到该无向图的权值矩阵$\boldsymbol{W}\in\mathrm{R}^{(n+m)\times(n+m)}$,下面讨论如何利用它得到最终的聚类结果.

2.4 聚类结果的计算

假设最终的聚类分组个数为c,我们采用一种评价制度,每个用户和物品都对这c个分组进行评价,然后我们根据评价情况将用户和物品分配到它们倾向的分组中去.为了找到上述问题的解,定义评价矩阵$\boldsymbol{P}=\begin{pmatrix}\boldsymbol{Q}\\ \boldsymbol{R}\end{pmatrix}\in\mathrm{R}^{(n+m)\times c}$,其中$\boldsymbol{Q}\in\mathrm{R}^{n\times c}$,$\boldsymbol{R}\in\mathrm{R}^{m\times c}$.矩阵$\boldsymbol{Q}$中的元素$Q_{ij}$表示用户$i$对第$j$组的评价,正值表示积极的评价,用户可能对该组的物品感兴趣;否则为负值则用户对该组的物品持有消极评价,对该组物品不感兴趣或厌恶.同理,矩阵$\boldsymbol{R}$中的元素R_{ij}表示物品i对第j组的评价.

显然,在2.3节的图模型中,权值越大的边对应的2个顶点,表明用户对该物品更加关注或者用户之间、物品之间更加相似,它们应该更有可能被分在同一组中,即它们在矩阵$\boldsymbol{P}$中对应的2行行向量近似.有损失函数:

$$\varepsilon(\boldsymbol{P})=\sum_{i,j}W_{ij}\|\boldsymbol{P}_i-\boldsymbol{P}_j\|^2. \tag{14}$$

本文利用Belkin等人[32]的方法来求解式(14),首先我们定义对角型矩阵$\boldsymbol{D}\in\mathrm{R}^{(n+m)\times(n+m)}$,其元素$D_{ii}$为矩阵$\boldsymbol{W}$的第$i$行之和,即:

$$D_{ii}=\sum_{j=1}^{n+m}W_{ij}, \tag{15}$$

则将式(14)展开并经过一定的线性变换,得到:

$$\varepsilon(\boldsymbol{P})=2\mathrm{tr}(\boldsymbol{P}^{\mathrm{T}}(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{P}). \tag{16}$$

因此,要使得$\varepsilon(\boldsymbol{P})$最小,则只需要找到一个矩阵$\boldsymbol{P}$,使其满足$\boldsymbol{P}^{\mathrm{T}}(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{P}$的迹最小.为了使得最优化问题式(14)有解并且使得解空间的维度不低于c,我们对矩阵$\boldsymbol{P}$进行限制:

$$\boldsymbol{P}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{P}=\boldsymbol{I}. \tag{17}$$

经过上述讨论,聚类结果的评价矩阵$\boldsymbol{P}$的求解转化为求解满足式(17)的矩阵$\boldsymbol{P}$,使得$\boldsymbol{P}^{\mathrm{T}}(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{P}$的迹最小.根据文献[32],这样的矩阵$\boldsymbol{P}$应当是由矩阵$\boldsymbol{D}-\boldsymbol{W}$的前$c$个非0最小特征根对应的特征向量组成的.则可以根据$(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{p}=\lambda\boldsymbol{D}\boldsymbol{p}$求得矩阵$\boldsymbol{D}-\boldsymbol{W}$的广义特征根,其中$\boldsymbol{p}$为广义特征根$\lambda$对应的广义特征向量.最后,将前$c$个非0最小特征根对应的特征向量组成的矩阵$\boldsymbol{P}$即为所求.

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

得到用户和物品的评价分布矩阵后，再根据评价值的大小确定某个用户或者物品所属的分组. 在本文的算法中，只要用户或者物品对某个分组持有正的评价值，则将该用户或者物品分配到该分组中去，这样便得到了最终的聚类结果.

2.5　推荐结果的生成

如图 2 所示，图 2(a)表示非聚类的协同过滤算法生成推荐结果的流程，它直接将原始的用户评分矩阵通过协同过滤算法进行计算，得到用户对其他物品的预测评分从而进行推荐；而在基于聚类的协同过滤算法中，如图 2(b)所示，首先利用聚类算法对原始的用户评分矩阵进行聚类，得到若干聚类分组，然后对每个聚类分组分别使用传统推荐方法得到在这些分组上的推荐结果，最后，将所有分组得到的推荐结果进行汇总即得到最终的推荐结果. 在将各个分组上的推荐结果汇总的过程中，由于聚类算法允许 1 个用户或者物品属于多个分组，因此可能会存在 1 个用户与 1 个物品同时属于 2 个不同分组但是预测得分不同，本文的处理方式为选择用户评价得分较高分组上的预测得分作为最终结果.

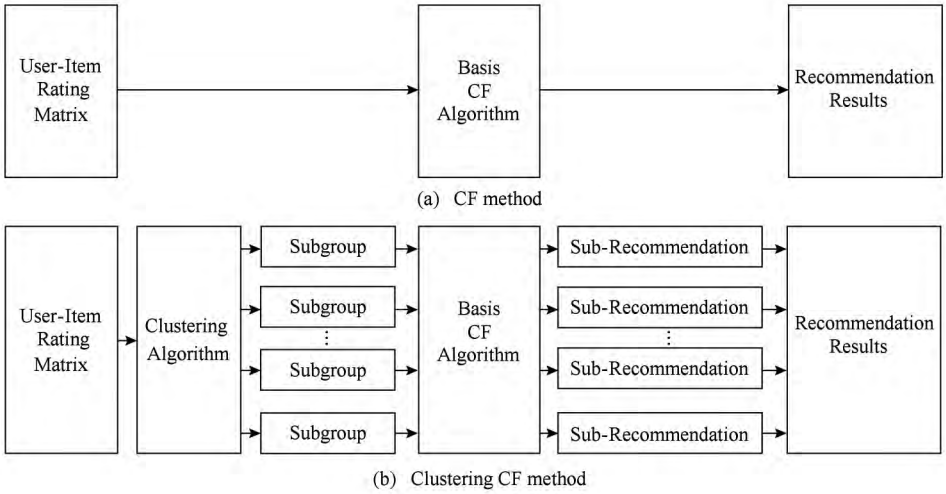

Fig. 2　The framework of the CF and clustering CF method

图 2　协同过滤算法与聚类的协同过滤算法的推荐框架

3　实验评估

3.1　数据集

本文的实验将使用 3 个公开数据集来检验聚类模型的性能：

1) MovieLens-100K

由于大部分推荐系统规模较小，这类系统的数据表现为数据少、用户平均打分数量中等，因此本文选用了 MovieLens-100K 数据集，它来自于 MovieLens 网站上的用户对电影的打分数据，包含 943 个用户对 1 682 部电影作品的近 100 000 条评分记录，用户的打分均为不大于 5 的正整数.

2) MovieLens-1M

部分推荐系统较为成熟，积累了一定的用户，但是物品品种类较为固定，因此此类推荐系统表现为用户量较多、商品数量较少、平均用户打分次数多. 针对该系统的特点，本文选用了 MovieLens-1M 数据集，它与 MovieLens-100K 数据集来源相同，不同的是它更晚发布且拥有了更多的用户与更多的评分记录，它包含了 6 040 个用户对 3 952 部电影的 1 000 209 条评分数据，且每个用户都至少对 20 部电影进行了打分.

3) Epinions

某些推荐系统表现为用户量与商品量都是庞大的，但是却只有很少的评分记录，很难从中分析用户喜好，因此，本文选用了 Epinions 数据集，该数据集包含了 22 166 个用户对 296 277 个商品的打分信息，显然，它比前 2 个数据集要大得多，但是，该数据集有很多用户打分次数较少以及很多物品缺乏有价值的评分信息，因此，本文对该数据集进行了预处理，将打分次数低于 10 次的用户从数据集中移除，同样地，将被打分次数少于 10 次的商品也移除，最后，得到了 21 109 个用户对 13 957 件商品的 433 507 条评分数据.

这 3 个数据集的统计信息如表 2 所示. 其中，$Sparsity$ 表示评分矩阵中空元素(未评分)所占的比例，即：

$$Sparsity=\left(1-\frac{Ratings}{Users\times Items}\right)\times 100\%.$$

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

Table 2 The Statistics of the Three Datasets

表 2 3 个数据集的统计信息

Dataset	*Users*	*Items*	*Ratings*	*Ratings* per User	*Ratings* per Item	*Sparsity*/%
MovieLens-100K	943	1 682	100 000	106.4	59.45	93.70
MovieLens-1M	6 040	3 952	1 000 209	165.6	253.09	95.81
Epinions	21109	13 957	433 507	20.54	31.06	99.85

从表 2 可以看出,MovieLens-100K 数据集规模最小;MovieLens-1M 数据集平均用户打分次数最高,意味着信息更加丰富;而 Epinions 数据集最为稀疏,有着庞大的用户与物品量却拥有着较低的打分数量.

3.2 评估指标

在实验性能的评估上,本文分别从预测性能与推荐性能 2 个方面进行测试.

1) 在预测性能上. 本文采用推荐系统领域内比较常用的平均绝对误差(mean absolute error, *MAE*)指标,其表达式为

$$MAE = \frac{1}{N}\sum_{i,j} | T_{ij} - \hat{T}_{ij} |,$$

其中,N 表示测试数据的个数,T_{ij} 表示测试数据中用户 i 对物品 j 的真实评分,$\hat{T}_{ij}$ 表示利用推荐算法得到的预测评分. 显然,*MAE* 值越低,表示预测的结果与用户的真实情况更接近,算法性能越好.

2) 在推荐性能上. 本文采用 Top$K$ 推荐方法,即从预测结果中选出预测评分最高的 K 个物品并排序,然后比较选出的这 K 个物品与测试集中用户最喜爱的 K 个物品的吻合程度. 采用的评估指标为归一化折损累积增益(normalized discounted cumulative gain, *NDCG*),对用户 u,其对应 *NDCG* 计算为

$$NDCG@K = \frac{1}{IDCG}\sum_{i=1}^{\min(K,N_u)} \frac{2^{r_i}-1}{\mathrm{lb}(i+1)},$$

其中,N_u 表示测试集中用户 u 的评分数量,当模型推荐结果的第 i 个物品出现在测试集中用户的 Top$K$ 列表中时,$r_i = 1$,否则 $r_i = 0$. *IDCG*(ideal discounted cumulative gain)为常量,表示完美推荐时的累积增益,其表达式为

$$IDCG@K = \sum_{i=1}^{\min(K,N_u)} \frac{1}{\mathrm{lb}(i+1)},$$

对所有用户的 *NDCG* 求平均值,即为最终的评估指标,显然,*NDCG* 值越大,表示模型性能越好.

3.3 评估方法与对比算法

如图 2(b)所示,利用本文算法进行推荐的过程是先将用户以及物品进行聚类,再在每个不同的聚类小组上调用其他协同过滤推荐算法得到推荐结果. 因此,本文对比方法为采用不同的聚类算法进行分组,然后在各种算法得到的聚类分组上调用同一种协同过滤推荐算法来对比最终的预测性能和推荐性能.

在协同过滤推荐算法的选择上,本实验选择了当下较为主流的推荐系统所采用的算法,包括 2 个基于内存的协同过滤算法以及 4 个著名的基于模型的协同过滤算法:

1) Item-Based(IB)[7]. 其基本思想是预先根据所有用户的历史偏好数据计算物品之间的相似性,然后把与用户喜欢的物品相类似的物品推荐给用户.

2) User-Based(UB)[6]. 该算法与 Item-Based 算法类似,不同之处在于该算法计算的是用户之间的相似度.

3) SVD[14]. 该算法是基于 SVD 矩阵分解算法,将评分矩阵分解为多个矩阵的乘积,最后将利用这些矩阵的乘积预测用户的评分.

4) NMF[15]. 和 SVD 方法类似,不同的是它将原始评分矩阵分解为 2 个矩阵的乘积.

5) MMMF[16]. 该算法也是一种矩阵分解模型的算法,由于其具有较好的推荐性能而被广泛采用.

6) NPCA[17]. 该算法在文献[17]中被提出,它拥有着更高的预测准确度.

在聚类算法的选择上,本文采用 4 种算法来与本文方法进行对比:

1) UserClu(UC). 它是单侧聚类算法,该算法基本思想是利用 k-means 算法对数据集进行基于用户的聚类,将用户分成多个组.

2) ItemClu(IC). 该算法与 UserClu 算法类似,不同之处在于该算法是利用 k-means 算法对物品进行聚类.

3) CoClust(CoC)[29]. 它是双侧协同聚类算法,其思想是将用户与物品进行联合聚类,但是 1 个用户或者 1 个物品只能属于 1 个分组.

4) MCoC[30]. 它是双侧多类协同聚类算法,它和 CoClust 算法思想一致,不同之处在于该算法允许 1 个物品或者用户属于多个分组,这一点与本文

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

的算法一致，这也更符合现实情况：1 个用户可能对多个领域感兴趣.

3.4　参数设置

在进行实验之前，需要对本文模型有关参数进行设置.

1）在特征提取阶段. 用户特征维度 a 和物品特征维度 b 与具体的数据集有关，在 MovieLens-100K 数据集中，我们设置 $a=16$，$b=18$；在 MovieLens-1M 数据集中，我们设置 $a=20$，$b=18$；在 Epinions 数据集中，我们设置 $a=24$，$b=22$；在学习率的设置上，在 3 个数据集中均为 $\eta=0.01$.

2）在图模型构建阶段. 参数 $\tau=1.5$，由于 3 个数据集的评分制度都是 1～5 分的整数，因此有评分制中值 $T_{\mathrm{med}}=3$. 参数 α 与 β 是为了均衡评分权重与相似度权重之间的差异，而且，不同数据集相似度取值情况也有所不同，在 MovieLens-100K 数据集中 $\alpha=1.3$，$\beta=1.3$；在 MovieLens-1M 数据集中 $\alpha=1.1$，$\beta=1.2$；在 Epinions 数据集中 $\alpha=1.9$，$\beta=1.5$.

3）在聚类计算阶段. 聚类分组个数 c 是实验中的控制因素，我们主要考查在不同分组下本文模型与其他算法性能优劣.

4）在性能评估阶段. 由于大部分推荐系统都采用的 Top10 推荐，因此，评测指标 $NDCG$ 的 K 值在 3 个数据集上均取 $K=10$.

关于上述参数的选择原因及其对模型性能的影响，本文将在 3.7 节进行讨论.

3.5　实验结果

在本文的实验中，我们将 3 个公开数据集都分成 2 部分，随机选取 80% 的数据作为训练数据，剩下的 20% 作为测试数据，用来检测算法的预测和推荐性能. 测试方法为先固定分组的个数，然后利用上述 4 种聚类算法对训练数据集进行分组，然后对所有算法得到的分组结果中选用同一种协同过滤推荐算法进行预测或推荐，统计各个算法的 MAE 值与 $NDCG$ 值. 由于矩阵分解模型的结果与分解的维度有关，因此，实验中分别对 SVD，NMF，MMMF，NPCA 算法的分解维度进行了不同的设置并进行实验，在所有不同纬度下取得的性能中，记录最好的

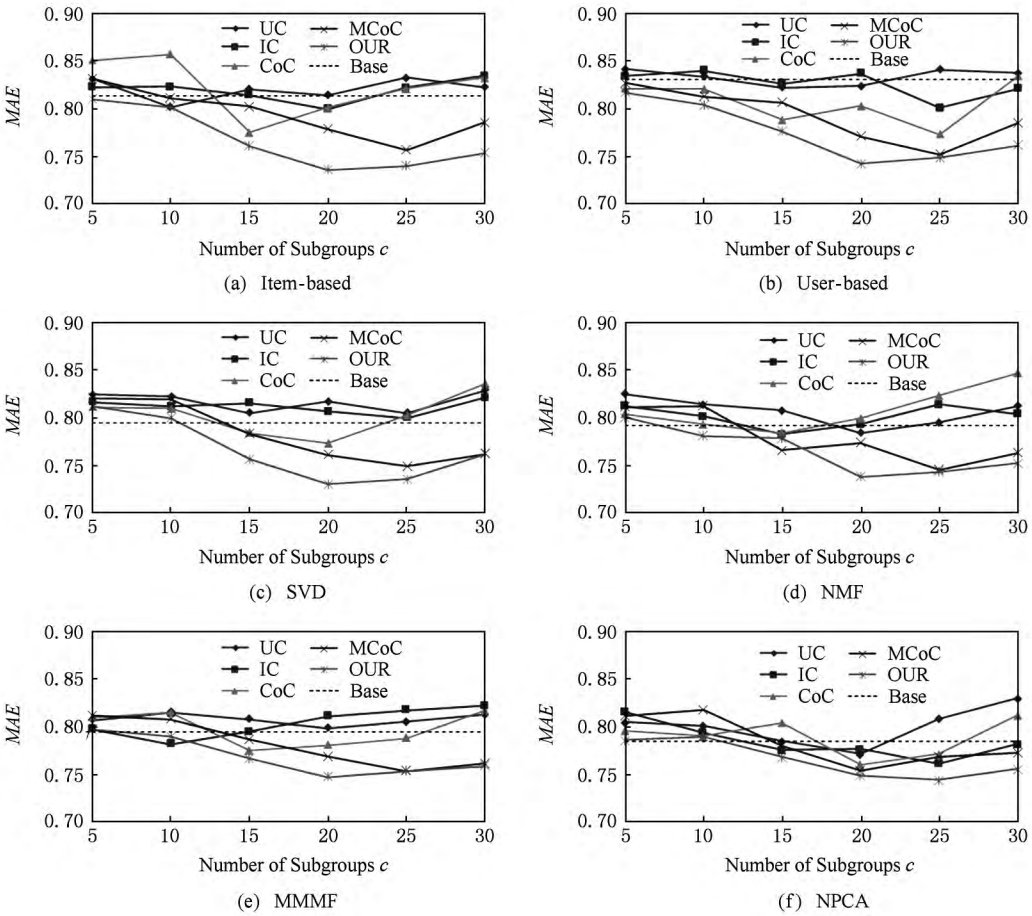

Fig. 3　The evaluation result of the prediction performance on MovieLens-100K

图 3　在 MovieLens-100K 数据集上的预测性能测试结果

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

一个来作为与本文方法的对比结果.

1) 在数据集 MovieLens-100K 上的测试

由于 MovieLens-100K 数据集规模最小,因此分组不易过多,否则由于数据过度分散导致信息丢失严重.在参数设置方面,本文通过多个分组规模测试,选取了最优效果较为集中的分组区间(5~30 组).

预测性能的测试结果如图 3 所示,6 个子图分别表示了 6 种不同协同过滤算法下的测试结果,在每个子图中,我们对比了包括本文算法在内的 5 种聚类算法在不同分组个数情况下的预测性能,横轴表示分组个数,纵轴表示测试指标 *MAE* 值,每个子图中的水平虚线(基准线)表示了在不分组的情况下直接使用协同过滤算法得到的 *MAE* 值,也即使用图 2(a)的方法得到的结果.

从图 3 中可以看出,当分组较少时,各不同聚类算法差别不太明显,但是随着分组个数的增多,本文算法的优势越来越突出,分组数为 20 左右时,除了 NPCA 算法以外,在其他算法上都表现出了明显的性能优势.验证了本文的算法能够很好地应对小规模的推荐系统,提升它们的预测性能.

在 Top10 推荐性能测试中,我们的实验结果如图 4 所示:

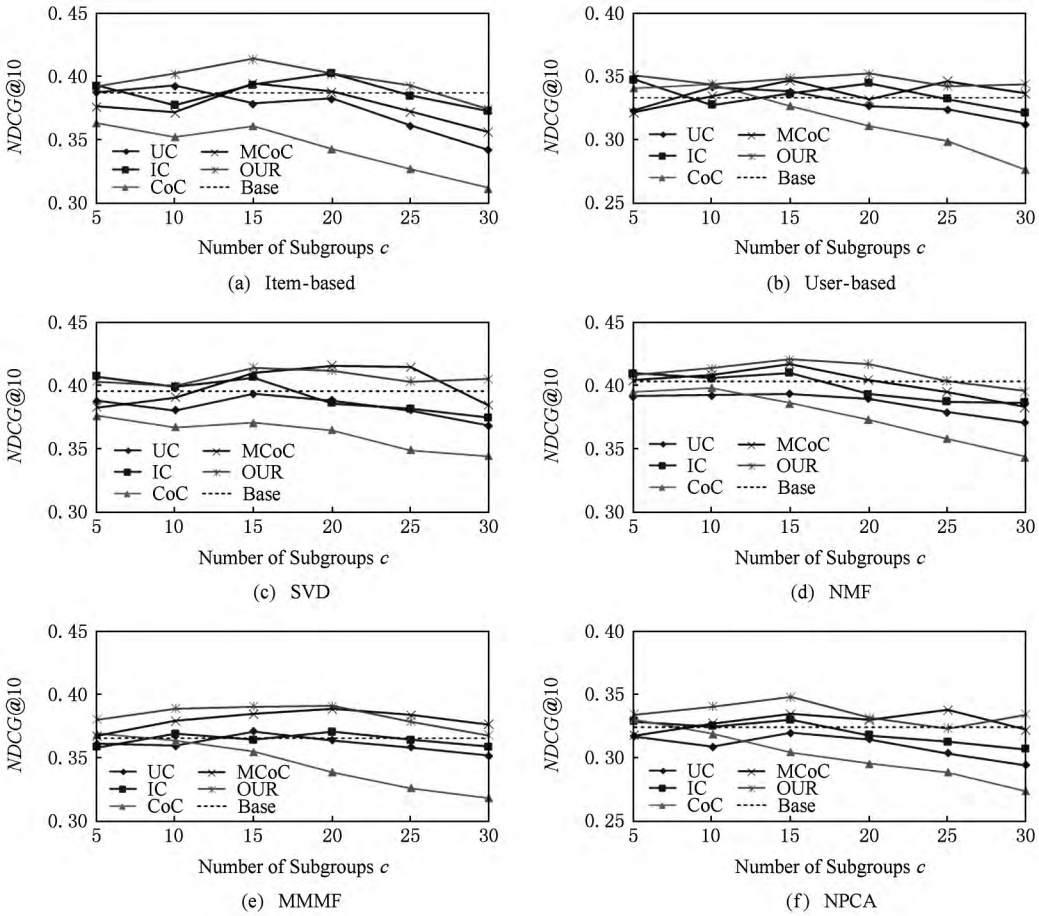

Fig. 4 The evaluation result of the Top10 recommendation performance on MovieLens-100K

图 4 在 MovieLens-100K 数据集上的 Top10 推荐性能测试结果

从实验结果可以看出,除了在图 3(c)中推荐性能比 MCoC 稍差以外,本文聚类模型在其他算法的测试上均取得了较好的推荐性能.另外,对比图 3 与图 4 我们注意到,采用 Item-based 推荐算法虽然预测性能不高,但是推荐性能却比较好,相反采用 NPCA 推荐算法虽然取得了最高的预测性能,但是推荐效果却很差,经分析,我们认为出现这种"反常"现象的原因除了算法本身特点外,还包括数据集特点与推荐规则等原因.首先,分析表 2 可知 MovieLens-100K 数据集平均用户打分数量在 106 左右,则测试集中打分数约为 20,这些评分中为 5 分的显然就更少了(不足 10 个),因此,用户 Top10 列表中会包含很多低分物品,而我们的推荐规则只推荐高分物品,这导致用户列表中多数物品无法被推荐到,因此推荐性能取决于对测试集中仅有的若干高分物品的预测准确度,尽管 NPCA 算法 *MAE* 值较低,但是有可

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

能其对高分物品的评估存在一定误差，从而导致预测性能好反而推荐效果不理想，另一方面，Item-based 推荐效果好的原因，则是由于基于物品相似度的推荐往往符合用户的心理，相似的物品往往更容易引起用户注意，尽管 Item-based 方法会将原本低评分却与高评分物品相似的物品预测为高分并推荐（预测性能较差），但该物品可能正好被用户因为与其喜欢的物品相似而购买或者关注（尽管用户购买后不喜欢），因而在测试集高分物品不足的情况下，Item-based 推荐算法可能会起到不错的效果.

尽管 *NDCG* 指标会出现上述"反常"现象，但仍然被众多推荐系统作为重要指标之一，因为它只关注算法对高分物品的预测准确度，而不必关心对低分物品是否也有准确的预测，这正是"推荐"所需要的.

2）在数据集 MovieLens-1M 上的测试

MovieLens-1M 数据集与 MovieLens-100K 数据源一致，区别在于前者数据规模更大，并且平均每个用户打分数与平均每个物品被打分次数在 3 个数据集里都是最高的. 该测试用来检验本文算法在数据信息相对丰富的情况下的性能. 我们在该数据集上做了同 MovieLens-100K 一致的实验，预测性能实验结果与 Top10 推荐性能实验结果分别如图 5 和图 6 所示.

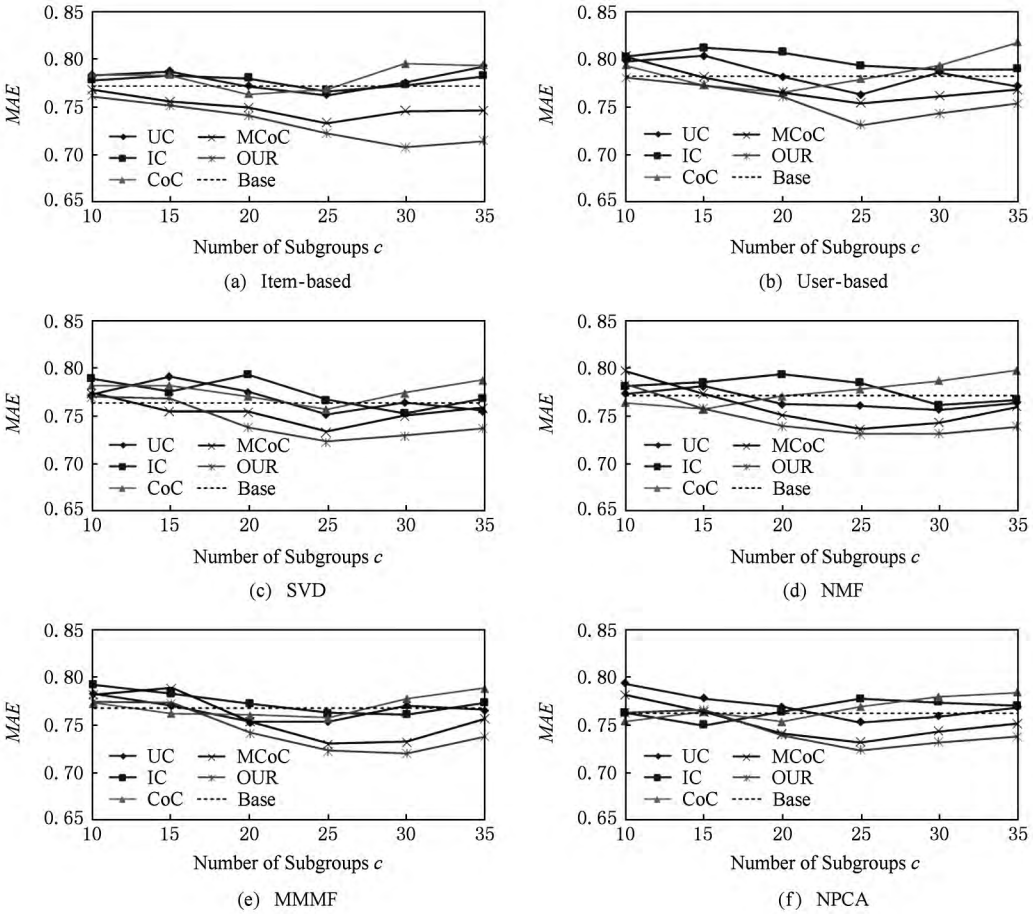

Fig. 5　The evaluation result of the prediction performance on MovieLens-1M

图 5　在 MovieLens-1M 数据集上的预测性能测试结果

从图 5 实验结果可以看出，在数据集打分数量足够多的情况下，各个算法都表现出一定的稳定性，说明足够多的评分数能够帮助各聚类算法更加准确地捕捉用户和物品的特征，降低不同算法之间的差异. 尽管如此，从实验结果来看本文算法所带来的预测性能提升与其他算法相比仍然是有优势的. 另一方面，通过观察图 6 的推荐结果也证实了上述分析：与图 4 相比，图 6 中各算法的推荐性能差异也变得较小，波动幅度在 0.03 左右. 但是本文模型以及 MCoC 与其他聚类模型相比仍然取得了微弱优势，另外，图 6 的实验结果也没有出现"反常"现象，正是由于测试集评分足够多，在用户的 Top10 列表中几乎都是高分物品，因此，预测性能越好一定程度上可以反映出推荐性能较好.

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

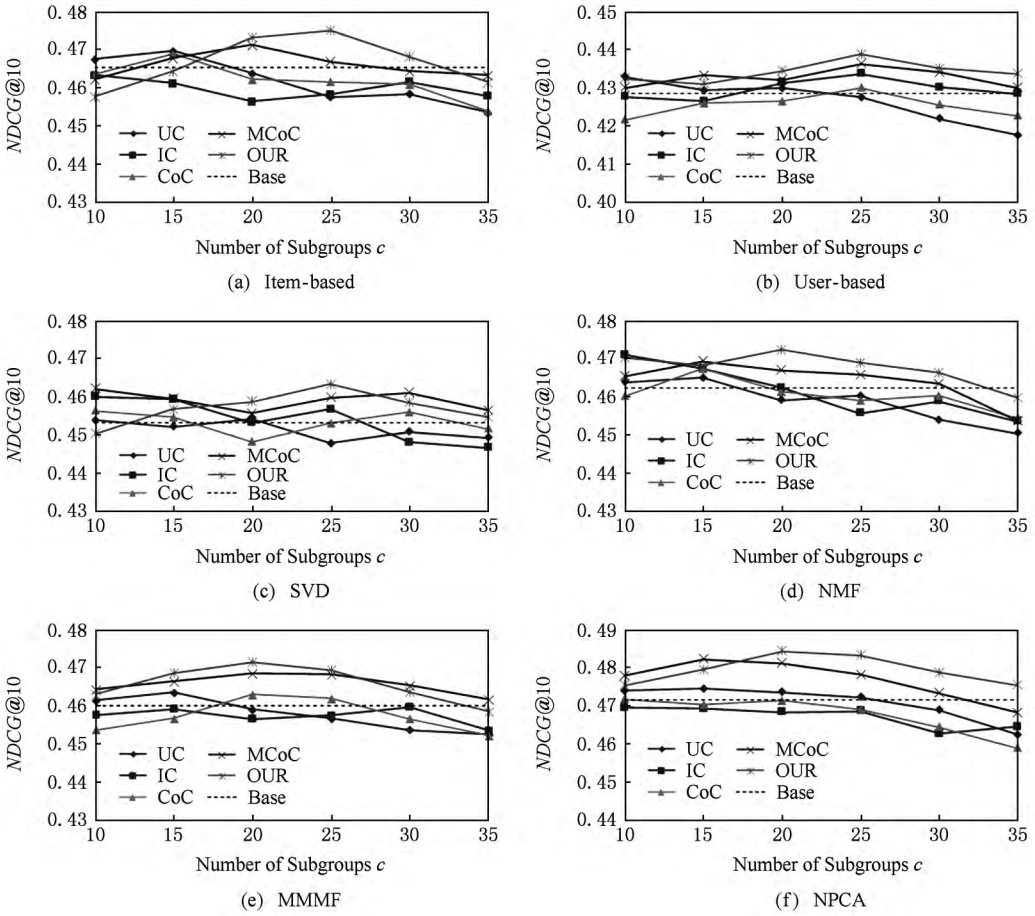

Fig. 6 The evaluation result of the Top10 recommendation performance on MovieLens-1M

图 6 在 MovieLens-1M 数据集上的 Top10 推荐性能测试结果

3）在数据集 Epinions 上的测试

在所有 3 个数据集中，Epinions 数据集规模最为庞大，该数据集稀疏率也是最高的，我们利用该数据集检验本文算法在应对大规模数据时的性能，实验结果如图 7 和图 8 所示.

从图 7 的预测结果可以看出，大部分算法的预测结果 *MAE* 值都在 0.85 以上，和在 MovieLens-1M 数据集上的 0.75 相比性能大打折扣，说明该数据集大的稀疏性使得各个算法都难以捕捉用户物品的特征，预测准确度降低，即使增大分组个数，能够发掘的有效信息也是有限的，无法大幅度提升准确度，但是本文的算法依然能够发掘比其他算法更多的信息，提升了更大的预测性能，在所有同类算法中表现优异. 验证了本文算法也能够应对具有高稀疏性数据的推荐系统.

但是另一方面，观察图 8 可以发现，各个算法的 *NDCG* 指标都在 0.20～0.25 之间，随着分组个数增大多数算法的推荐性能呈现无规律波动，且波动幅度达到 0.1 左右，通过分析表 2 我们认为，造成这种情况的原因是由于该数据集用户评分太少，只有平均 20 个左右，因此测试集中用户评分的物品数往往只有一两个甚至没有，这使得所有算法都难以正确推荐出测试集中的物品，即使增大分组个数也不一定能够提升推荐性能.

3.6 实验结果的总结和分析

在 3.1 节所述的 3 个不同数据集上的实验中，从实验结果图 3、图 5、图 7 对比各个算法的预测性能可以得到 6 个结论：

1）图 3、图 5 以及图 7 中大部分实验结果都位于基准线下方，这表示通过聚类得到的推荐结果要比无聚类的要好，说明通过聚类的方式能够提升协同过滤算法的预测准确度.

2）基于特征的协同聚类算法在所有 3 个数据集上都有杰出的表现，证明了该算法在预测功能上的有效性.

3）在 3 个数据集上本文算法与 MCoC 同其他聚类算法如 UC 与 IC 等相比，有更好地表现. 原因是数据集的高稀疏性使得仅仅使用 *k*-means 进行聚

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

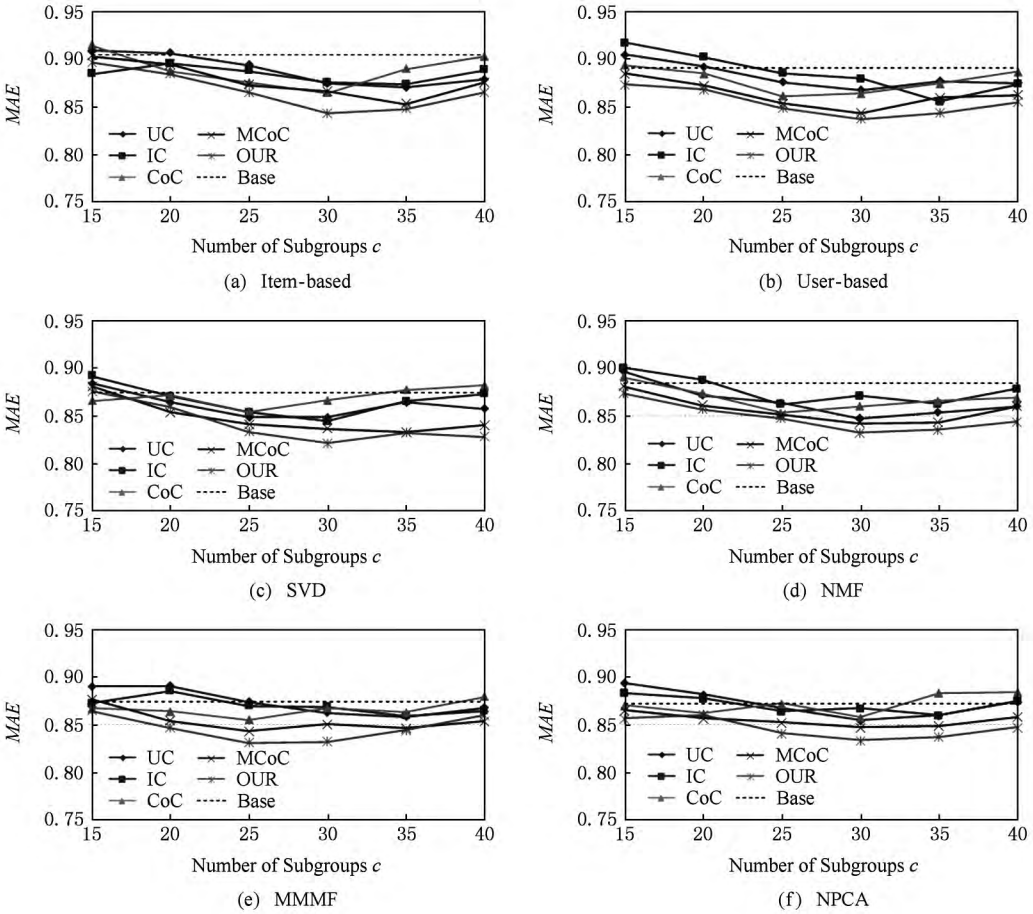

Fig. 7　The evaluation result of the prediction performance on Epinions

图 7　在 Epinions 数据集上的预测性能测试结果

类不能够得到理想的结果，并且双侧聚类算法能够获得比单侧聚类更多的信息.

4）虽然 MCoC 算法也能够提升不错的预测性能，但是从实验结果上来看，本文的算法总是比 MCoC 更好一些，说明加入用户之间以及物品之间的相似性联系比单纯的依赖用户对物品的打分进行聚类能够得到更好的结果，能够利用更多的数据本身带来的信息.

5）当分组数较少时，所有聚类算法的性能差别不大，有的聚类后的性能要比不聚类的协同过滤算法还要差. 这说明较少的分组不能够完全区分用户以及物品的特点，分类不够准确，当分组个数适当时，大部分聚类后的预测性能都有明显提升.

6）分组不易过多，所有的实验中当分组过大时，在折线图中都表现出了上升的趋势，意味着继续增大分组个数可能会得到更差的结果. 原因可能是分组数太多，导致数据过度分散，丢失了过多的数据之间的联系，因此导致性能下降.

另一方面，观察实验结果图 4、图 6 以及图 8 对比各个算法的推荐性能，我们有得到 3 个结论：

1）在 MovieLens-100K 数据集上，*NDCG* 指标值大约在 0.3～0.45 之间；在 MovieLens-1M 上则提升至 0.41～0.48 之间；而在 Epinions 数据集上，则只有 0.2～0.25. 在通过观察表 2 的数据集统计信息后，我们认为数据集中用户的打分数量一定程度上影响各个算法的推荐性能，打分数量越少，TopK 推荐时测试集包含的低分物品就会越多，而这些物品一般不会被推荐到，因此推荐性能就会变差，显然，评分数量越多，越有可能获得准确的推荐.

2）尽管数据集的特征对模型推荐性能产生一定的影响，但是在 3 个数据集上本文模型都取得了不错的推荐效果，经本节分析，推荐性能很大程度上取决于对测试集中高分物品的预测准确度，而本文模型在聚类过程中，评分越高越有可能出现在一个聚类结果中，因此，本文模型对于高分物品的预测准确度有一定的保障，从实验结果也可以看出，正是由于该原因，MCoC 算法也保持着高预测准确度以及高推荐准确度.

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

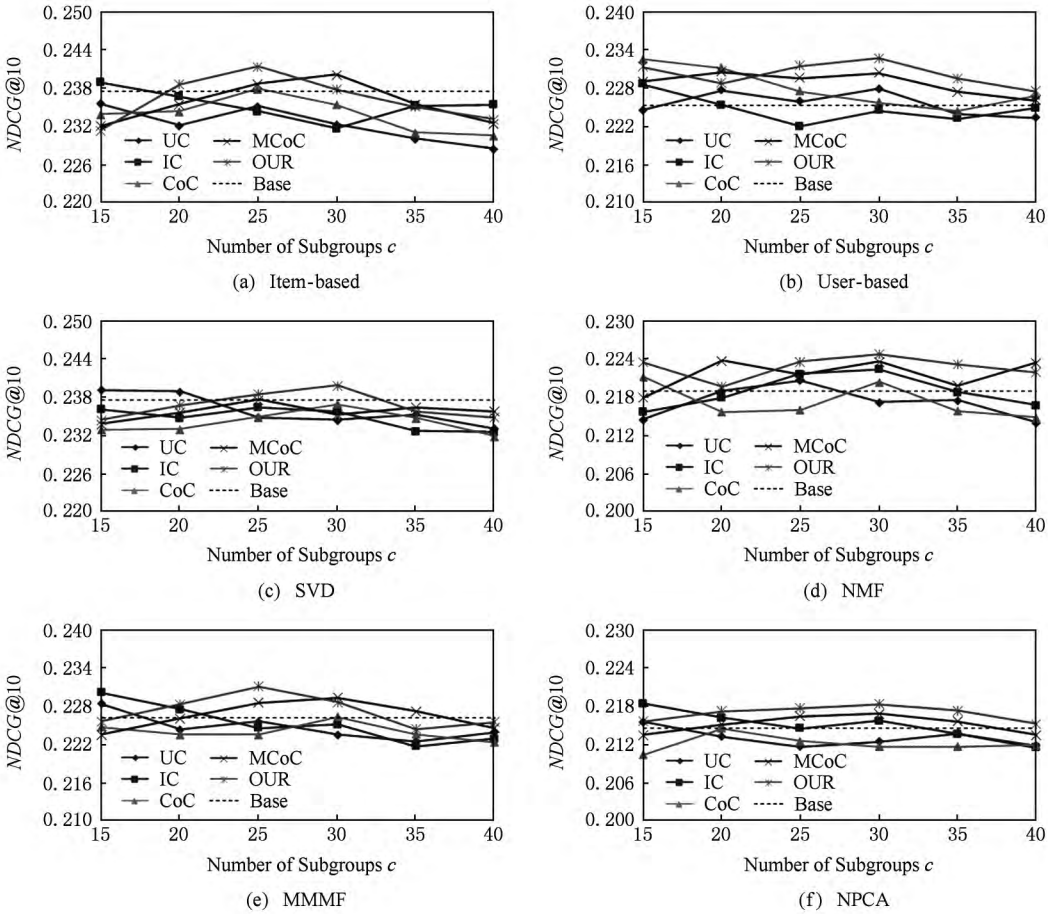

Fig. 8 The evaluation result of the Top10 recommendation performance on Epinions

图 8 在 Epinions 数据集上的 Top10 推荐性能测试结果

3) 在所有 3 个数据集上,本文模型取得最佳推荐性能时($NDCG$ 值最大)在聚类结果上所用的推荐算法均为 Item-based,说明在推荐性能上,Item-based 算法最简单也最为有效,因为它总是推荐和用户兴趣相似的物品,但是由于该原因,导致 Item-based 算法的推荐结果缺乏多样性[29].

通过以上对预测性能与推荐性能的分析,我们可以认为,本文模型能够有效处理不同特点的推荐系统,提升它们的预测性能与推荐性能.

3.7 参数的影响

在本节,我们将讨论本文实验所采用 3.4 节参数设定的原因,主要讨论参数 a,b,τ,α,β 对模型性能的影响,以下涉及实验的部分均采用 MAE 指标.

首先分析参数 τ,它是式(13)中的参数,该函数将评分映射成图模型中边的权重,我们分别选取不同的 τ 值来观察它将评分映射为权重的情况,如图 9

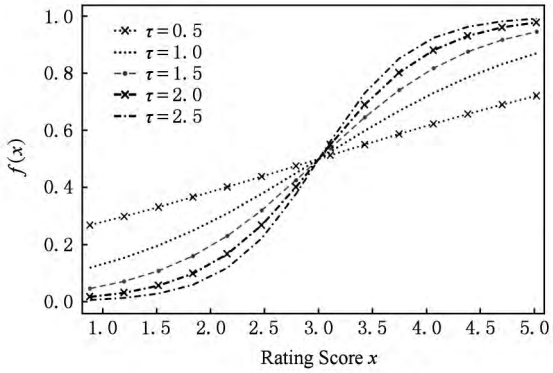

Fig. 9 The image of the mapping function with different τ value

图 9 不同 τ 值对应的映射函数图像

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

所示. 从图 9 可以看出，当参数 $\tau=1.5$ 时，1～5 分的映射权重比较合理，既弱化了高分间的差距，又不至于像 $\tau=2.5$ 时那样差距过小，当 τ 过小比如 $\tau=0.5$ 时，可以发现映射后的权值基本还是线性关系，没有起到弱化高分之间的差距的作用.

参数 α 与 β 是为了平衡图模型中上述函数映射的权重与基于相似度的权重之间的差异，当 $\tau=1.5$ 时，评分为 5 分时映射后的权重最大，为 0.95，由于相似度的最大值可能大于 0.95，也可能远小于 0.95，因此需要用 α 与 β 去平衡这种差异.

参数 α 对模型性能的影响情况如表 3 所示，除了 α 外，其他参数同 3.4 节相同，从表 3 我们可以看出，在 MovieLens-100K 数据集上，本模型在不同推荐算法下取得最佳性能的 α 约为 1.3，在 MovieLens-1M 数据集上取得最佳性能的 α 约为 1.1，而在 Epinions 数据集上就需要 α 达到 1.9. 在 3 个数据集上，*MAE* 值都随着 α 的增大而呈现先降低后升高的趋势，说明 α 过小会降低用户相似度的重要性，而导致聚类效果变差；如果 α 过大时，聚类主要取决于用户相似度了，因此模型退化为基于用户的聚类模型. 同理可以分析，当 β 过大时，模型就退化为基于物品的单侧聚类模型，而表 4 的实验结果也证实了这一分析，观察表 4 可以发现，取得最好性能时的 β 值与我们 3.4 节所设置的一致.

Table 3　Comparison of *MAE* Results with Different α on Three Datasets

表 3　不同 α 值在 3 个数据集上的 *MAE*

Method	MovieLens-100K($c=20$)					MovieLens-1M($c=25$)					Epinions($c=30$)				
	$\alpha=1.1$	$\alpha=1.2$	$\alpha=1.3$	$\alpha=1.4$	$\alpha=1.5$	$\alpha=0.9$	$\alpha=1.0$	$\alpha=1.1$	$\alpha=1.2$	$\alpha=1.3$	$\alpha=1.7$	$\alpha=1.8$	$\alpha=1.9$	$\alpha=2.0$	$\alpha=2.1$
IB	0.7382	**0.7342**	0.7355	0.7378	0.7421	0.7301	0.7264	**0.7214**	0.7249	0.7283	0.8513	0.8489	**0.8424**	0.8458	0.8487
UB	0.7497	0.7450	**0.7422**	0.7464	0.7501	0.7373	0.7348	**0.7304**	0.7355	0.7389	0.8387	**0.8356**	0.8371	0.8424	0.8476
SVD	0.7387	0.7324	**0.7306**	0.7368	0.7410	0.7303	0.7274	**0.7236**	0.7263	0.7297	0.8285	0.8246	**0.8208**	0.8230	0.8266
NMF	0.7412	**0.7367**	0.7379	0.7390	0.7447	0.7372	0.7344	0.7310	**0.7271**	0.7311	0.8379	0.8345	**0.8321**	0.8358	0.8392
MMMF	0.7525	0.7488	**0.7467**	0.7501	0.7556	0.7260	**0.7223**	0.7236	0.7254	0.7276	0.8380	0.8344	**0.8316**	0.8377	0.8413
NPCA	0.7540	0.7502	0.7488	**0.7474**	0.7493	0.7289	0.7245	**0.7229**	0.7266	0.7291	0.8396	0.8351	**0.8337**	0.8364	0.8385

Note: The best experimental results by setting different α values under the same data set and recommendation algorithm are in bold.

Table 4　Comparison of *MAE* Results with Different β on Three Datasets

表 4　不同 β 值在 3 个数据集上的 *MAE* 结果对比

Method	MovieLens-100K($c=20$)					MovieLens-1M($c=25$)					Epinions($c=30$)				
	$\beta=1.1$	$\beta=1.2$	$\beta=1.3$	$\beta=1.4$	$\beta=1.5$	$\beta=1.0$	$\beta=1.1$	$\beta=1.2$	$\beta=1.3$	$\beta=1.4$	$\beta=1.3$	$\beta=1.4$	$\beta=1.5$	$\beta=1.6$	$\beta=1.7$
IB	0.7348	**0.7321**	0.7355	0.7384	0.7430	0.7268	0.7247	**0.7214**	0.7233	0.7259	0.8486	0.8447	**0.8424**	0.8438	0.8477
UB	0.7489	0.7453	**0.7422**	0.7436	0.7485	0.7325	**0.7287**	0.7304	0.7337	0.7368	0.8415	0.8392	0.8371	**0.8356**	0.8389
SVD	0.7376	0.7311	**0.7306**	0.7337	0.7381	0.7297	0.7264	0.7236	**0.7221**	0.7278	0.8284	0.8237	**0.8208**	0.8229	0.8260
NMF	0.7446	0.7403	**0.7379**	0.7392	0.7438	0.7371	0.7324	**0.7310**	0.7339	0.7382	0.8387	0.8348	0.8321	**0.8310**	0.8343
MMMF	0.7512	0.7480	**0.7467**	0.7495	0.7547	0.7284	0.7252	**0.7236**	0.7249	0.7275	0.8362	0.8335	**0.8316**	0.8348	0.8384
NPCA	0.7490	**0.7463**	0.7488	0.7502	0.7531	0.7288	0.7253	0.7229	**0.7216**	0.7248	0.8396	0.8367	**0.8337**	0.8351	0.8380

Note: The best experimental results by setting different β values under the same data set and recommendation algorithm are in bold.

最后，我们简单讨论参数 a 和 b 对模型性能的影响，该参数控制着特征提取过程中用户特征与物品特征的维度，实验结果如表 5 和表 6 所示. 可以看出，随着特征维度的增长，模型性能提升较快，直到达到最好性能，但是继续增大维度，*MAE* 值就会缓慢上升. 分析认为，当维度较低时，特征提取不完全，没有很好的区分度，导致相似度计算不准确；当维度较高时，有可能会出现过拟合现象，因此模型性能缓慢下降. 从实验结果可以看出，取得最好性能时的特征维度正如 3.4 节所设置.

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

Table 5 Comparison of *MAE* Results with Different *a* on Three Datasets

表 5 不同 *a* 值在 3 个数据集上的 *MAE* 结果对比

Method	MovieLens-100K(c=20)					MovieLens-1M(c=25)					Epinions(c=30)				
	a=12	a=14	a=16	a=18	a=20	a=16	a=18	a=20	a=22	a=24	a=20	a=22	a=24	a=26	a=28
IB	0.7432	0.7381	**0.7355**	0.7362	0.7384	0.7284	0.7252	**0.7214**	0.7229	0.7244	0.8483	0.8457	0.8424	**0.8389**	0.8402
UB	0.7501	0.7468	0.7422	**0.7417**	0.7425	0.7381	0.7346	0.7304	**0.7271**	0.7296	0.8434	0.8392	**0.8371**	0.8385	0.8391
SVD	0.7384	0.7338	**0.7306**	0.7315	0.7336	0.7319	0.7273	**0.7236**	0.7251	0.7274	0.8274	0.8237	**0.8208**	0.8226	0.8243
NMF	0.7417	**0.7355**	0.7379	0.7392	0.7413	0.7344	**0.7295**	0.7310	0.7326	0.7347	0.8398	0.8364	**0.8321**	0.8342	0.8357
MMMF	0.7511	0.7483	0.7467	**0.7452**	0.7472	0.7293	0.7258	**0.7236**	0.7252	0.7273	0.8382	0.8340	**0.8316**	0.8338	0.8356
NPCA	0.7542	0.7513	0.7488	**0.7470**	0.7482	0.7302	0.7264	**0.7229**	0.7241	0.7263	0.8390	0.8367	**0.8337**	0.8355	0.8372

Note：The best experimental results by setting different *a* values under the same data set and recommendation algorithm are in bold.

Table 6 Comparison of *MAE* Results with Different *b* on Three Datasets

表 6 不同 *b* 值在 3 个数据集上的 *MAE* 结果对比

Method	MovieLens-100K(c=20)					MovieLens-1M(c=25)					Epinions(c=30)				
	b=14	b=16	b=18	b=20	b=22	b=14	b=16	b=18	b=20	b=22	b=18	b=20	b=22	b=24	b=26
IB	0.7422	0.7383	**0.7355**	0.7368	0.7382	0.7274	0.7245	**0.7214**	0.7229	0.7240	0.8491	0.8457	0.8424	**0.8392**	0.8415
UB	0.7480	0.7453	**0.7422**	0.7436	0.7450	0.7379	0.7338	**0.7304**	0.7317	0.7336	0.8431	0.8398	**0.8371**	0.8384	0.8408
SVD	0.7375	0.7338	**0.7306**	0.7318	0.7332	0.7305	0.7271	**0.7236**	0.7244	0.7252	0.8274	0.8238	**0.8208**	0.8215	0.8234
NMF	0.7436	0.7394	**0.7379**	0.7392	0.7411	0.7386	0.7349	0.7310	**0.7287**	0.7301	0.8386	0.8358	**0.8321**	0.8339	0.8352
MMMF	0.7522	0.7489	**0.7467**	0.7476	0.7493	0.7287	0.7258	0.7236	**0.7218**	0.7233	0.8384	0.8352	**0.8316**	0.8328	0.8343
NPCA	0.7538	0.7511	0.7488	**0.7452**	0.7471	0.7279	0.7252	**0.7229**	0.7247	0.7259	0.8392	0.8365	0.8337	**0.8313**	0.8339

Note：The best experimental results by setting different *b* values under the same data set and recommendation algorithm are in bold.

4 总 结

本文提出了一种新的基于特征的协同聚类模型，它通过从用户评分信息中提取用户以及物品的特征，并将其应用在聚类过程中以提高聚类准确度，从而提高协同过滤算法的预测性能与推荐性能. 本文模型首先通过二次回归模型发掘用户与物品的特征，然后通过一个图模型将用户物品的特征与用户对物品的评分等信息综合起来，从中计算出最终聚类结果. 经过在 3 个真实的数据集上从预测性能与推荐性能 2 个方面来进行测试，实验结果有效地验证了本文模型的确能够更好地提升协同过滤算法的预测性能与推荐性能. 在今后的工作中，我们考虑将这种从用户评分矩阵中发掘用户物品特点的方式进行拓展，添加一些其他信息来更准确地发掘用户和物品的特征，也能更进一步提升本文模型的性能.

参 考 文 献

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

[2] Huang Zan，Zeng Danial，Chen Hsinchun. A comparison of collaborative-filtering recommendation algorithms for e-commerce[J]. IEEE Intelligent Systems，2007，22(5)：68-78

[3] Su Xiaoyuan，Khoshgoftaar T M. A survey of collaborative filtering techniques[J]. Advances in Artificial Intelligence，2009，2009(4)：421-445

[4] Pazzani M，Billsus D. Content-based recommendation systems[G]//The Adaptive Web. Berlin：Springer，2007：325-341

[5] Breese J S，Heckerman D，Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]//Proc of the 14th Conf on Uncertainty in Artificial Intelligence. San Francisco：Morgan Kaufmann，1998：43-52

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

[6] Herlocker J L, Konstan J A, Borchers A, et al. An algorithmic framework for performing collaborative filtering [C] //Proc of the 22nd Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 1999: 230-237

[7] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms [C] //Proc of the 10th Int Conf on World Wide Web. New York: ACM, 2001: 285-295

[8] Miyahara K, Pazzani M J. Collaborative filtering with the simple Bayesian classifier [C] //Proc of the 6th Pacific Rim Int Conf on Artificial Intelligence. Berlin: Springer, 2000: 679-689

[9] Vucetic S, Obradovic Z. Collaborative filtering using a regression-based approach [J]. Knowledge and Information Systems, 2005, 7(1): 1-22

[10] Hofmann T, Puzicha J. Latent class models for collaborative filtering [C] //Proc of the 16th Int Joint Conf on Artificial Intelligence. San Francisco: Morgan Kaufmann, 1999: 688-693

[11] Liu Jing, Jiang Yu, Li Zechao, et al. Domain-sensitive recommendation with user-item subgroup analysis [J]. IEEE Trans on Knowledge and Data Engineering, 2016, 28(4): 939-950

[12] Guo Hongyi, Liu Gongshen, Su Bo, et al. Collaborative filtering recommendation algorithm combining community structure and interest clusters [J]. Journal of Computer Research and Development, 2016, 53(8): 1664-1672 (in Chinese)
(郭弘毅, 刘功申, 苏波, 等. 融合社区结构和兴趣聚类的协同过滤算法[J]. 计算机研究与发展, 2016, 53(8): 1664-1672)

[13] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems [J]. IEEE Computer, 2009, 42(8): 30-37

[14] Sarwar B, Karypis G, Konstan J, et al. Application of dimensionality reduction in recommender system—A case study, TR-00-043 [R]. New York: ACM, 2000

[15] Lee D D, Seung H S. Algorithms for non-negative matrix factorization [C] //Proc of the 14th Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2001: 556-562

[16] Srebro N, Rennie J D M, Jaakkola T S. Maximum-Margin matrix factorization [C] //Proc of the 17th Conf on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2004: 1329-1336

[17] Yu Kai, Zhu Shenghuo, Lafferty J, et al. Fast nonparametric matrix factorization for large-scale collaborative filtering [C] //Proc of the 32nd Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2009: 211-218

[18] Pham M C, Cao Yiwei, Klamma R, et al. A clustering approach for collaborative filtering recommendation using social network analysis [J]. Journal of Universal Computer Science, 2011, 17(4): 583-604

[19] Esslimani I, Brun A, Boyer A. A collaborative filtering approach combining clustering and navigational based correlations [C] //Proc of the 5th Web Information Systems and Technologies. Setubal, Portugal: INSTICC Press, 2009: 364-369

[20] Chen Chaochao, Zheng Xiaolin, Wang Yan, et al. Context-aware collaborative topic regression with social matrix factorization for recommender systems [C] //Proc of the 28th Association for the Advancement of Artificial Intelligence. Menlo Park, CA: AAAI, 2014: 9-15

[21] Huang Mengxing, Sun Longfei, Du Wencai. Collaborative filtering recommendation algorithm based on item attributes [C] //Proc of the 15th IEEE/ACIS Int Conf on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). Los Alamitos, CA: IEEE Computer Society, 2014: 1-6

[22] Nilashi M, Jannach D, bin Ibrahim O, et al. Clustering-and regression-based multi-criteria collaborative filtering with incremental updates [J]. Information Sciences, 2015, 293: 235-250

[23] Jiang Yu, Liu Jing, Zhang Xi, et al. TCRec: Product recommendation via exploiting social-trust network and product category information [C] //Proc of the 22nd Int Conf on World Wide Web. New York: ACM, 2013: 233-234

[24] Zhang Xi, Cheng Jian, Yuan Ting, et al. TopRec: Domain-specific recommendation through community topic mining in social network [C] //Proc of the 22nd Int Conf on World Wide Web. New York: ACM, 2013: 1501-1510

[25] Sarwar B M, Karypis G, Konstan J, et al. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering [C] //Proc of the 5th Int Conf on Computer and Information Technology. Dhaka: East West University, 2002: 27-28

[26] Xue Guirong, Lin Chenxi, Yang Qiang, et al. Scalable collaborative filtering using cluster-based smoothing [C] //Proc of the 28th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2005: 114-121

[27] O'Connor M, Herlocker J. Clustering items for collaborative filtering [C] //Proc of the 22nd ACM SIGIR Workshop on Recommender Systems. New York: ACM, 1999

[28] Ungar L H, Foster D P. Clustering methods for collaborative filtering [C] //Proc of the 15th Association for the Advancement of Artificial Intelligence Workshop on Recommendation Systems. Menlo Park: AAAI, 1998: 114-129

[29] George T, Merugu S. A scalable collaborative filtering framework based on co-clustering [C] //Proc of the 5th IEEE Int Conf on Data Mining. Los Alamitos, CA: IEEE Computer Society, 2005: 625-628

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

[30] Xu Bin，Bu Jiajun，Chen Chun，et al. An exploration of improving collaborative recommender systems via user-item subgroups [C] //Proc of the 21st Int Conf on World Wide Web. New York：ACM，2012：21-30

[31] Li Cong，Liang Changyong，Ma Li. A collaborative filtering recommendation algorithm based on domain neighbor [J]. Journal of Computer Research and Development，2008，45(9)：1532-1538 (in Chinese)

(李聪，梁昌勇，马丽. 基于领域最近邻的协同过滤推荐算法[J]. 计算机研究与发展，2008，45(9)：1532-1538)

[32] Belkin M，Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering [C]//Proc of the 14th Conf on Neural Information Processing Systems. Cambridge，MA：MIT Press，2001：585-591

Zhang Fei，born in 1992. Received his bachelor degree in computer science from Nankai University. Master candidate in the University of Chinese Academy of Sciences，Beijing，China. His main research interests include recommendation system and big data.

Zhang Libo，born in 1989. Received his PhD degree in computer software and theory from the University of Chinese Academy of Sciences. Assistant professor in the Institute of Software，Chinese Academy of Sciences. His main research interests include image processing，pattern recognition，knowledge graph and deep learning.

Luo Tiejian，born in 1962. PhD，professor. He is also the Director of the Information Dynamic and Engineering Applications Laboratory. His main research interests include Web mining，large scale Web performance optimization and distributed storage systems (tjluo@ucas.ac.cn).

Wu Yanjun，born in 1979. He received his PhD degree in computer software and theory from Institute of Software，Chinese Academy of Sciences in 2006. Researcher. His main research interests include operating system and artificial intelligence (yanjun@iscas.ac.cn).

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net