第 43 卷 第 6A 期
2016 年 6 月

计 算 机 科 学
Computer Science

Vol. 43 No. 6A
June 2016

基于两层社区混合计算的个性化推荐方法

黄亚坤 王 杨 苏 洋 陈付龙 赵传信
(安徽师范大学数学计算机科学学院 芜湖 241000)

摘 要 社区发现在个性化推荐系统中有着良好的应用。考虑到具有联系的不同层次社区之间能够构成一种混合的计算模型(HCPR),将该混合计算模型从用户-项目关系图演化到三维立体混合计算模型中,采用不同的融合相似度分别构建项目层社区和用户层社区,并基于用户-项目之间关注-被关注关系定义混合计算层。提出了一种基于两层社区混合计算的个性化推荐方法,面对新用户、旧用户、新项目、旧项目的不同输入定义相应的计算,其能推荐较为精准、个性化的信息。在 3 种不同类型的数据集上进行了实验,结果表明该模型能够较好地表示用户之间、项目之间以及用户和项目之间的关系,与 U-CF 和 I-CF 的推荐方法相比,HCPR 借助构建的混合计算层在保证推荐精确度的同时,推荐结果更为个性化。

关键词 层级社区,社区发现,混合计算,个性化推荐
中图法分类号 TP391 文献标识码 A

Personalized Recommendation Method Based on Hybrid Computing in Two Layers of Community

HUANG Ya-kun WANG Yang SU Yang CHEN Fu-long ZHAO Chuan-xin
(School of Mathematics & Computer Science, Anhui Normal University, Wuhu 241000, China)

Abstract Researching the inner structure of social network has great performance in community detection. A hybrid computing model can be constructed by the different levels of communities which have contact between them. Considering the hybrid computing model in two-layers community, we applied it to personal recommendation system. The method makes evolution from users-items diagram into three dimensional hybrid computing model, and constructs the different layer communities respectively by the fusion similarity. We also defined the hybrid computing layer based on the relationship in users and items. Defining different computing for new user, old users, new items and old items, HCPR can recommend the precise and diverse information. The experiments result show that the model has great performance in representing the relationship between users and items. Compared to the U-CF and I-CF, HCPR can ensure the precise of the recommendation and rich diversity.

Keywords Hierarchy community, Community detection, Hybrid computing, Personalized recommendation

1 引言

随着互联网规模的飞速增长和“互联网+”时代的到来,无处不在的网络、计算、数据、信息遍布于当今社会。例如,“互联网+”时代的电子商务,特别是移动互联网对原有的传统行业起到了巨大的升级换代的作用,用户如何从多域海量数据中获取有价值的信息成为当前研究的热点。个性化推荐作为一种重要的信息过滤手段,通过分析用户的兴趣偏好和历史行为,主动向用户推荐其感兴趣的项目,有效解决了互联网的信息超载问题[1]。典型的基于协同过滤的推荐系统从其他类似的用户或项目中收集评价信息来预测给定用户的偏好[2]。协同过滤推荐已经成为“互联网+”时代下的电子商务、社交网络、金融、智慧城市、旅游、教育等方面应用的核心内容[3]。

然而,传统的推荐系统大多仅针对用户-项目关系评分矩阵,忽略了用户自身存在的社会关系以及项目之间存在的社区关系。事实上,处于同一年龄段或者同一职业的人具有的兴趣偏好相似性可能更大。此外,通常某人喜欢某一类型的电影、音乐,则其对相似类型、品味的喜好的可能性也较大。因此,推荐系统不仅需要深层次挖掘用户-项目间的复杂关系,还需考虑用户之间和项目之间存在的关系,从而提高推荐的精准率。文献[4]中提出了一种两层混合图模型,用户之间基于小世界理论构建社区,项目之间基于贝叶斯网络进行推荐。文献[5]从用户-项目评分角度进行个性化推荐,上述基于内容和同过滤等方法存在以下 3 个问题。

1)评分数据稀疏性

考虑到当前个性化推荐系统中用户与项目数据的海量性,协同过滤推荐基于用户对项目的评分矩阵将更加稀疏,稀

本文受国家自然科学基金(61572036),教育部人文社科青年基金(11YJC880119),安徽省高校人文社会科学类研究重大项目(SK2014ZD033),安徽省教育科学规划课题项目(JG14022)资助。
黄亚坤(1992—),男,硕士,主要研究方向为数据挖掘、机器学习、个性化推荐,E-mail:hyk-it@foxmail.com;王 杨(1971—),男,博士,教授,主要研究方向为数据挖掘、机器学习、智能 Agent 等,E-mail:wycap@126.com(通信作者)。

疏矩阵在协同过滤算法中的推荐精度以及推荐结果的多样性存在着较大的误差,甚至有时候推荐结果完全不正常,使得推荐质量无法提高。特别是“互联网+”时代的电子商务,项目的数量已经达到亿级别,即使是活跃用户的评价也无法做到对大部分项目的评分。因此,稀疏性成为影响基于用户-项目评分推荐方法的重要因素。

2)过滤冷启动

无论是基于内容还是协同过滤的算法,均无法避免“冷启动”问题。当前大多数推荐算法从用户-项目之间的评分关系挖掘基于用户和项目的历史信息,但当新用户或新项目加入到系统中时,无法进行建模,即无法进行推荐。这样将导致这些新加入的用户或项目不会出现在推荐列表中,对推荐系统没有任何作用。“冷启动”不仅影响新加入用户或项目的推荐服务,而且对系统中已存在的用户或项目在关联上也造成一定的影响。

3)推荐可扩展性

推荐系统中需要考虑海量用户、项目推荐的实效性,因此面临着巨大的计算量,对服务器提出了更高的要求。虽然通过不断增加服务器等硬件设施在一定程度能够解决问题,但随着用户、项目数据量的增长,推荐系统的成本的增加需求对推荐系统提出了提前预测用户相关信息的要求。综上所述,基于协同过滤的推荐在数据稀疏性、“冷启动”以及推荐的可扩展性方面还存在一定的问题。为了综合考虑系统中用户-项目关系、用户-用户关系以及项目-项目关系,本文从社区发现和混合计算角度提出了一种两层混合计算框架,主要贡献有以下两方面:

①分别挖掘项目层与用户层内存在的社区结构,结合用户-项目二维关系,提出一种用户层社区和项目层社区混合计算的推荐模型(A Method of Personalized Recommendation Based on Hybrid Computing-HCPR),区别于传统方法仅挖掘用户-项目之间的关系,它更深层次地在推荐过程中融合了用户自身之间以及项目自身之间存在的关系,更精确地进行了社区构建与推荐,形成了三维立体的推荐模型,一定程度上解决了新用户、项目的“冷启动”、数据稀疏以及可扩展问题。

②实际不同类型的3个数据集进行仿真实验,从推荐结果的查准率、查全率、多样性和平均绝对误差对推荐结果进行验证,结果表明本文提出的混合计算推荐方法能够在保证推荐结果一定准确率的前提下,在多样性方面相比基于协同过滤的推荐系统具有较大的提高。

2 相关工作

2.1 个性化推荐

个性化推荐服务是分析用户的历史行为及其兴趣偏好等特征,利用这些特征从海量信息中挖掘满足用户需求或引导用户的相关兴趣,进而推荐给目标用户。个性化推荐系统的关注点在于精准性和多样性。推荐系统的智能性对个性化推荐服务的质量具有重要的影响。推荐系统主要从用户和推荐对象进行建模,通过推荐算法计算结果提供服务,并对推荐系统的性能进行评价反馈来提高服务质量。根据不同的推荐准则,个性化推荐系统主要包括协同过滤系统、基于内容的推荐系统、基于内容和协同过滤混合策略的推荐系统。

基于内容的推荐系统主要根据用户的历史选择项目,从被推荐项目中选择与其相似的项目作为被推荐的目标。该方法对推荐项目的特征提取要求较高,且存在新用户“冷启动”问题,同时在处理海量数据项目时也难以获得满意的服务质量;此外,在面对不同语言描述的用户或项目时,兼容性也是基于内容的推荐需要解决的首要问题之一。

协同过滤推荐根据用户节点的相邻用户的评分进行推荐,主要分为基于用户的协同过滤推荐(User-based Collaborative Filtering)、基于项目的协同过滤推荐(Item-Based Collaborative Filtering)以及基于模型的协同过滤推荐(Model-Based Collaborative Filtering)。U-CF主要指用户最终的选择推荐结果是根据其朋友向用户的推荐结果。而I-CF则是基于用户对推荐项目本身的信任关注,即找到目标对象的最近邻居,由于当前用户对最近邻居的评分与对目标推荐对象的评分比较相似,因此将当前用户对最近邻居的评分预测作为其对目标推荐对象的评分,然后选择预测评分最高的Top-N对象作为推荐结果,如文献[2,6]。M-CF则利用用户对历史项目信息的评分得到该用户的模型[7],进而对某项目进行打分。通常采用机器学习模型[8]、统计模型模型[9]和贝叶斯模型[10]等来建立用户的模型。

基于社会网络分析的算法也有较多的研究,主要把社会网络理论在协同过滤的基础上进行延伸拓展。用户在浏览项目或者对项目进行评价时,形成了用户与项目之间的联系,即用户-项目社会网络关系。因而通过社会网络分析方法可以从多方面(用户之间、项目之间、用户和项目之间)挖掘节点之间的关联性,依据此进行推荐。Massa等人[11]通过节点当前存在的关系计算节点信任度,基于信任度进行推荐,从而能够比一般协同过滤方法获得的推荐结果更为精确,但在多样性方面有所欠缺。

2.2 社区发现

根据社会网络的属性的不同,研究者从不同角度提出了许多社区结构划分算法。考虑社区划分过程中是否包含网络中的所有节点,可分为静态计算和动态计算两种。静态计算的社区发现主要包括模块度最优算法、多目标优化算法、基于概率模型的算法以及信息编码算法。典型的模块度最优算法有经典贪心算法,Mark Newman基于贪心思想提出基于模块度最大的贪心算法FN[12]。此外最优化模块度的启发式算法还有模拟退火法和极值优化算法等;多目标优化算法主要针对传统基于模块度的社区优化算法的固有缺陷,采用如基于元胞自动机的社区发现算法[13];基于概率模型的算法主要有基于混合模型的算法,如Newman等人[14]基于混合模型和期望最大化,提出一种面向有向网络的社区发现算法。另外还有基于LDA的概率模型算法[15];信息编码算法主要有Infomap算法,如Martin Rosvall等人[16]基于信息论提出了Infomap算法。该算法用随机游走作为网络上信息传播的代理,网络上的随机游走会产生相应的数据流。

动态计算中主要有派系过滤算法、基于相似度的聚合算法以及标签传播算法等。典型的派系过滤中提出了重叠社区问题,最初由Gergely Palla等人提出。CPW算法[17]、CPMw算法都是用于重叠社区发现的派系过滤算法;EAGLE算法[18]基于相似度发现具有层次化结构的虚拟社区;标签传播[19]是基于图的半监督学习算法,采用已标记节点的标签信息来预测未标记节点的标签信息。为了适合在线社交网络的

重叠虚拟社区结构，Steve Gregory 基于 LPA 对 LPA 进行扩展，提出了 COPRA 算法[20]。

3 整体框架

图 1 给出了基于两层社区交叉验证个性化推荐方法的框架。首先，从 User-Profile 中综合分析用户的自身特征属性以及用户的历史行为信息，以用户之间的融合相似性作为准则，构建具有相似兴趣偏好的用户社区；再根据被推荐项目信息 Item-DB 的特征分析，基于项目融合相似性分析发现项目层社区；根据构建的用户社区层和项目信息社区层的联系，挖掘用户社区与项目社区之间的特征关系，采用社区间混合计算的方式处理需要推荐的用户或者项目资源；同时，采用该推荐框架能够有效解决新用户加入带来的“冷启动”问题，混合计算的推荐方式对于新的项目资源也能够有效、及时进行推荐。为了准确、多样地获得混合计算的推荐信息，需要能够对被推荐项目的属性以及用户的属性和行为特征进行获取、预处理、规格化表示等。

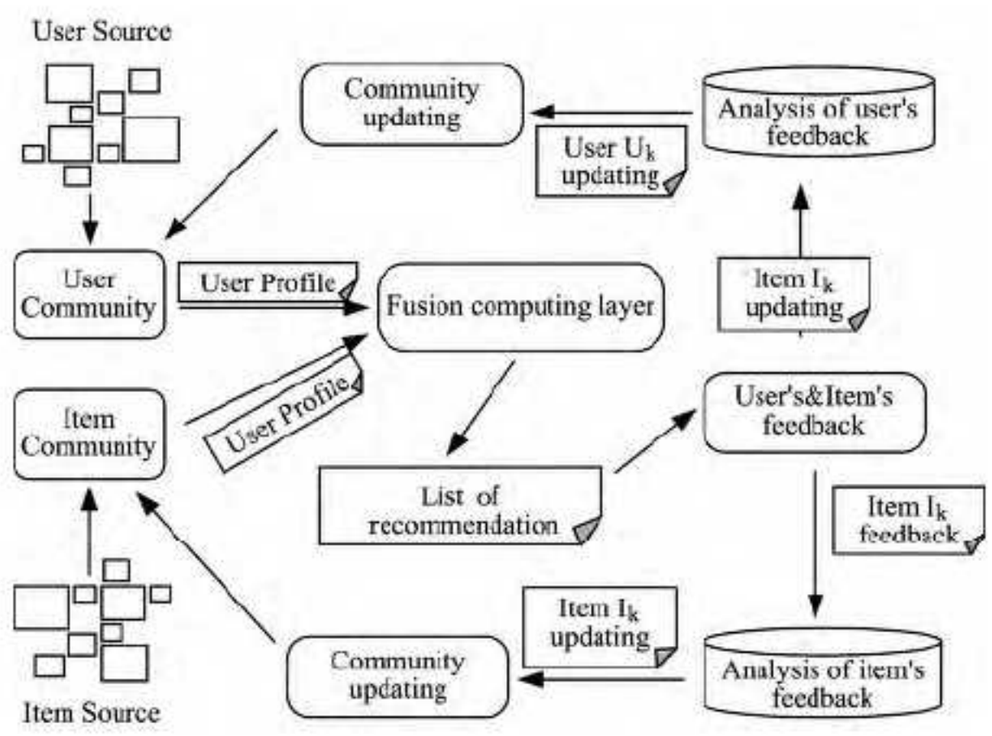


图 1 个性化推荐整体框架

3.1 用户建模

用户模型对系统中用户的信息进行获取、表示、存储和预处理。目前，个性化推荐系统面向的不仅是结构化数据表示的用户信息，还存在非结构化的行为信息等。针对用户模型的结构化与非结构化信息，主要从以下 3 个方面进行建模：

1) 用户基本属性信息。推荐系统中的主体用户通常具有用户姓名、昵称、性别、年龄、社交联系方式等属性，此外不同领域的推荐系统一般还具有该领域相关的属性选项。

2) 用户反馈信息。用户反馈信息包括注册登录时系统推送的相关领域问题的填、选信息，例如用户的偏好关键词和标签等；此外，用户对系统中的项目信息进行反馈评论、建议的信息也是一种重要的非结构化信息来源，通常包括用户对项目的评级或评分。

3) 用户历史行为。一般系统的服务器日志中保存了用户的历史浏览和操作行为，包括用户的用户搜索关键词、用户搜索关键词、用户页面留存时间、页面访问的频率、具体按钮的点击次数、浏览过程中涉及到的收藏、保存操作等。

本文基于向量空间模型构建用户模型。定义用户模型三元组 $User\text{-}M=\langle BI, FB, HV\rangle$。其中 $BI=\{name, age, job, \ldots\}$ 向量表示用户基本信息；$FB=\{m_1, m_2, \cdots, m_n\}$ 向量表示用户的反馈信息；HV 为一个 n 维特征向量，表示 n 个兴趣特征：

$$HV=\{h_1, h_3, \cdots, h_n\}$$
$$=\{(ke_1, we_1, \tau_1), (ke_2, we_2, \tau_2), \cdots, (ke_n, we_n, \tau_n)\}$$

其中，$h_i=(ke_i, we_i, \tau_i)$ 为用户兴趣的第 i 个特征节点；ke_i 为兴趣信息；we_i 为兴趣信息的权重；τ_i 为兴趣更新的最近时间。

3.2 推荐项目建模

在不同的推荐系统应用领域，推荐对象也各不相同，具体包括文章、音乐、电影、电视节目、菜谱、旅游景点等内容的推荐。其中，基于文本的特征提取方法已经较为成熟，而主要面向非结构类数据的多媒体对象的特征提取方法仍是研究热点。在确定了推荐对象后，首要的任务就是考虑如何结合对象的特点提炼对象特征以及考虑这些特征对推荐算法的影响。若当前的被推荐项目资源中包含 n 个不同的特征属性，则定义如下的结构向量：

$$ISV=\{f_1, f_2, \cdots, f_n\}$$
$$=\{(inf_1, fre_1, wlist_1), (inf_2, fre_2, wlist_2), \cdots, (inf_n, fre_n, wlist_n)\}$$

其中，$(inf_1, fre_1, wlist_1)$ 是项目资源库中第 i 个特征可索引到的信息；inf_i 是第 i 个关键词；fre_i 是特征的频数，即项目包含该特征属性的信息数；$wlist_i$ 是包含特征数 inf_i 的项目及该特征在项目中的权占比。如果有 m 个项目包含特征属性 inf_1，则可表示为：

$$wlist_i=\{(di_1, wi_1), (di_2, wi_2), \cdots, (di_m, wi_m)\}$$

其中，di_k 是包含特征 k_i 的项目；wi_k 是该特征在项目 di_k 中的权重。

4 两层社区混合计算推荐模型

通过对用户和项目特征进行建模，传统基于内容和协同过滤以及基于社会网络用户社区发现的推荐方法大多分析用户与用户、用户与项目、项目与项目中一个或两个之间的关系，本文将其推广到研究三者之间的混合关系。建立三者之间的两层社区混合模型，并将两层之间的用户与项目关系层定义为混合计算隐含层。图 2 所示为用户-项目关系二维图，图 3 为对二维用户项目关系图构建的立体两层混合社区模型，最上层为根据项目特征建模构建的社区，最底层为根据用户特征建模以及不同用户之间的协同关系构建的社区，根据用户-项目之间的多维立体关系建立中间混合计算层，对用户进行多样、精准化的推荐。

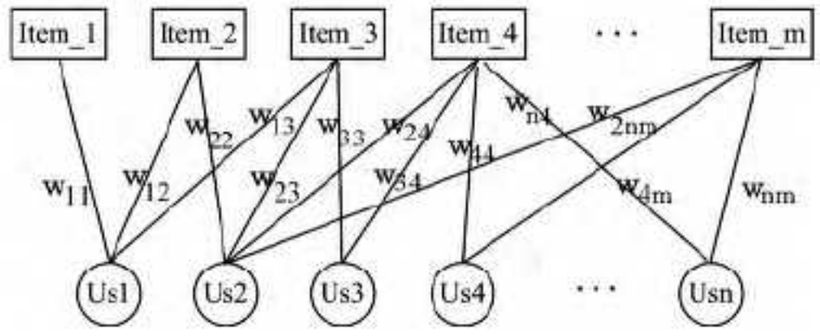


图 2 用户项目关系图

图 2 中给出了推荐系统中用户与项目之间存在的 3 种关系，即用户与用户、用户与项目和项目与项目之间的关系。最上层是构建的项目社区层，根据被推荐项目的特征属性挖掘发现出项目之间存在的社区；最底层是用户社区层，该层的社区发现主要依据于用户自身特征和用户行为特征两方面。推荐系统挖掘出用户与项目之间的精准关系，以为用户推荐更精准、多样的服务。而用户与项目之间的关系依据于上述构

建的项目社区层和用户社区层，两层社区相互交叉混合计算，得出有效的推荐结果。

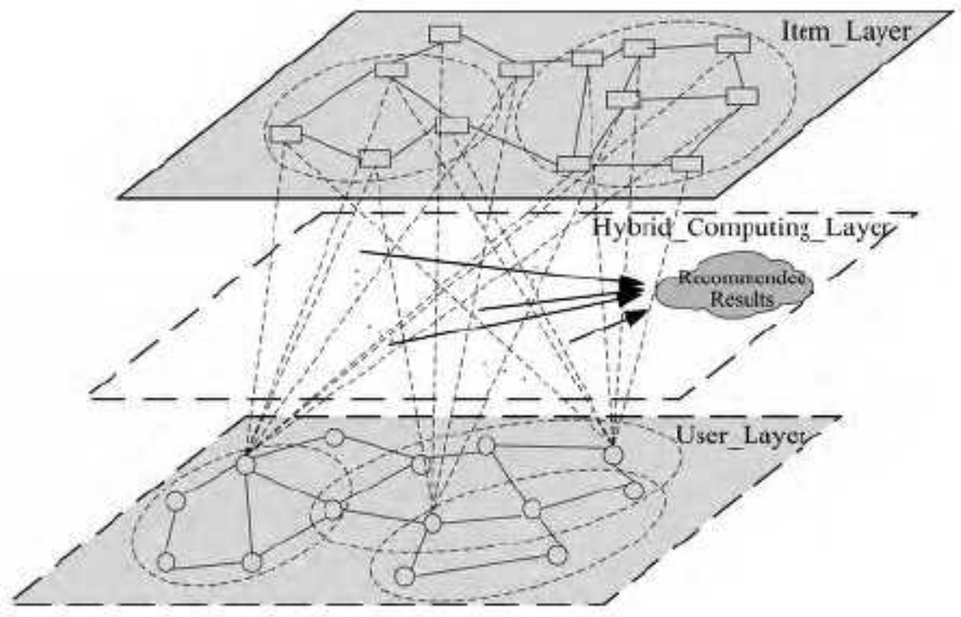


图 3　两层社区混合计算模型

为了更好地描述项目层与用户层社区发现，针对构建形成的网络社区和网络节点给，出以下相关定义。

定义 1(网络社区)　给出无向网络 $G=(V,E)$，V 是点的集合，$E=\{e=(u,v)\mid u\in V,v\in V\}$ 表示边的集合。大小为 $|V|*|V|$ 的矩阵 A 表示 G。下式给出了 A 的计算公式：

$$A_{ij}=\begin{cases}1, & e_{ij}\in E\\ 0, & e_{ij}\notin E\end{cases} \tag{1}$$

$D=\{V_1,V_2,\cdots,V_m\}$ 用来表示网络 G 的一种划分，V_i $(i=1,2,\cdots,m)$ 是按照某种特定的规则而形成的稳定社区。

定义 2(网络节点)　$N=\langle Id,Th,De\rangle$ 是一个三元组表示网络节点。Id 是节点的序列编号，Th 表示节点的主要信息，De 表示节点的度。通常，De 用来区分单节点社区节点和重叠社区节点。

$$influence(N_i,D_j)=\frac{De(N_i,D_j)}{\sum_{j=1}^{m}De(N_i,D_j)} \tag{2}$$

上式给出了节点在社区中的影响，$\sum_{j=1}^{m}De(N_i,D_j)$ 为节点的度，$De(N_i,D_j)$ 表示社区 D_j 中节点 N_i 的度，$influence(N_i,D_j)$ 用于判断节点的类型。

4.1　基于相似度的社区发现

4.1.1　项目社区层发现

区别于其他社区发现的个性化推荐算法，大多数方法仅考虑用户层的社区发现，由于被推荐项目的社区层构建不仅与自身项目属性相关，还应注意用户关注、评分对项目层社区发现的影响。本文针对项目层中被推荐项目之间的相似度采用对不同因素导致的相似度进行融合表示，下式给出了基于 Gower 模型[21]的被推荐项目用户关系之间多种相似度的融合相似度。

$$Sim_{Item}(x,y)=\frac{\sum_{k=1}^{d}w_{xyk}sim_k(x,y)}{\sum_{k=1}^{d}w_{xyk}} \tag{3}$$

$sim_k(x,y)$ 为第 x 个样本和第 y 个样本在第 k 个属性上的相似性，若其第 k 个属缺失，则取值 0，否则取值 1，这里采用基于 Jaccard 的相似度计算公式：

$$sim_k(x,y)=\frac{|N_x\bigcap N_y|}{|N_x\bigcup N_y|} \tag{4}$$

其中，N_x 表示节点 x 的领域，$|*|$ 表示集合的势，即元素的个数。基于向量模型计算项目特征之间的相似度仅考虑项目自身的属性，而在推荐系统中，项目层社区构建的作用是向用户进行推荐，该层与用户的联系较为紧密，通常用户行为对项目层社区的构建具有重要的影响。采用加权的方式综合考虑较为合理、完整地反映出这两种因素对被推荐项目的相似度影响，从而更高效地为后文进行混合计算推荐提供有效的社区基础。

4.1.2　用户社区层发现

为了综合考虑用户评分相似性和自身属性相似性，采用加权融合的相似度给出了用户社区构建的相似度公式为：

$$Sim_{hy\to}(x,y)=\alpha Sim_{item\to}(x,y)+(1-\alpha)Sim_{user\to}(x,y) \tag{5}$$

基于用户网络社区的个性化推荐方法。需要首先建立用户_项目评分矩阵以得到可量化的指标。若存在 m 个用户和 n 个项目，则采用用户_项目评分矩阵 R 进行表示：

$$R=\begin{bmatrix}r_{1,1} & r_{1,2} & \cdots & r_{1,n}\\ r_{2,1} & r_{2,2} & \cdots & r_{2,n}\\ \cdots & \cdots & \cdots & \\ r_{m,1} & r_{m,2} & \cdots & r_{m,n}\end{bmatrix}$$

其中，$r_{i,j}$ 表示第 i 个用户对第 j 个项目的评分。通过该评分矩阵能够基于 Pearson 相似度计算用户之间在项目因素中产生的相似度，下面给出了 Pearson 相似度公式：

$$Sim_{item\to}(x,y)=\frac{\sum_{j=1}^{k}(r_{x,j}-\overline{r_x})(r_{y,j}-\overline{r_y})}{\sqrt{\sum_{j=1}^{k}(r_{x,j}-\overline{r_x})^2\sum_{j=1}^{k}(r_{y,j}-\overline{r_y})^2}} \tag{6}$$

其中，$\overline{r_x}$ 和 $\overline{r_y}$ 分别代表两个用户的平均打分。考虑到用户自身属性之间的相似性较大，其在兴趣偏好也存在较大的相似性，因此基于 3.1 小节中建立的用户自身属性特征，对抽象出的用户属性向量，采用不同用户向量之间夹角的余弦值表示，公式为：

$$Sim_{user\to}(U_1,U_2)=\frac{\sum_{i=1}^{n}W_{1k}\times W_{2k}}{\sqrt{(\sum_{i=1}^{n}W_{1k})(\sum_{i=1}^{n}W_{2k})}} \tag{7}$$

其中，W_{1k}、W_{2k} 分别表示项目 I_1 和 I_2 第 k 个特征项的权值，$1\leqslant k\leqslant N$，θ 越小，Sim 越大，被推荐项目之间自身属性相似度越大。

4.1.3　社区发现算法

基于上述准则描述，同时考虑到实际推荐项目数及用户数较多，基于一种层次贪心算法[22]进行社区发现，算法主要包括两个阶段，第一阶段合并社区，初始状态将每个节点作为独立社区，基于最近邻相似度最大标准决定哪些社区应该被合并；第二阶段，将第一阶段发现的社区重新视为独立节点社区，重复构建。这两个阶段重复进行，直到网络社区划分的相似度达到一个阈值

算法 1　Construction of the Hybrid Communities

```
Input: G=(V,E) ; |V| is the size of nodes& |E|is the size of edges
Output: different communities DC={c1 ,c2 ,…,cn}
1. τ ←0.65; // defining the initial threshold τ
2.   while γd>τ
3.       //Phase one
4.       while γi>τ
5.           ci←ni// each node as an independent community
6.           sim(i,j)←Cal-nei-mer(ni ,nj);
7.           if sim(i,j)>0.5 then
8.               Merge(ni ,nj);//merge the goal of nodes
```

```
9.          end if
10.     end while
11.     //Phase two
12.     while γi>τ
13.         ni←ci//each division of the community then
                regarded as independent nodes
14.         ci'←ni'// the work in the step one is repeated
15.         sim(i,j)←Cal-nei-mer(ni,nj);
16.         if sim(i,j)>0.5 then
17.             Merge(ni,nj);//merge the goal of nodes
18.         end if
19.     end while
20. end while
```

采用层次贪心算法进行社区发现，比较直观且易于实现；算法无需提前设定网络的社区数，构建的社区能够呈现网络完整的分层社区结构，由文献[22]可知在稀疏网络上，算法的时间复杂度是线性的，适合超大规模中复杂网络的社区发现。

4.2 用户_项目社区混合计算推荐算法

通过挖掘项目内部和用户内部独自存在的社区关系，如果仅从用户层或项目层角度考虑，则将目标用户的 k 个最近邻用户的相关兴趣推荐给该目标用户，或者将新项目推荐给该项目的 k 个最近邻项目的关注追随者。此种推荐方式会陷入单独社区联系的兴趣推荐，导致推荐精准性降低，例如，若仅考虑用户层社区进行推荐，则推荐结果可能带来的是同年龄段较为受欢迎的项目。而忽略了用户自身喜欢的某特定类型的项目。若结合用户针对的特定的项目层社区的推荐结果与用户层社区的推荐结果进行混合计算形成推荐结果，则更能精准地进行推荐，图 4 给出了基于两层社区的混合计算框架。计算模型根据推荐系统中对象的不同(新用户、旧用户、新项目、旧项目)，主要包括以下 4 种计算。

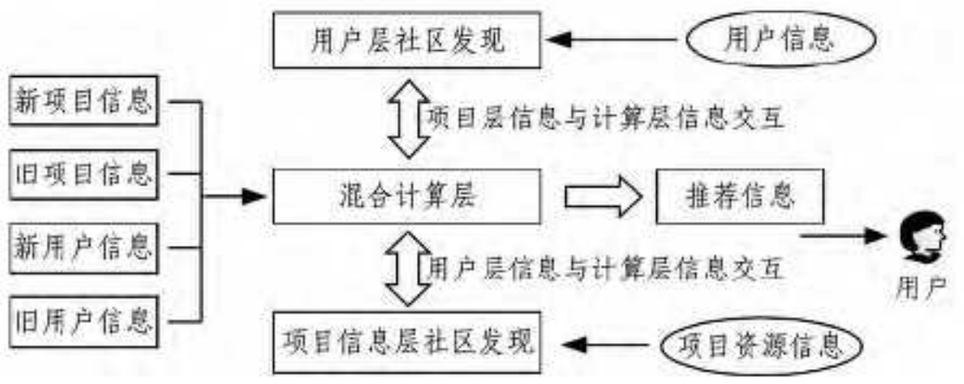


图 4 两层社区混合计算模型

首先定义两层社区的混合计算模型为：

$$THCal=(C_{user},C_{item},Cal(user|item),\alpha) \tag{8}$$

其中，C_{user} 和 C_{item} 为对已有的用户数据集和项目数据集采用算法 1 进行的社区发现结果，$Cal(user|item)$ 为对不同类型用户或者项目进行计算，α 为用户相似计算的融合因子。

1)新项目加入的计算

推荐系统中不仅需要考虑新加入用户带来的“冷启动”问题，当有新的项目资源加入时，也需要对其进行计算并推荐。对于新项目，由于其仅包含自身属性，没有用户对其的历史行为，将项目进行社区归属。

New-Item-Adding

1. item-sim←item-attribute
2. Cal(New-item)←item-sim
3. Dif(user-community)←item-community-user
4. Rec-User←Min(Sim(user-community))
5. item-sim←user-action//update

2)旧项目的推荐计算

对已存在且被用户关注的项目进行推荐时，在构建项目社区时还需要考虑该项目与用户之间的关注关系。Old-Item-Adding

1. item-sim←item-attribute&item-user-relation
2. Cal(Old-item)←item-sim
3. Dif(user-community)←item-community-user
4. Rec-User←Min(Sim(user-community))
5. item-sim←user-action//update

3)新用户加入的计算

对于新加入系统中的用户，通常分为两种，第一种为注册信息仅有用户 id 和登录密码，无其他特征的类游客用户以及未注册的游客用户，采用推荐整个社区的最热资源 top N 进行推荐；第二种，对于注册用户中填选了较多信息的新用户，在进行用户自身属性建模后，考虑该用户的历史行为信息量为 0，令 $\alpha=0$(仅计算用户自身属性的相似度)，将用户进行社区归属。

New-User-Adding

1. user-sim←user-attribute
2. Cal(New-user)←user-sim
3. Dif(item-community)←user-community-item
4. Rec-Res←Min(Sim(item-community))
5. user-sim←user-action//update

4)旧用户推荐的计算

对于系统已经活跃一段时间的用户，需要针对其进行更细化的推荐。此类型用户不仅需要考虑自身属性，更重要的是其在系统中的历史行为对推荐的计算更为重要，其中 $0<\alpha<1$。

Old-User-Adding

1. user-sim←user-attribute&user-his-action
2. Cal(Old-user)←user-sim
3. Dif(item-community)←user-community-item
4. Rec-Res←Max(Sim(item-community))
5. user-sim←user-action//update

基于上述 4 种不同形式的计算结果，推荐算法根据不同的输入 $UC=\{uc_1,uc_2,\cdots,uc_n\}$ & $IC=\{ic_1,ic_2,\cdots,ic_n\}$；$U_i|I_i$ 为输入用户参数或者项目推荐新的项目或者新的用户。

4.3 算法复杂度分析

本文提出的两层社区混合计算个性化推荐算法的复杂度主要来自社区构建和混合计算推荐两部分。用户社区和项目社区在进行社区发现前，需对用户和项目进行特征建模。由于用户及项目属性特征数相对于用户、项目数量的建模计算复杂度为 $O(n)$，本文基于融合相似度采用贪心策略进行层次社区发现。考虑到推荐系统中用户、项目数据的海量性，且形成的社区是稀疏的这一特性，该算法在大规模网络中的算法复杂度是线性的[22]。

当社区发现处于稳定状态后，主要计算源于新项目、新用户的加入。考虑新用户、新项目加入，若当前社区数为 n，社区中核心节点数为 m。对新加入的用户进行社区归属的计算时间复杂度为 $O(n*m)$；对于系统中已存在的用户、项目，其推荐是通过计算其所处社区的其他邻居对应的混合层中的邻居。若当前节点为 v，其关注的项目数为 p，其关注的社区考虑与其相似度大于 0.5 的邻居节点数为 n，其所有邻居节点的关注项目社区包含的项目数为 m，需要计算 m 个项目之间与 p 之间的相似性，时间复杂度为 $O(m*p)$。

5 实验分析

5.1 数据集及实验设置

使用以下 3 个不同类型数据集对本文提出的模型进行性能评估。

· MovieLens

MovieLens 数据集中包含了 943 个独立用户对 1682 部电影的 10000 次评分的数据，用户对自己看过的电影进行评分，等级为 1～5。我们对其中的用户和项目分别构建社区层。

· BookCrossing

该数据集是 Book-Crossing 图书社区的 278858 个用户对 271379 本书进行的评分，包括显式和隐式的评分，评分等级为 1～10。这些用户的姓名、年龄属性都以匿名的形式保存并供分析。这个数据集是由 Cai-Nicolas Ziegler 使用爬虫程序在 2004 年从 Book-Crossing 图书社区上采集的。

· RestaurantComments

该数据集是使用爬虫从某点评网站上获取的 2014 年北京 4000 家餐厅 140 万条点评数据，评分等级为 1～5。对其中的用户和餐厅分别进行社区构建，进行混合计算推荐。

由于数据集 MovieLens、RestaurantComments 采用 1～5 分评分，提取评分大于 3 的项目作为用户喜好的项目；BookCrossin 中采用的是 1～10 评分制，提取大于 6 的项目作为用户喜好的项目。将用户_项目数据集分为社区构建训练集和测试集两部分，定义 θ 为训练集所占的百分比数。从数据集 MovieLens 中随机抽取到用户 ID 为 1524 的节点进行推荐测试；从 Book-Crossing 中随机抽取到用户 ID 为 11047 的节点作为测试用户；在 RestaurantComments 中对清蒸菜类型的餐馆进行推荐测试。

5.2 推荐结果评价

5.2.1 评价标准

实验中主要采用以下几种指标进行评价。

(1) 查准率(Precision)[23]

$$Precision=\frac{|P_i \bigcap TE_i|}{|P_i|} \tag{9}$$

(2) 查全率(Recall)[23]

$$Recall=\frac{|P_i \bigcap TE_i|}{|TE_i|} \tag{10}$$

其中，P_i 为用户 u_i 的最终 top-N 推荐；TE_i 为用户 u_i 的测试集。

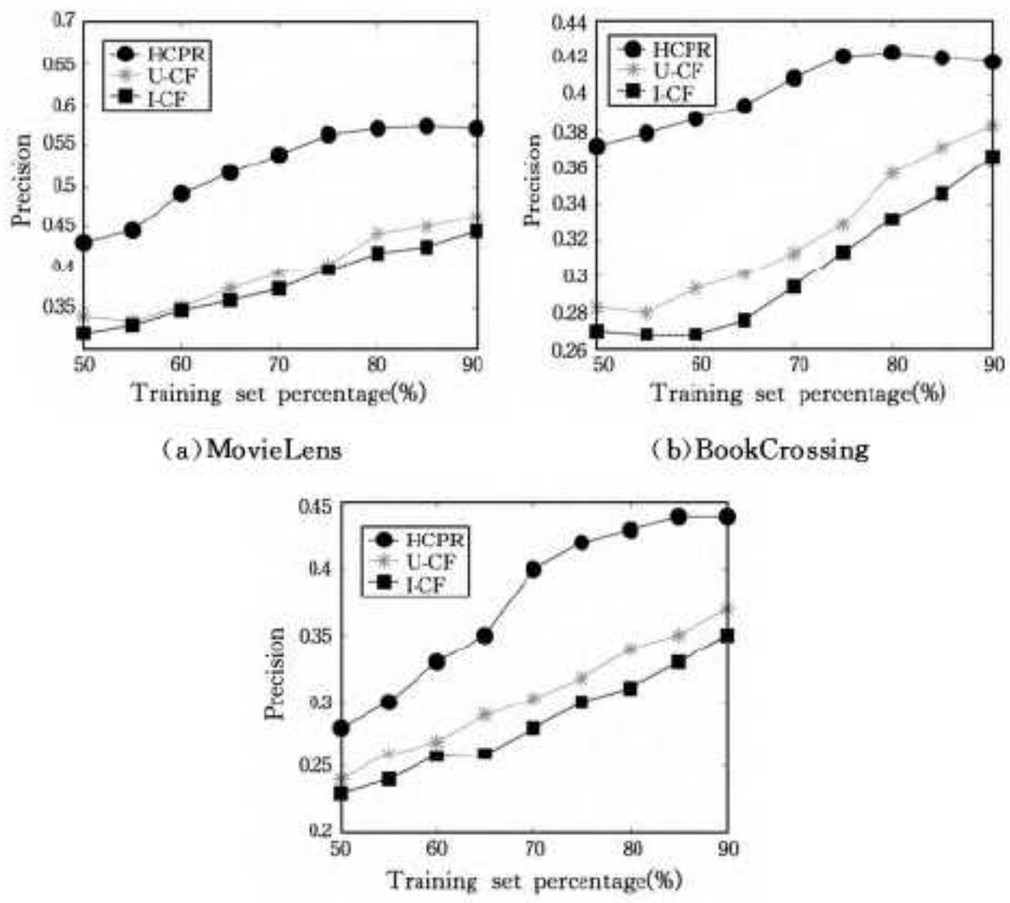
(3) 多样性评估 ILS(Intralist Similarity)[24]

$$ILS(P_i)=\frac{1}{2}\sum_{b_f \in P_i}\sum_{b \in P_i, b_f \neq b_e} g(b_f, b_e) \tag{11}$$

其中，b_f 和 b_e 为推荐列表 P_i 中的推荐项目；$g(b_f, b_e)$ 定义为 b_f 和 b_e 的相似度。$ILS(P_i)$ 越小，表示推荐列表 P_i 中的项目种类相似度越小，推荐的多样性越好。

5.2.2 实验结果与分析

查准率直观地表达了推荐结果的效果。图 5 中 3 个不同类型数据集实验的结果显示：虽然查准率在不同数据集上的结果具有一定的差异性，如查准率在 Movielens 中高于 Bookcrossing 和 Restautrant Comments，但是分析数据集可知 Movielens 中的数据集信息更为全面、有效，因而在构建社区时，社区内部的稳定性更高，推荐的精度更高。

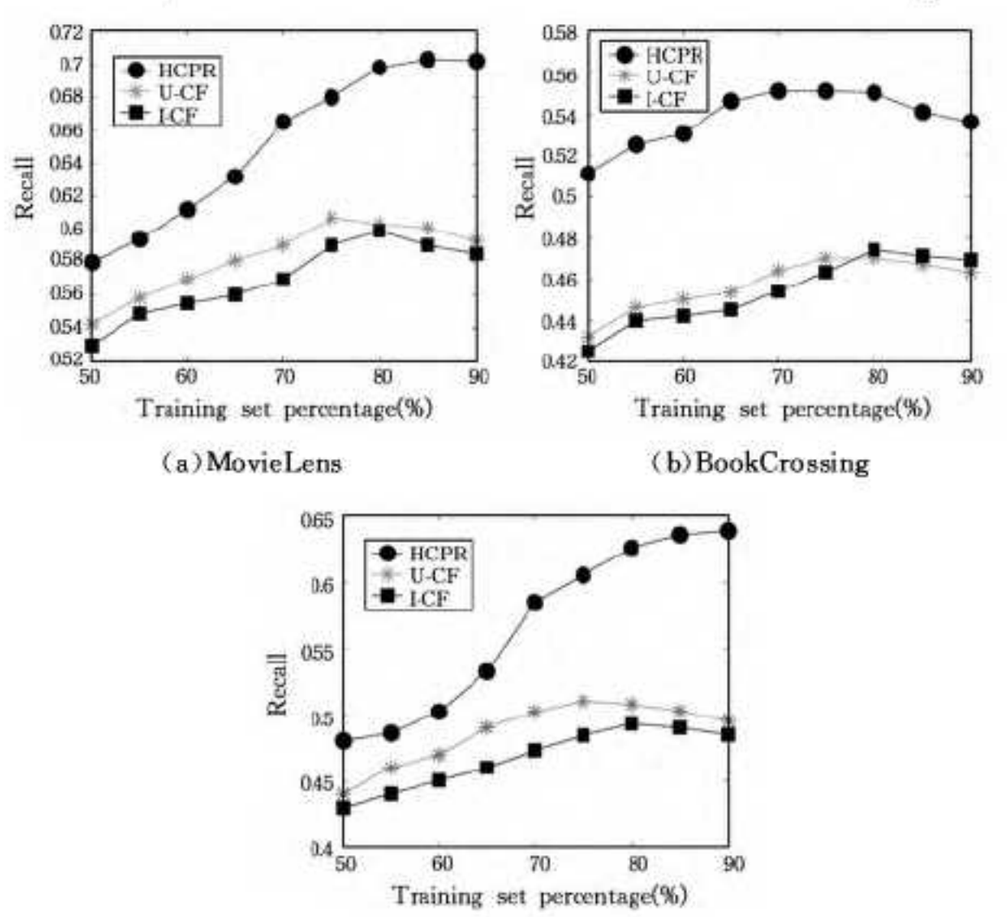


(a) MovieLens　　(b) BookCrossing

(c) RestaurantComments

图 5　不同数据集查准率比较

尽管数据集的差异造成推荐结果的查准率有一定的区别，但通过这 3 个具有差异性的数据集发现在 HCPR 的实验方案中，训练集百分比在 80% 左右时，测试集的结果能够接近最大的查准率，而 U-CF 或 I-CF 中训练集越多则其查准率越高，说明基于协同过滤的方法对已有信息的完整性要求更高。HCPR 方法主要是对社区进行构建，当社区的稳定状态构建成功以后，其他节点的加入不会导致社区发生本质性的改变，因此，对于新加入的节点，根据仅有的信息也能够将其归属于某个或多个社区的边缘节点，同样也能够为其推荐相应的项目，随着其历史行为的增加，其在社区中角色也发生相应地变化，对其推荐的精度以及多样性都会相应地增加。



(a) MovieLens　　(b) BookCrossing

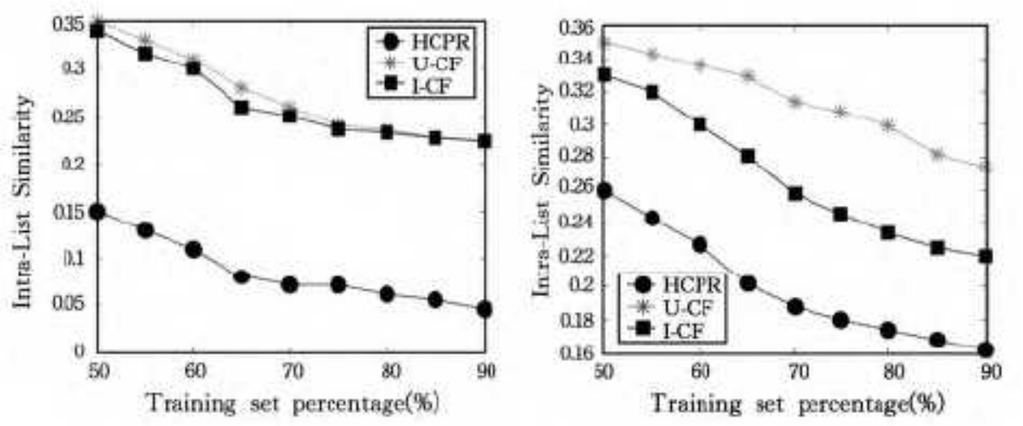
(c) RestaurantComments

图 6　不同数据集查全率比较

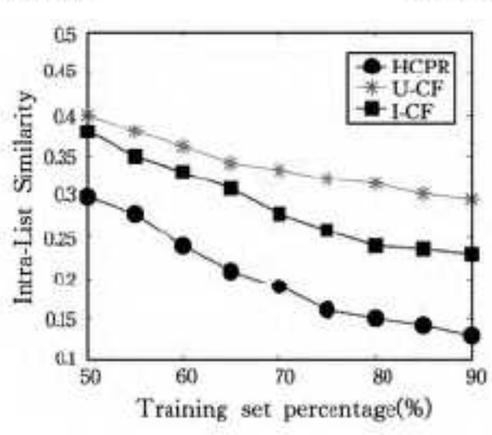
根据查全率与查准率的互逆相关性，易知查全率从另一方面反映了推荐结果是否全面。从图 6 中可以得出：U-CF、I-CF 推荐方法对数据集的信息完整性不敏感，不同数据下随着训练集的增加，查全率也相应提高，当查全率达到一定值后(在 Movielens 中达到 0.58，在 BookCrossing 和 RestaurantComments 数据集中达到 0.46 左右)，查准率的提高使得查全率具有一定的下降趋势；HCPR 算法查全率整体上较 U-

CF、I-CF 要高，但从图 6(a)、6(c)与 6(b)的对比看出算法对数据集信息的完整性较为敏感，如 MovieLens 和 Restaurant-Comments 数据集中查全率稳定增加，而在 BookCrossing 中则较为波动，这主要考虑到数据集信息的完整性对构建社区的影响较大，当信息量不足以构建稳定、具有区分性的社区时，推荐结果的全面性和准确性也得不到保证。

通常推荐系统不仅仅需要关注推荐结果的精确性，多样性也是推荐结果的一项重要指标。图 7 中是对 3 个数据集下的推荐结果的多样性结果的展示。本文提出的两层混合计算推荐能够充分考虑项目层社区内的多样性因素，在不同数据集上的结果显示，多样性较 U-CF 和 I-CF 推荐方法的效果更好，同时注意到，U-CF 和 I-CF 推荐方法的查准率与多样性具有一定的负相关性，当查准率较高时，其多样性相对较低，当保证多样性时，其查准率则偏低；HCPR 方法能够将查准率和多样性置于一个较为平衡的水平，查准率随多样性的改变波动较小，HCPR 在用户和项目层构建出了具有区分性的稳定社区，本质上是对多样性的一种处理。根据混合计算结果从不同社区推荐出来的结构的多样性指标显示出较好的效果。



(a)MovieLens　(b)BookCrossing



(c)RestaurantComments

图 7　推荐多样性比较

5.3　实验参数影响

HCPR 算法在构建社区的过程中，用户社区构建的融合相似度因子 α 以及社区划分算法的阈值因子的取值对推荐结果具有较大的影响，接下来将对这两个因子从平均绝对误差[25]角度在不同数据集中进行实验讨论，下面给出了 MAE 的计算公式：

$$MAE=\frac{1}{\|A\|}\sum_{a\in A}\frac{\sum_{j\in B}|r_{a,j}-p_{a,j}|}{\|B\|} \tag{12}$$

其中，$r_{a,j}$ 是用户 u_a 对项目 v_j 的实际评分，$p_{a,j}$ 是预测评分，A 和 B 分别表示目标用户集合和测试集。

5.3.1　融合相似度 α 参数分析

大多数协同过滤算法在计算用户属性相似度时通常仅考虑 Pearson 相似度，这就缺少了对用户自身属性的考虑。HCPR 算法中融合了两种相似度，图 8 结果显示：1)α 因子的取值影响着推荐结果，可以看出，当 $\alpha=0$(即只考虑用户_项目评分关系)与 $\alpha=1$(即只考虑用户自身属性)时 MAE 相对较小，说明用户_评分关系是推荐的主要部分；2)随着 α 值的变化，当 $0.7<\alpha<0.85$ 时，出现了最低值，此处为融合相似性的最佳权占比。针对不同的数据集，α 的取值可能不同，但是 α 的范围基本可以确定，说明了用户自身属性相似性虽然在整体推荐过程中并不是决定因素，但带来的影响可以提升推荐结果的精准度。

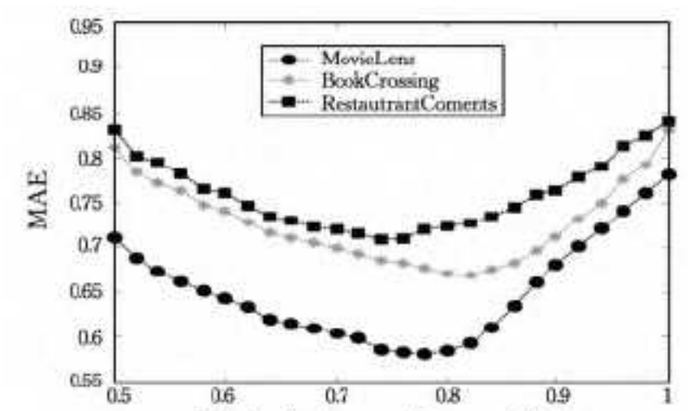


图 8　融合 α 因子影响

5.3.2　社区划分 γ 参数分析

HCPR 算法的一个核心组成部分是对两层社区的发现，文中采用了一种快速层次发现算法，它是适用于大规模网络的社区算法。γ 因子的含义是体现两个项目或用户之间相似度的阈值。在特定的数据集以及实验环境探讨 γ 的取值对社区划分的最佳状态。理论上 γ 的值越大，社区内部节点的相似性越大，社区的稳定性越高，然而，图 9 中显示随着 γ 从 0.6 增加到 0.8 后，推荐结果的 MAE 反而呈现出了增大的趋势。分析可知混合计算推荐中基于两层社区的信息量对社区的稳定性具有的影响，当 γ 的值大于一定的值以后，社区内的稳定性确实提高，但结果是越来越多的独立小社区，若 $\gamma=1$(即两者之间完全相似)，则转化为每个节点用户、项目单独为一个社区的情形，因此，推荐结果将产生很大的偏差，MAE 的值则会逐渐增大。

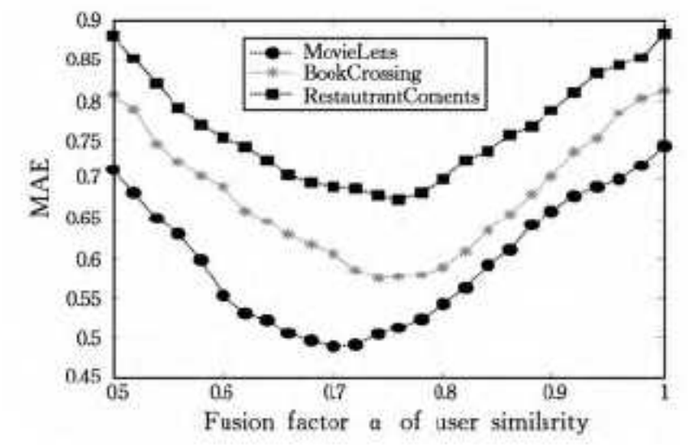


图 9　社区划分阈值 γ 影响

结束语　从项目层和用户层内在的社区结构出发，结合用户_项目层中的混杂关系，提出一种基于两层社区混合计算的推荐方法。仿真结果表明本文提出的混合计算推荐方法能够在保证推荐准确率的前提下提高推荐的多样性以及可扩展性。但对社区节点的动态性改变的更新状态速度较慢，对用户的隐式信息反馈的收集没有做分析，下一步工作将考虑动态社区发现并结合隐式信息反馈下的推荐算法进行研究。

参 考 文 献

[1] Tunkelang D. Recommendations as a conversation with the user [C]// Proc of the 5th ACM Conf on Recommender Systems. New York: ACM, 2011: 11-12

[2] X Ma, C Wang, Q Yu, X Zhou. An FPGA-Based Accelerator for Neighborhood-Based Collaborative Filtering Recommendation Algorithms[C] // IEEE International Conference on Cluster Computing. Chicago: IEEE, 2015: 494-495

[3] Manzato M, Goularte R. A multimedia recommender system based on enriched user profiles[C]// Proc of the 27th Annual ACM Symp on Applied Comuputing. New York: ACM, 2012: 975-980

[4] Zhang Shao-zhong, Chen De-ren. Hybrid Graph Model with Two Layes for Personnalized Recommendation[J]. Journal of Software, 2009(20): 123-130

[5] Soo-Cheo K, Jung-Wan K, Jung-Sik C. Collaborative filtering recommender system based on social network[J]. IT Convergence and Services, 2011, 7(2): 503-510

[6] Zhao Z L, Wang C D, Wan Y Y, et al. Pipeline Item-Based Collaborative Filtering Based on MapReduce[C]// IEEE Fifth International Conference on Big Data & Cloud Computing. Dalian: 2015: 9-14

[7] Pavlov D, Pennock D. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains[C]// Proc of the 16th Annual Conf on Neural Information Proces-sing Systems. 2002

[8] Getoor L, Sahami M. Using probabilistic relational models for collaborative filtering[C]// Proc of the Workshop Web Usage Analysis and User Profiling under KDD'99. San Diego. 1999

[9] Unger L H. Foster D P. Clustering methods for collaborative filtering[C]// Proc of the Workshop on Recommendation Systems. Menlo Park: AAAI Press, 1998: 112-125

[10] Chien Y H, George E I. A Bayesian model for collaborative filtering[C]// Proc of the 7th Int'l Workshop on Artificial Intelligence and Statistics. San Francisco: Morgan Kanfmarm, 1999

[11] Massa P, Avesani P. Trust-aware collaborative filtering for recommender systems[C]// Proc of Fourth International Conference on Trust Management. Pisa, Italy, 2006

[12] Mark Newman. Fast algorithm for detecting community structure in networks[J]. Physical Review E, 69, 066133, Jun. 2004

[13] Du Jing-fei, Lai Jiang-yang, Shi Chuan. Multi-objective Optimization for Overlapping Community Detection[J]. Advanced Data Mining and Applications, 2013: 489-500

[14] Newman M, Leicht E. Mixture models and exploratory analysis in networks. [J]. Proc. Natl. Acad. Sci, 2007, 104(23): 9564-9569

[15] Yu Le, Wu Bin, Wang Bai. LBLP: Link-Clustering-Based Approach for Overlapping Community Detection[J]. Tsinghua Science and Technology, 2013, 18(4): 387-397

[16] Martin Rosvall and Carl Bergstrom. Maps of random walks on complex networks reveal community structure[J]. Proc. Natl. Acad. Sci., 2008, 105(4): 1118-1123

[17] Palla G, Derenyi I, Farkas I, et al. Uncovering the overlapping community structure of complex networks in nature and society [J]. Nature, 2005, 435: 814-818

[18] Shen Hua-wei, Cheng Xue-qi, Cai Kai, et al. Detect overlapping and hierarchical community structure[J]. Physica A, 2008, 388 (8): 1706-1717

[19] Zhu Xiao-jin, Ghahramani Zou-bin. Learning from Labeled and Unlabeled Data with Label Propagation[R]. Techinicalreport, CMU CALD tech report CMU-CALD-02, 2002

[20] Gregory S. Finding overlapping communities in networks by label propagation[J]. New J. Phys., 2010, 12: 103018

[21] Gower J C. A General Coefficient of Similarity and Some of Its Properties[J]. International Biometric Society, 1971, 27(4): 857-871

[22] Blondel V, Guillaue J L. Renaud Lambiotte and Etienne Lefebvre. Fast unfolding communities in large networks[J]. Journal of Statistical: Mechanics Theory & Experiment, 2008, 30(2): 155-168

[23] Sarwar B, Karypis G, Konstan J, Riedl J. Analysis of recommendation algorithm for e-commerce[C]// Proc of the 2nd ACM Conference on Electronic commerce. New York: 2000: 158-167

[24] Ziegler C, McNee S, Konstan J. Improving recommendation lists through topic diversification[C]// Proc of the 14th International Conference on World Wide Web. New York: 2005: 22-32

[25] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithm[C]// Procedings of the 10th International Conference on World Wide Web. New York: ACM Press, 2001: 285-295

(上接第 421 页)

[7] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[C]// Proceedings of Workshop at ICLR. 2013

[8] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]// Proceedings of NIPS. 2013

[9] Mikolov T, Yih W, Zweig G. Linguistic regularities in continuous space word representations[C]// Proceedings of NAACL HLT. 2013

[10] Joachims T. Training linear SVMs in linear time [C]// Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD). 2006

[11] Joachims T. A support vector method for multivariate performance measures[C]// Proceedings of the International Conference on Machine Learning(ICML). 2005

[12] Liu B, Zhang L. A survey of opinion mining and sentiment analysis[J]. Synthesis Lectures on Human Language Technologies, 2010, 2: 459-526

[13] Tang H, Tan S, Cheng X. A survey on sentiment detection of reviews[J]. Expert Systems with Applications, 2009: 10760-10773

[14] Joachims T, Yu C. Sparse kernel SVMs via Cutting-Plane training[M]// Machine Learning and Knowledge Discovery in Dtabases. Springer. Berlin Heidelbergy. 2009

[15] Tang H, Tan S, Cheng X. A survey on sentiment detection of reviews[J]. Expert Systems with Applications, 2009, 36: 10760-10773

[16] Zhai Z, Liu B, Xu H, et al. Grouping product features using semi-supervised learning with soft-constraints[C]// Proceedings of the 23rd International Conference on Computational Linguistics (COLING). Beijing: ACL, 2010: 1272-1280

[17] Zhai Z, Liu B, Xu H, et al. Clustering product features for opinion mining[C]// Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM). Hong Kong: ACM, 2011: 347-354

[18] Jose C. A Fast On-line Algorithm for PCA and Its Convergence Characteristics [J]. IEEE Transactions on Neural Network, 2000, 4(2): 299-307