融合社交网络信息的协同过滤推荐算法*

郭兰杰 梁吉业 赵兴旺

[1](山西大学 计算机与信息技术学院 太原 030006)
[2](山西大学 计算智能与中文信息处理教育部重点实验室 太原 030006)

摘 要 在推荐系统中,协同过滤推荐算法往往面临数据集的高度稀疏性和推荐精度有限的问题. 为了解决上述问题,在基于物品的协同过滤推荐框架下,分别在物品相似度的计算和用户对物品的评分预测阶段,利用社交网络中朋友关系信息选择性地填充评分矩阵中的缺失值,最大化利用评分矩阵中的已有信息,提出融合社交网络信息的协同过滤推荐算法. 最后,在 Epinions 数据集上的实验表明,文中算法在一定程度上缓解数据稀疏性问题,同时在评分误差和分类准确率两个指标上优于其它协同过滤算法.

关键词 协同过滤,社交网络,缺失值填充,数据稀疏性
中图法分类号 TP 391 **DOI** 10.16451/j.cnki.issn1003-6059.201603010
引用格式 郭兰杰,梁吉业,赵兴旺. 融合社交网络信息的协同过滤推荐算法. 模式识别与人工智能,2016,29(3):281 – 288.

Collaborative Filtering Recommendation Algorithm Incorporating Social Network Information

GUO Lanjie, LIANG Jiye, ZHAO Xingwang
(*School of Computer and Information Technology, Shanxi University, Taiyuan* 030006)
(*Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan* 030006)

ABSTRACT

To solve the problems of high data sparsity and limited recommendation precision of collaborative filtering recommendation algorithms, a collaborative filtering algorithm incorporating social network information is proposed under the framework of item-based collaborative filtering recommendation. In item similarity calculation period and user rating prediction period, social network information is utilized to fill missing values in rating matrix selectively and thus the existing rating information is utilized as much as possible. Finally, experiment is conducted on Epinions dataset. Results show that the proposed algorithm alleviates the data sparsity problem and outperforms other collaborative filtering algorithms on rating error and precision.

* 国家自然科学基金项目(No. 61573229,61432011,U1435212)、山西省科技基础条件平台建设项目(No. 2012091002 – 0101)、山西省科技攻关计划项目(No. 20110321027 – 01)资助
Supported by National Natural Science Foundation of China (No. 61573229,61432011,U1435212), Construction Project of Science and Technology Basic Condition Platform of Shanxi Province (No. 2012091002 – 0101), Key Technology R&D Program of Shanxi Province (No. 20110321027 – 01)
收稿日期: 2015 – 05 – 19; **修回日期:** 2015 – 10 – 13; **录用日期:** 2015 – 10 – 26
Manuscript received May 19, 2015; revised October 13, 2015; accepted October 26, 2015

Key Words Collaborative Filtering, Social Network, Missing Value Filling, Data Sparsity

Citation GUO L J, LIANG J Y, ZHAO X W. Collaborative Filtering Recommendation Algorithm Incorporating Social Network Information. Pattern Recognition and Artificial Intelligence, 2016, 29(3): 281-288.

随着信息技术和互联网的飞速发展,人们开始面临严重的信息过载问题,这使推荐系统在信息获取中成为一种重要技术[1]. 推荐系统利用知识发现技术帮助用户从海量物品中发现自己感兴趣的物品,进而进行个性化推荐.

在已有的推荐技术中,协同过滤[2]是发展较快也较为流行的一种方法. 然而在实际应用领域中,评分信息数据集中用户和物品的维度规模逐渐增大,数据的稀疏程度不断增加,导致传统的协同过滤推荐算法的准确性随着数据集稀疏程度的增加而降低[3]. 为了解决这一问题,一些学者已从不同角度开展一些探索性的研究[4-7]. 冷亚军等[4]针对稀疏评分导致的最近邻搜寻不准确的问题,提出两阶段最近邻选择算法,首先找到用户近邻倾向性高的集合,然后计算它们之间的等价关系相似性,得到最终的最近邻集合,有效提高近邻搜寻的准确性. Wang 等[5]融合基于用户的协同过滤和基于物品的协同过滤的思想,并利用相似用户对相似物品的评分进行预测,在一定程度上缓解单一协同过滤方法面临的数据稀疏问题. Liang 等[6]利用联合聚类方法,聚类原始评分矩阵,并融合类别相似性和传统评分相似性,有效缓解数据稀疏性带来的相似性计算不准确的问题. Koren 等[7]将原始稀疏评分矩阵分解为低维稠密的潜因子矩阵,解决协同过滤算法对数据稀疏性敏感的问题. 上述方法虽在一定程度上缓解评分矩阵的稀疏性问题,但这些研究仅依赖极度稀疏的用户评分信息进行处理,算法性能缺乏一定的可靠性.

在传统的协同过滤推荐算法中,用户是否喜欢目标物品只受个人偏好的影响,并假设用户之间相互独立. 然而在现实生活中,用户的决定是由多方面因素共同作用而产生,除了自身原因之外,还会受到身边朋友的影响. 近年随着 Web 2.0 技术的快速发展和日趋成熟,已产生许多诸如用户社交关系、用户评论转发等社交网络. 社交网络的出现为推荐技术的研究提供新的途径[8]. 已有研究表明,在推荐系统评分预测过程中,如果能合理填充评分矩阵中的缺失值,可提升预测精度[9]. 同时,Tang 等[10]也指出,用户的社交网络信息对于提升推荐系统的性能有着重要的作用.

基于上述考虑,为了克服由于评分信息稀疏性导致推荐精度较低的问题,在基于物品的协同过滤推荐框架下,本文提出融合社交网络信息的协同过滤推荐算法. 该算法分别在物品相似度的计算和用户对物品的评分预测阶段,利用社交网络中朋友关系信息填充评分矩阵中的部分缺失值,使评分矩阵中的已有信息利用达到最大化. 本文算法可有效缓解传统协同过滤算法在相似度计算和预测评分阶段由于数据稀疏性引起的邻居数量和可靠性不足而导致推荐能力有限的问题. 最后,在真实数据集 Epinions 上与已有的 4 种推荐算法进行的实验分析表明,本文算法在评分误差和分类准确率两个指标上优于其它算法.

1 相关知识

协同过滤推荐算法由 Goldberg 等[2]提出,现已成为推荐系统中发展最快、应用最广的推荐算法之一. 该算法主要利用用户对物品的评分矩阵信息 $\boldsymbol{R}_{m\times n}$,如表 1 所示,其中行代表用户,列代表物品,矩阵中元素 $r_{u,i}$ 表示用户 u 对物品 i 的评分,通常采用 1 ~ 5 分表示,空值代表无评分. 在大部分应用系统中评分数据稀疏度都在 99% 以上,这成为限制传统协同过滤算法性能的重要原因之一.

表 1 原始用户-物品评分矩阵

Table1 Original user-item rating matrix

	i_1	i_2	i_3	i_4	i_5	…	i_n
u_1	$r_{1,1}$			$r_{1,4}$			
u_2		$r_{2,2}$	$r_{2,3}$		$r_{2,5}$		$r_{2,n}$
u_3	$r_{3,1}$	$r_{3,2}$		$r_{3,4}$			
⋮							
u_m			$r_{m,3}$	$r_{m,4}$			

传统协同过滤算法通常分为两类: 基于模型的协同过滤算法和基于内存的协同过滤算法. 基于模型的协同过滤算法采用机器学习、数据挖掘中的方法进行建模,如聚类模型[11]、回归模型[12]、潜在语义模型[13]、贝叶斯模型[14]等,具有良好的可扩展性. 基于内存的协同过滤算法利用整个用户-物品

评分矩阵进行推荐. 一般分为基于用户的协同过滤算法和基于物品的协同过滤算法[11]. 从基于物品协同过滤的角度来说,算法过程主要分为两个阶段.

1) 邻居物品的形成. 首先基于评分信息,计算物品 i 与其它物品的相似度,然后从大到小排序,选择前 k 个物品作为物品 i 的邻居. 常用的相似度度量方法为皮尔逊相似度[8],定义如下:

$$sim(i,j) = \frac{\sum_{u\in U(i)\cap U(j)}(r_{u,i}-\bar{r}_i)\cdot(r_{u,j}-\bar{r}_j)}{\sqrt{\sum_{u\in U(i)\cap U(j)}(r_{u,i}-\bar{r}_i)^2}\cdot\sqrt{\sum_{u\in U(i)\cap U(j)}(r_{u,j}-\bar{r}_j)^2}}, \quad (1)$$

其中,$U(i)$、$U(j)$ 分别表示对物品 i 和物品 j 评分过的用户集合,$U(i)\cap U(j)$ 表示同时对物品 i 和物品 j 评分过的用户集合,$r_{u,i}$ 表示用户 u 对物品 i 的评分,$\bar{r}_i$ 和 $\bar{r}_j$ 分别表示物品 i 和物品 j 所有评分的均值.

2) 预测评分. 在找到的物品 i 的邻居集合的基础上,通过目标用户 u 对物品 i 邻居的评分值进行加权求和,得到最终的预测结果. 预测公式如下[15]:

$$P_{u,i} = \bar{r}_i + \frac{\sum_{j\in S(i)} sim(i,j)\cdot(r_{u,j}-\bar{r}_j)}{\sum_{j\in S(i)} sim(i,j)}, \quad (2)$$

其中,$S(i)$ 表示物品 i 的 k 近邻集合,$Sim(i,j)$ 表示式(1)定义的物品与物品之间的皮尔逊相似度,$\bar{r}_i$ 和 $\bar{r}_j$ 分别表示物品 i 和物品 j 所有评分的均值,$r_{u,j}$ 表示用户 u 对物品 j 的实际评分.

近年出现一些融合社交网络信息进行协同过滤推荐的方法,弥补传统协同过滤推荐算法在评分数据稀疏性方面存在的缺陷. 例如,Qian 等[16]将用户个人兴趣、朋友之间的影响力等 3 种因素融合到概率矩阵分解模型中,提高协同算法的推荐精度. Liu 等[17]利用用户社交关系信息改进传统协同过滤算法寻找最近邻的过程,将评分相似用户和朋友共同作为用户邻居,缓解数据稀疏引起的邻居数量不足的问题. Yang 等[18]利用社交网络数据,将用户圈与物品类别对应,针对特定类别的物品,根据对应用户圈的评分信息进行预测,避免对不相关社交关系的误用.

上述研究在一定程度上缓解数据稀疏性对推荐准确性的影响,但对社交网络信息仅在计算用户邻居的过程中得到应用,未能与评分信息进行深度融合,进而使其在整个推荐过程中充分发挥作用,同时对已有评分信息的利用还很不充分. 为了解决这一问题,本文提出社交网络信息与评分矩阵深度融合的协同过滤推荐算法.

2　融合社交网络信息的协同过滤推荐算法

通过上述分析,在基于物品的协同过滤推荐算法的框架下,将社交网络信息融入推荐过程中,通过对评分矩阵中的缺失值进行有选择地填充,缓解数据稀疏性的问题,达到提高协同过滤算法推荐准确度的目的. 下面将研究如下两个问题: 1) 选择哪些缺失值进行预测填充; 2) 如何利用社交网络信息对评分矩阵中的缺失值进行填充.

2.1　两阶段缺失值填充策略

本节主要研究第一个问题,即选择哪些缺失值位置进行预测填充. 为此,本文提出两阶段缺失值填充策略.

1) 第一阶段. 从式(1)可看到,物品间相似度的计算依赖于共同评价过它们的用户. 如果用户量较大,计算结果较准确; 如果用户量太小,计算结果就缺乏可信度. 而由于数据稀疏性,满足要求的用户很少,导致相似度计算不准确,同时物品获得的评分未得到充分利用. 因此,针对物品单方面缺失的情况,如果能给予恰当的填充值,就可最大限度利用物品已有的评分数据,同时避免引入过多的噪声数据,提高相似度计算的可靠性.

例如,表 1 为推荐系统原始的用户–物品评分矩阵,假如要计算物品 i_1 和 i_2 的相似度,在传统协同过滤中,只能利用 u_3 对它们评分的差异性. 虽然 u_1 对 i_1 进行评分,但因为未对 i_2 进行评分,因此无法利用这部分信息. 同理,u_2 对 i_2 的评分信息面临同样的问题. 在这种情形下,本文利用用户的社交关系信息,填充 u_1 对 i_2 的评分及 u_2 对 i_1 的评分,达到对已有评分信息的充分利用. 需注意的是,用户 u_m 对物品 i_1、i_2 的评分不进行填充,因为该用户对两个物品都未有评分,对这部分填充会加入更多的噪声,在可靠性上存在较大不足.

2) 第二阶段. 从式(2)可知,在评分预测过程中,传统基于物品的协同过滤依赖于用户对相似物品的评分. 但在实际应用中,由于评分数据的稀疏性,极有可能用户对相似物品无评分,而评分的不是相似物品. 因此,针对该情况,如果能对相似物品进行恰当的缺失值填充,就能充分利用相似度计算的结果,给予更准确的预测.

例如,在表 1 中,假如要预测用户 u_3 对物品 i_5 的评分,而已计算出与物品 i_5 最相似的是物品 i_3,但由于 u_3 未对 i_3 进行评分,在传统方法中将会利用相似度较小的物品计算. 针对这种情况,本文利用用户的社交关系信息,填充 u_3 对 i_3 的评分值,以期提高算法的预测精度.

2.2 基于用户社交关系的缺失值预测算法

在确定填充位置后,现主要解决如何利用社交网络信息进行填充的问题. 给定用户社交关系网络,记为 $G=(U,E)$,其中

$$U=\{u_1,u_2,\cdots,u_m\}$$

表示用户节点集合,E 表示用户之间的关系集合,如果 $(u,v)\in E$,表示用户 u 与用户 v 之间有好友关系,相反,表示不存在好友关系. 用户 u 的直接好友集合可表示为 $N(u)$,即如果 $v\in N(u)$,则 $(u,v)\in E$.

不同于协同过滤中相似度的计算方法,本文使用熟悉度度量好友之间的亲近程度,基于如下假设:共同好友越多,用户间越熟悉,而越熟悉的用户越有可能有相似的兴趣爱好.

2.2.1 熟悉度

使用常用的 Salton 指标和大度节点不利指标(Hub Depressed Index, HDI)[19] 度量用户间的社交关系强度,以此评价不同熟悉度定义下对最终推荐性能的影响.

Salton 指标定义如下:

$$ST_{u,v}=\frac{|N(u)\cap N(v)|}{\sqrt{k(u)k(v)}},\tag{3}$$

其中,$ST_{u,v}$ 表示用户 u 和用户 v 的熟悉程度;$N(u)$ 和 $N(v)$ 分别表示用户 u 和用户 v 的好友集合;$k(u)$ 和 $k(v)$ 分别表示用户 u 和用户 v 的度,即好友数量.

HDI 定义如下:

$$HDI_{u,v}=\frac{|N(u)\cap N(v)|}{\max(k(u),k(v))},$$

其中,$HDI_{u,v}$ 表示用户 u 和用户 v 的熟悉程度,其余符号表示含义与式(3)相同.

2.2.2 填充缺失值

使用类似基于用户协同过滤的思想,通过融合朋友偏好,代表用户偏好. 计算公式如下:

$$P_{u,i}=\bar{r}_u+\frac{\sum_{v\in F(u)}f_{u,v}(r_{v,i}-\bar{r}_v)}{\sum_{v\in F(u)}f_{u,v}},$$

其中,$P_{u,i}$ 表示用户 u 对物品 i 的填充评分,$\bar{r}_u$ 和 $\bar{r}_v$ 分别表示用户 u 和用户 v 对物品所有评分的平均值,$F(u)$ 表示用户 u 的好友中对物品 i 评分过的用户集合,$f_{u,v}$ 表示用户 u 和用户 v 的熟悉程度,本文使用流行的 Salton 方法进行计算,$r_{v,i}$ 表示用户 v 对物品 i 的评分值.

需要注意的是,有时候用户的朋友也未对目标物品进行过评分. 在这种情况下,使用物品均值和用户均值进行填充,以保证已有的评分数据得到充分利用,此时计算公式如下:

$$P_{u,i}=\frac{\bar{r}_i+\bar{r}_u}{2},$$

其中 $\bar{r}_i$ 表示目标物品 i 的评分均值.

综上所述,缺失值填充的计算公式如下:

$$P_{u,i}=\begin{cases}\bar{r}_u+\dfrac{\sum_{v\in F(u)}f_{u,v}(r_{v,i}-\bar{r}_v)}{\sum_{v\in F(u)}f_{u,v}}, & F(u)\text{ 不为空}\\ \dfrac{\bar{r}_i+\bar{r}_u}{2}, & \text{其它}\end{cases}\tag{4}$$

2.3 基于缺失值填充的协同过滤推荐算法

基于 2.2 节提出的用户社交关系的缺失值填充算法,通过缓解评分数据的稀疏性,改善传统协同过滤推荐算法推荐精度较低的问题. 算法步骤如下.

算法 基于缺失值填充的协同过滤推荐算法

输入 用户-物品评分矩阵 $\boldsymbol{R}_{m\times n}$,用户社交关系信息 $G=(U,E)$

输出 目标用户 u 生成长度为 L 的推荐列表 $L(u)$

step 1 在评分矩阵 $\boldsymbol{R}_{m\times n}$ 找到用户 u 的未评分物品集合 I',即评分矩阵中第 u 行空值对应的物品.

step 2 计算物品 i 与所有物品集合 $I(i\notin I)$ 中物品之间的皮尔逊相似度. 首先利用式(4)针对评分向量单方面存在缺失值的情况进行填充,然后利用式(1)进行相似度的计算.

step 3 选择相似度最大的 k 个物品作为目标物品 i 的邻居,并根据式(4)对目标用户未对齐评分的物品进行缺失值填充,然后采用式(2)计算用户 u 对物品 i 的预测评分.

step 4 循环 step 2 ~ step 3,计算用户 u 对未评分物品集合 I' 中每个物品的预测评分,然后按照评分值大小降序排列,选择前 L 个物品加入到用户 u 的推荐列表 $L(u)$ 中.

本文算法步骤的关键是在传统的基于物品协同过滤推荐算法的两个阶段(物品间相似度计算和预测评分)中,通过引入用户社交网络信息,缓解评分数据稀疏性带来的问题. 在 step 2 中,传统协同过滤算法仅利用稀疏的评分数据计算物品间相似度,往往会导致相似度计算的不可靠. 本文算法在物品相似度计算之前,对单方面评分缺失的情况利用2.2

节中提出的方法进行选择性填充,提高物品相似度计算的可靠性. 同时,在 step 3 中,传统协同过滤算法在找到最近邻后,只能利用用户对其有评分的部分,而数据稀疏性导致这部分可用评分极少,导致预测结果精度有限. 而本文算法在计算之前,采用 2.2 节中提出的填充方法,针对目标用户对应的缺失评分,利用用户好友的评分信息表示,使可利用评分增加,提高预测精度.

3　实验及结果分析

首先从推荐结果的有效性方面对比本文算法与已有推荐算法,其次通过实验分析本文算法一些参数的设置对性能的影响.

实验环境如下: 4 GB 内存, Intel (R) Core (TM) i7 - 2600 处理器,3.4 GHz,Windows 7 操作系统.

3.1　实验数据集

实验应用通过用户评价网站 Epinions.com 采集的 Epinions 数据集. 该数据集的信息表示用户给汽车、图书、电影等物品的评分,通常用数值 1 ~ 5 表示. 同时,用户可表达是否信任系统中其他用户,如果信任,在数据集中标记为 1,否则标记为 0.

该数据集包含 40 163 位用户对 139 738 种物品的664 824 个评分数据,其中每个用户至少评价过一个物品. 该评分数据的稀疏度为 99.99%,其它统计信息见表 2. 此外,该数据集还包含 442 979 条用户之间的信任关系,信任数据的稀疏度为 99.97%.

表 2　Epinions 评分数据统计

Table 2　Statistics of user-item rating matrix of Epinions

统计项	最小评分数量	最大评分数量	平均评分数量
用户	1	1022	16.55
物品	1	2018	4.76

3.2　评价指标

采用如下两类指标评价推荐结果的优劣.

第一类指标度量评分预测误差率,通过对比真实评分和预测评分之间的差距评价推荐算法预测评分的准确度. 本文使用平均绝对误差(Mean Absolute Error, MAE) 和均方根误差(Root Mean Squared Error, RMSE) [9],值越小表示推荐效果越好.

MAE 定义为

$$MAE = \frac{1}{N}\sum_{i,j} |r_{i,j} - r'_{i,j}|,$$

其中,$r_{i,j}$ 表示用户 i 对物品 j 的实际评分,$r'_{i,j}$ 表示用户 i 对物品 j 的预测评分,N 表示测试集中包含的评分数量.

RMSE 定义为

$$RMSE = \sqrt{\frac{1}{N}\sum_{i,j} (r_{i,j} - r'_{i,j})^2},$$

其中各符号的意义与 MAE 的相同. MAE 和 $RMSE$ 的值越小,表示预测误差越小,预测精度越高.

第二类指标度量最终推荐列表的分类准确性. 由于本文实验数据集的评分范围为 1 ~ 5 分,因此本文认为在测试集中大于 3 的评分,表示用户喜欢相应物品,如果能将这部分物品排到推荐列表前端,则用户可能对推荐列表更满意. 使用平均准确率(Mean Average Precision, MAP) 进行度量,值越大表示生成的推荐列表质量越高,具体定义如下:

$$MAP = \frac{1}{|U|}\sum_{j=1}^{|U|}\frac{1}{m_j}\sum_{k=1}^{m_j} Precision(r_{j,k}),$$

其中,$|U|$ 表示总共推荐的用户数量,m_j 表示用户在测试集中喜欢的物品数量,$Precision(r_{j,k})$ 表示用户 j 喜欢的物品 k 在推荐列表中位置的倒数,即用户真正喜欢的物品排序越靠前,准确率越高. 整个系统的平均准确率为所有用户准确率的平均值.

考虑到用户未评分物品无法验证准确性,因此最终推荐列表中只对测试集中存在的物品进行排序,以此判断算法是否将用户喜欢的物品排到列表前面.

3.3　5 种算法实验结果对比分析

为了验证用户的社交关系在推荐过程中的作用和对推荐性能的影响,在 MAE、RMSE 和 MAP 这 3 个指标上将本文算法与如下 4 种算法对比分析.

1) 传统的基于用户的协同过滤算法(User-Based Collaborative Filtering, UCF) [8]. 只利用评分信息,通过度量用户间皮尔逊相似度,进而基于用户最近邻进行预测推荐.

2) 传统的基于物品的协同过滤算法(Item-Based Collaborative Filtering, ICF) [20]. 只利用评分信息,通过度量物品间皮尔逊相似度,进而基于物品最近邻进行预测推荐.

3) Liu 等[17]提出的融合用户朋友和相似用户的协同过滤算法(Combine Neighbors and Friends Based Collaborative Filtering, CNCF). 利用评分信息和社交关系信息,由社交关系强度最大的用户和评分相似度最大的用户共同构成传统最近邻,然后采用与 UCF 相同的策略进行预测推荐.

4) 基于用户朋友的协同过滤算法(Friend-Based Collaborative Filtering, FCF) [17]. 由社交关

系强度最大的用户构成最近邻，然后采用与 UCF 相同的策略进行预测推荐.

本文算法选择使用 Salton 指标度量用户间社交关系强度，选择使用皮尔逊相似度度量计算用户间或物品间的评分相似度. 实验采用五折交叉验证方法，最终结果为 5 次实验结果的平均值.

已有研究表明，协同过滤推荐算法在用户或物品邻居数为 30 时推荐效果较优[19]. 因此，5 种算法都将用户或物品邻居数设置为 30 进行对比分析，实验结果如表 3 所示，表中，↓ 表示该指标值越小越好，↑ 表示该指标值越大越好.

表 3 在 30 近邻时 5 种推荐算法的实验结果

Table 3 Results of the 5 recommendation algorithms with 30 neighborhoods

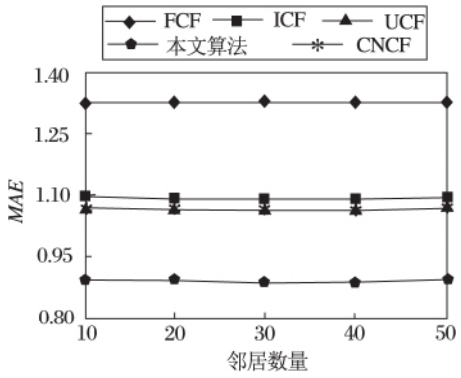
指标	UCF	ICF	CNCF	FCF	本文算法
MAE ↓	1.0646	1.0917	1.0633	1.3257	**0.8865**
RMSE ↓	1.3286	1.3500	1.3262	1.5170	**1.1604**
MAP ↑	0.9209	0.9205	0.9216	0.8983	**0.9406**

另外，5 种算法在协同过滤过程中度量用户间、物品间的相似度时均采用皮尔逊相关系数. 特别地，本文算法在基于社交网络对缺失值进行选择性填充时，朋友数量设置为 30，并选择 Salton 指标度量用户间社交关系强度.

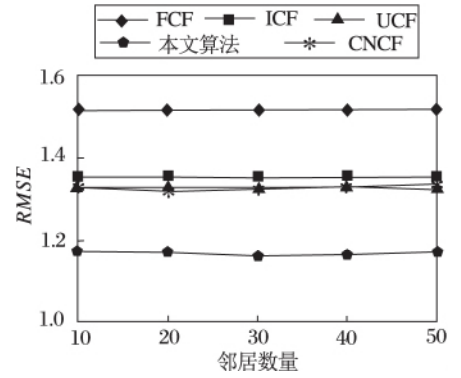
由表 3 可知，CNCF 在 3 个指标上都优于传统的协同过滤算法，说明用户社交网络关系信息在推荐过程中发挥重要作用，弥补传统协同过滤算法可能存在的邻居不足问题. FCF 性能表现最差，原因可能在于并不是用户与其所有朋友都有相同的兴趣偏好，不能简单地将其视为传统意义上的邻居. 同时可看出，本文算法在 3 个指标上都取得最好效果，验证本文算法基于用户社交关系信息进行评分缺失值填充的有效性，较大幅度提高传统协同过滤算法的推荐准确度.

在传统的协同过滤算法中，用户或物品的邻居数量会对推荐结果产生一定影响. 图 1 给出用户或物品邻居数量的变化对 5 种推荐算法的推荐精度的影响.

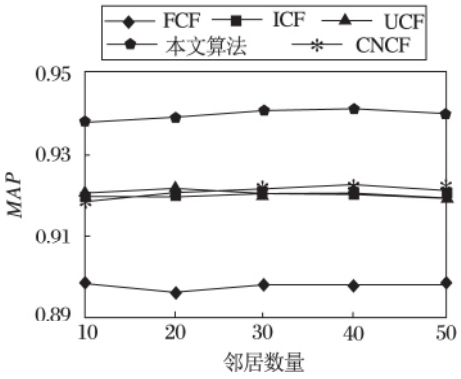
从图 1 可看出，随着邻居数量的增加，5 种算法的 *MAE* 和 *RMSE* 都呈现出先减后增的趋势，而 *MAP* 呈现出先增后减的趋势. 这是由于当邻居数量太少时，因为只参考少数人的意见，导致预测评分受个人因素影响较大. 而当邻居数量过多时，又有可能加入质量较低用户的意见，对正确用户做出的预测产生错误诱导. 另外，相比其余 4 种算法，本文算法在 3 个指标上都有更好结果，再次验证本文算法推荐结果的有效性.



(a) *MAE*



(b) *RMSE*



(c) *MAP*

图 1 目标物品邻居数量对 5 种算法推荐性能的影响

Fig. 1 Influence of neighborhood number of target item on the recommendation performance of 5 algorithms

为了更进一步研究本文算法，以下部分将对影响本文算法性能的因素进行更深入的实验分析，主

要包括缺失值填充过程中朋友数量及朋友之间社交关系强度的度量指标的选择.

3.4　缺失值填充时社交关系强度计算方法对比

在本文算法中,基于用户社交关系缺失值填充是极其重要的一步. 而用户之间社交关系的强度决定用户朋友对于填充值的贡献程度,因此如何准确度量用户间社交关系强度至关重要. 实验中,对比 Salton 方法与 HDI 对推荐结果的影响. 为了验证基于网络结构刻画用户间关系强度的优势,还对比传统的基于用户评分向量的皮尔逊相似度. 最后,为了说明不同社交关系强度方法的有效性,对比所有社交关系强度都为 1 的情形.

表4 给出3 个指标对推荐性能的影响情况. 相比 HDI,Salton 可在 3 个评价指标上都取得更优结果. 同时发现,相比传统的皮尔逊方法,基于网络结构的 HDI 和 Salton 在3 个指标上都更优. 这主要是因为基于网络结果的方法在考虑共同评分部分外,还考虑用户节点度的影响,因此对用户的相似度刻画更精细准确. 而上述任何一种度量方法都要比社交关系强度区分的常量为 1 的方法更好. 可见,使用何种方法计算用户间的社交关系强度对于推荐性能影响非常重要.

表 4　3 个指标对推荐性能的影响

Table 4　Influence of 3 evaluation indexes on recommendation performance

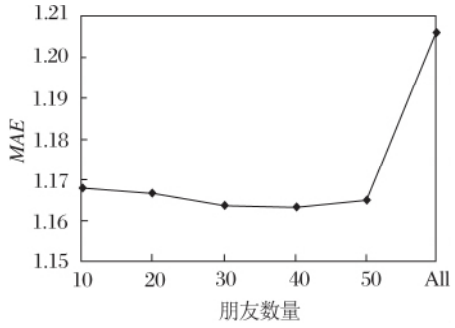
指标	HDI	Salton	Pearson	*Const* = 1
MAE ↓	0.8898	**0.8865**	0.8910	0.9175
RMSE ↓	1.1652	**1.1604**	1.1660	1.2037
MAP ↑	0.9404	**0.9406**	0.9387	0.9387

3.5　缺失值填充时朋友数量对推荐结果的影响

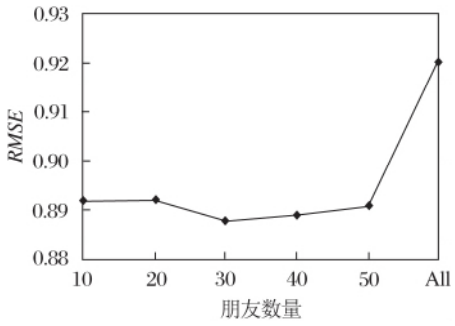
在本文算法中,基于用户社交关系进行缺失值填充是极其重要的一步,而参与填充的朋友数量对推荐精度有一定影响. 如果朋友数量过少,可能缺失值的预测集中在某几个人的意见,从而产生较大偏差. 而朋友数量过多,可能会带入一些噪音,反而降低预测精度.

图 2 给出在预测评分时目标物品邻居数量为 30,好友熟悉度度量采用 Salton 指标的情况下,缺失值填充时好友数量对最终推荐精度的影响,图中,All 表示目标用户的所有朋友.

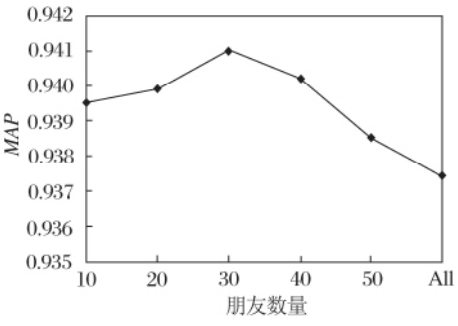
从图 2 可见,不同的朋友数量对于实验结果产生一定程度的影响. 当朋友数量由少变多时,各个指标都趋于更优. 而当朋友数量超过 30 时,随着朋友数量的继续增加,各个指标逐渐变差,而当利用用户所有的朋友关系时,推荐性能变得很差. 可见,恰当选择朋友数量,可提高本文算法性能. 由实验分析可得,本文选择 30 个朋友对用户缺失值进行预测性能最佳.



(a) *MAE*



(b) *RMSE*



(c) *MAP*

图 2　缺失值填充时朋友数量对本文算法推荐性能的影响

Fig. 2　Influence of friend number on recommendation performance of the proposed algorithm after filling missing values

4　结 束 语

本文提出基于社交网络对评分矩阵进行缺失值

填充的协同推荐算法,在传统的基于物品的协同过滤推荐算法中,通过利用丰富的用户社交网络信息,对稀疏的原始评分数据进行两阶段缺失值的填充,缓解制约传统协同过滤算法推荐性能的数据稀疏问题. 实验表明,本文算法有效提升传统推荐算法的推荐准确度,且比已有的利用社交关系信息的推荐算法更有效.

本文只利用用户社交关系信息,忽略用户及物品的一些内容属性信息、评分时间信息等. 在未来研究中,将在推荐过程中融入更多的社交上下文信息,以便进一步改善传统协同过滤算法的推荐性能.

参 考 文 献

[1] 冷亚军,陆 青,梁昌勇. 协同过滤推荐技术综述. 模式识别与人工智能, 2014, 27(8): 720 – 734.
(LENG Y J, LU Q, LIANG C Y. Survey of Recommendation Based on Collaborative Filtering. Pattern Recognition and Artificial Intelligence, 2014, 27(8): 720 – 734.)

[2] GOLDBERG D, NICHOLS D, OKI B M, *et al*. Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM, 1992, 35(12): 61 – 70.

[3] ADOMAVICIUS G, TUZHILIN A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans on Knowledge and Data Engineering, 2005, 17(6): 734 – 749.

[4] 冷亚军,梁昌勇,丁 勇,等. 协同过滤中一种有效的最近邻选择方法. 模式识别与人工智能, 2013, 26(10): 968 – 974.
(LENG Y J, LIANG C Y, DING Y, *et al*. Method of Neighborhood Formation in Collaborative Filtering. Pattern Recognition and Artificial Intelligence, 2013, 26(10): 968 – 974.)

[5] WANG J, DE VRIES A P, REINDERS M J T. Unifying User-Based and Item-Based Collaborative Filtering Approaches by Similarity Fusion // Proc of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, USA, 2006: 501 – 508.

[6] LIANG C Y, LENG Y J. Collaborative Filtering Based on Information-Theoretic Co-clustering. International Journal of Systems Science, 2014, 45(3): 589 – 597.

[7] KOREN Y, BELL R, VOLINSKY C. Matrix Factorization Techniques for Recommender Systems. Computer, 2009, 42(8): 30 – 37.

[8] 孟祥武,刘树栋,张玉洁,等. 社会化推荐系统研究. 软件学报, 2015, 26(6): 1356 – 1372.
(MENG X W, LIU S D, ZHANG Y J, *et al*. Research on Social Recommender Systems. Journal of Software, 2015, 26(6): 1356 – 1372.)

[9] BREESE J S, HECKERMAN D, KADIE C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering // Proc of the 14th Conference on Uncertainty in Artificial Intelligence. Minneapolis, USA, 1998: 43 – 52.

[10] TANG J L, HU X, LIU H. Social Recommendation: A Review. Social Network Analysis and Mining, 2013, 3(4): 1113 – 1133.

[11] 吴 湖,王永吉,王 哲,等. 两阶段联合聚类协同过滤算法. 软件学报, 2010, 21(5): 1042 – 1054.
(WU H, WANG Y J, WANG Z, *et al*. Two-Phase Collaborative Filtering Algorithm Based on Co-clustering. Journal of Software, 2010, 21(5): 1042 – 1054.)

[12] VUCETIC S, OBRADOVIC Z. Collaborative Filtering Using a Regression-Based Approach. Knowledge and Information Systems, 2004, 7(1): 1 – 22.

[13] CAI D, WANG X H, HE X F. Probabilistic Dyadic Data Analysis with Local and Global Consistency // Proc of the 26th Annual International Conference on Machine Learning. Montreal, Canada, 2009: 105 – 112.

[14] HARVEY M, CARMAN M J, RUTHVEN I, *et al*. Bayesian Latent Variable Models for Collaborative Item Rating Prediction // Proc of the 20th ACM International Conference on Information and Knowledge Management. Glasgow, UK, 2011: 699 – 708.

[15] CHENG G H, GONG S J. An Efficient Collaborative Filtering Algorithm with Item Hierarchy // Proc of the 2nd International Symposium on Intelligent Information Technology Application. Shanghai, China, 2008, III: 28 – 31.

[16] QIAN X M, FENG H, ZHAO C S, *et al*. Personalized Recommendation Combining User Interest and Social Circle. IEEE Trans on Knowledge and Data Engineering, 2013, 26(7): 1763 – 1777.

[17] LIU F K, LEE H J. Use of Social Network Information to Enhance Collaborative Filtering Performance. Expert Systems with Applications, 2010, 37(7): 4772 – 4778.

[18] YANG X W, STECH H, LIU Y. Circle-Based Recommendation in Online Social Networks [C/OL]. [2015 – 04 – 25]. http://ee-web.poly.edu/faculty/yongliu/docs/CircleRec.pdf.

[19] LÜ L Y, ZHOU T. Link Prediction in Complex Networks: A Survey. Physica A: Statistical Mechanics and Its Applications, 2011, 390(6): 1150 – 1170.

[20] DESHPANDE M, KARYPIS G. Item-Based Top-*N* Recommendation Algorithms. ACM Transaction on Information Systems, 2004, 22(1): 143 – 177.

作者简介

郭兰杰,男,1991 年生,硕士研究生,主要研究方向为推荐系统. E-mail: 1178315430@qq.com.
(**GUO Lanjie**, born in 1991, master student. His research interests include recommendation system.)

梁吉业(通讯作者),男,1962 年生,博士,教授,主要研究方向为粒计算、数据挖掘、机器学习. E-mail: ljy@sxu.edu.cn.
(**LIANG Jiye**(Corresponding author), born in 1962, Ph. D., professor. His research interests include granular computing, data mining and machine learning.)

赵兴旺,男,1984 年生,博士研究生,主要研究方向为数据挖掘、机器学习. E-mail: zhaoxw84@163.com.
(**ZHAO Xingwang**, born in 1984, Ph. D. candidate. His research interests include data mining and machine learning.)