考虑用户活跃度和项目流行度的基于项目最近邻的协同过滤算法

王锦坤 姜元春 孙见山 孙春华

(合肥工业大学管理学院 合肥 230009) (过程优化与智能决策教育部重点实验室 合肥 230009)

摘 要 项目相关性度量是基于项目最近邻的协同过滤算法的关键。已有的项目相关性度量方法在数据集稀疏或推荐低流行度产品时会面临较大挑战,因此提出一种考虑用户活跃度和项目流行度的基于项目最近邻的协同过滤算法。该算法在度量两个项目的相关性时,若有记录只对两个项目之一有评分,则利用该记录所对应的评分用户的活跃度和被评价项目的流行度进行相关性惩罚,从而提高数据稀疏环境下低流行度产品被推荐的概率。实验表明,所提算法在保证评分预测精度的情况下提升了推荐结果的多样性和新颖性。

关键词 个性化推荐,相关性度量,协同过滤,用户活跃度,项目流行度

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.12.028

Item-based Collaborative Filtering Algorithm Integrating User Activity and Item Popularity

WANG Jin-kun JIANG Yuan-chun SUN Jian-shan SUN Chun-hua

(School of Management, Hefei University of Technology, Hefei 230009, China)

(Key Laboratory of Process Optimization and Intelligent Decision-making, Ministry of Education, Hefei 230009, China)

Abstract Item correlation computation is the most critical component in item-based collaborative filtering algorithm. The traditional correlation computation scheme can be challenged by the sparse data set and the situation of recommending unpopular products. In this paper, a novel item-based collaborative filtering algorithm that incorporates the activity of users and popularity of items was proposed. The proposed computation scheme decreases the correlation between items using the activity of users and popularity of items in those rating records where only one item is rated. In this way, the unpopular products can be recommended to users in the sparse data. Experimental evaluation shows that the diversity and novelty of the recommendation list can be improved while maintaining the prediction accuracy.

Keywords Personalized recommendation, Correlation computation, Collaborative filtering, Activity of users, Popularity of items

1 引言

基于项目最近邻的协同过滤算法是推荐系统领域最为经典的算法,其基本思想是根据目标用户的评分记录计算未评分项目的最近邻项目,并根据最近邻项目的评分预测目标用户对未评分项目的偏好。由于具有较强的可扩展性和可解释性,基于项目最近邻的协同过滤算法在电商网站产品推荐[1]、社交网络中的广告推荐[2]等领域得到了广泛的应用。

相关度计算是基于项目最近邻协同过滤方法的关键步骤。在计算两个项目的相关性时,传统方法只考虑对两个项目均有评分的记录,并运用皮尔逊相关系数、余弦相关度等方法得到项目相关度[3]。传统方法在实际应用过程中存在如下两方面问题。一方面,由于实际数据通常具有稀疏性特征,对两个项目均有评分的记录通常较少。例如,通过分析MovieLens100k数据集发现,没有任何用户同时对高达30%的项目对(项目的两两组合)进行评分;对于单一项目,平均只有60个用户对其进行评分;对于项目对,平均只有10个用户对它们均有评分,其它统计指标如表1所列。在研究中发现,对于任意给定的两个项目,它们之间存在着大量的仅对两个项目之一有评分的记录,直接忽略这些记录对相关性度量的精度有着较大影响。另一方面,按照传统方法得到的最近邻居中,具有少量共同评价数据的项目对往往会得到很高的相关度。基于MovieLens100k数据集的统计数据(见表2)表明,与i^*相关度较高的项目只是基于很少的共同评分记录通过计算得到。基于少量共同评价数据计算得到的邻居可靠性不高,对个性化推荐的效果会产生负面影响。

表1 MovieLens100K的统计数据

没有共同评分项的项目对所占比例	对两个项目均有评价的平均记录数	对一个项目有过评价的平均记录数	数据集中的用户和项目个数
30.45%	10.22	59.45	9431682

到稿日期:2015-11-14 返修日期:2016-03-23 本文受973计划(2013CB329600),国家自然科学基金(71371062,71302064),教育部人文社科基金(12YJC630073)资助。

王锦坤(1990—),男,博士生,主要研究方向为个性化推荐、数据挖掘,E-mail:wangjinkun90@foxmail.com;姜元春(1980—),男,博士,副教授,主要研究方向为个性化营销;孙见山(1987—),男,博士,讲师,主要研究方向为个性化推荐、社交网络;孙春华(1977—),女,博士,副教授,主要研究方向为互联网营销。

表 2　传统相关性度量方法所得项目最近邻居项与目标项的共同评分记录数

项目 id	共同评分记录个数 1st	2nd	3rd	4th	5th	6th	i^* 的评分记录个数
i_1	2	2	1	1	1	1	452
i_2	2	7	2	2	2	2	392
i_3	2	2	1	1	1	1	219
…	…	…	…	…	…	…	…

GroupLens 的研究人员在开源推荐工具 LensKit 中提出了一种相关性度量的新方法，该方法假设用户评价了一个项目而没有评价另一个项目代表着用户对未评价项目不感兴趣[4]，并对仅评价一个项目的记录进行相关性惩罚。虽然该算法取得了优于传统算法的评分预测精度，但是其无法推荐低流行度的产品，推荐结果的多样性和新颖性受到负面影响。本文认为，除了用户对项目不感兴趣，用户未看到项目也是用户没有给出项目评分的重要原因。用户没有看到项目的原因主要有两方面：1)用户本身的活跃度较低；2)项目本身的流行度较低。因此，对于不同的用户和项目导致评价缺失的原因不同，应当区别对待。为了解决上述问题，提出一种考虑用户活跃度和项目流行度的项目相关性度量方法。针对仅对两个项目之一有评分的记录，利用该记录所对应的评分用户的活跃度和被评价项目的流行度进行项目的相关性惩罚，在此基础上提出一种考虑用户活跃度和项目流行度的基于项目最近邻的协同过滤算法。实验表明，所提算法在保证评分预测精度的情况下增加了推荐列表的多样性和新颖性。

2　相关研究

在一个典型的推荐系统中，用户集合用 $U=\{u_1,u_2,\cdots,u_m\}$ 表示，项目集合用 $I=\{i_1,i_2,\cdots,i_n\}$ 表示。用户对项目的评分集合用矩阵 R 表示，用户对项目的评分等级集合为 S(例如，$S=\{1,2,3,4,5\}$)，若用户 u 没有对项目 i 进行评分，记 $r_{ui}=0$。推荐系统的两个重要任务是评分预测和 top-N 推荐，为进行这两项任务需要预测活跃用户 u_a 对未评价项目 i^* 的偏好。推荐系统中预测偏好的方法有基于最近邻的方法[5]和基于模型的方法[6,7]。在基于项目最近邻的协同过滤算法中，项目相关性度量是基于项目最近邻的协同过滤算法的关键，也是研究者的研究焦点[3-5,8-10]。Sarwar 等人[3]在基于项目最近邻协同过滤算法中强调在计算两个项目的相关度时，必须首先隔绝出对这两个项目都有评分的用户，然后应用相关度计算方法(如皮尔逊相关系数)计算两个项目的相关度。这种方法存在的主要弊端是：1)忽略了大量只对其中一个项目评分的记录；2)基于此方法得到的相关度排名较高的项目与待测项目只有少量的共同评价记录，因而计算得到的相关度可靠性较低。为解决第一个问题，Breese 等人[8]提出：对于仅评价一个项目的记录，将该记录中的未评分项用一个缺省值代替进而计算相关度。但是实验发现这种方法的效果要差于原始方法，主要原因在于用户对于未评分项的评分并不完全相同，用缺省值方法太过主观。而针对第二个问题，Herlocker 等人[9]提出了一种基于可信度的相关度线性下降计算方法，设定一个项目共同评分记录阈值来惩罚有着较少共同评分记录的项目相关度。这样就降低了有着较少共同评分记录的项目相关度的值。这种做法没有解决第一个问题，而且很容易预见的是推荐系统中流行度较低的产品很难被推荐给用户。2011 年，Ekstrand 在 RecSys 年会上公布开源推荐工具 LensKit 时，针对基于项目最近邻的协同过滤算法提出了一种新的相关性度量公式，并称之为“均值标准化向量上的余弦相似度”[4]。在分子上，该公式与皮尔逊公式完全相同；而在分母上，利用只对一个项目有评分的记录惩罚两个项目的相关性。这样当存在大量的用户只对其中一个项目有评分行为时，两个项目之间的相关性会降低。离线实验结果表明这种相关性度量方法在平均绝对误差等预测评分精确度指标上取得了最好的效果。Sarwar[3]，Herlocker[9] 和 Ekstrand[4] 的项目相关性度量公式如表 3 所列。

表 3　项目相关性度量公式

度量方法	具体公式
Pearson[3]	$\mathrm{sim}(i,j)=\dfrac{\sum_{u\in U_i\cap U_j}(r_{u,i}-\bar{r}_i)(r_{u,j}-\bar{r}_j)}{\sqrt{\sum_{u\in U_i\cap U_j}(r_{u,i}-\bar{r}_i)^2}\sqrt{\sum_{u\in U_i\cap U_j}(r_{u,j}-\bar{r}_j)^2}}$
WeightedPearson[9]	$\mathrm{sim}(i,j)=\dfrac{\max(\lvert U_i\cap U_j\rvert,\gamma)}{\gamma}\mathrm{sim}'(i,j)$
CosineNormItem[4]	$\mathrm{sim}(i,j)=\dfrac{\sum_{u\in U_i\cap U_j}(r_{u,i}-\bar{r}_i)(r_{u,j}-\bar{r}_j)}{\sqrt{\sum_{u\in U_i}(r_{u,i}-\bar{r}_i')^2}\sqrt{\sum_{u\in U_j}(r_{u,j}-\bar{r}_j')^2}}$

3　算法设计

在实际电子商务系统中，用户对项目的评分矩阵中存在大量的未评分数据。缺失数据的统计理论表明对缺失数据的建模会影响推荐算法的精度[11,12]。针对这些未评分数据，假设用户对于未评价项目的偏好与其本身的活跃度和项目的流行度相关。具体地，在计算项目相关度时，若有记录只对两个项目之一进行评分，则利用该记录对应评分用户的活跃度和被评价项目的流行度进行相关性惩罚，具体实现方法如下。

3.1　用户活跃度

推荐系统中用户的活跃度存在较大的差异，用户活跃度可以用于隐性推断用户对未评分项目感兴趣的可能性。例如，存在两个微博用户，一个是微博活跃用户，其每天花费大量的时间浏览微博内容；另一个是不活跃用户，其每个星期可能仅仅登录一两次微博。假设这两个用户都只对“火箭击败开拓者”这条篮球新闻点了赞，而没有对“皇马加时完胜马竞”这条足球新闻点赞，我们认为对于活跃的用户，(她)他有很大可能看到了这则新闻，但是她(他)对篮球类新闻感兴趣而对足球类新闻不感兴趣；而对于不活跃用户，她(他)没有对足球类新闻点赞的原因更有可能是没有看到这则新闻。据此，本文假设用户的活跃度越高，用户对未评价项目不感兴趣的可能性越大。本文定义用户的活跃度与用户浏览过的项目的数量成正比，在计算每个用户的活跃度时进行向量标准化：

$$UserAct(u)=\frac{\lvert I_u\rvert}{\sqrt{\sum_{u\in U}\lvert I_u\rvert^2}} \tag{1}$$

3.2　项目流行度

在计算项目对的相关度时，项目对的流行度差异也能隐式表征用户的潜在偏好。任意给定两个项目 i 和 j，假设项目 i 的流行度高，项目 j 的流行度低。此时，如果用户评价了项目 i 而没有评价项目 j，很有可能是因为用户不知道项目 j。项目 i 和项目 j 的流行度差异越大，因为不知道项目 j 而未评分的可能性越大。另一种情况，若用户评价了流行度较低的项目 j 而没有评价流行度较高的项目 i，则此时用户不太可能

是因为没有看到项目 i，而是因为用户对项目 j 感兴趣而对项目 i 没有兴趣。基于上述分析，项目流行度差异影响项目相关度计算。据此定义项目 i 和项目 j 的流行度差异为：

$$PopDiff_{i,j}=ItemPop_i-ItemPop_j \tag{2}$$

其中，$ItemPop_i$ 为项目流行度，本文定义项目流行度与浏览过该项目的用户数量成正比，在计算每个项目的流行度时进行向量标准化：

$$ItemPop_i=\frac{|U_i|}{\sqrt{\sum_{i\in I}|U_i|^2}} \tag{3}$$

3.3 项目相关度计算

若有记录只对两个项目之一进行评分，本文利用该记录对应评分用户的活跃度和两个项目的流行度差异设计项目相关度惩罚权重。若仅对一个项目有评分的记录所对应的评分用户的活跃度越高，用户不喜欢未评价项目的可能性越大，该条评分记录对于项目相关性的贡献越低，相关性惩罚权重也越大，因而相关性惩罚权重正相关于用户活跃度；当两个项目的流行度差异越小，用户只看到两个项目之一的可能性也越低，相关性惩罚权重也越大，因而相关性惩罚权重负相关于项目流行度之差。仅对两个项目之一有评分的记录的相关性惩罚权重计算公式为：

$$w_u(i,j)=\begin{cases}1, & \text{if } PopDiff_{i,j}>\beta\\ \dfrac{UserAct_u}{|PopDiff_{i,j}|}, & \text{if } 0<PopDiff_{i,j}<\beta\\ \dfrac{UserAct_u}{|PopDiff_{i,j}|}*\alpha, & \text{if } PopDiff_{i,j}<0\end{cases} \tag{4}$$

其中，β 是算法设定的项目流行度差异阈值，即当两个项目的流行度差异超过一定阈值时，若有记录只对两个项目之一进行评分，不再考虑该记录对应的项目流行度差异和用户活跃度；α 是当用户评价了流行度较低的项目而没有评价流行度较高的项目的额外惩罚权重。

综上，对任意项目 i 和项目 j，本文定义其相关度计算公式为：

$$sim(i,j)=\frac{\sum_{u\in U_i\cap U_j}(r_{u,i}-\bar{r}_i)(r_{u,j}-\bar{r}_j)}{\sqrt{\sum_{u\in U_i\cap U_j}(r_{u,i}-\bar{r}_i)^2+\sum_{u\in U_i\&\,u\notin U_j}(r_{u,i}-\bar{r}_i)^2*w_u(i,j)}\sqrt{\sum_{u\in U_i\cap U_j}(r_{u,j}-\bar{r}_j)^2+\sum_{u\in U_j\&\,u\notin U_i}(r_{u,j}-\bar{r}_j)^2*w_u(i,j)}} \tag{5}$$

3.4 算法描述

基于所提的用户活跃度、项目流行度、相关性惩罚函数和项目相关性度量公式，本文提出考虑用户活跃度和项目流行度的基于项目最近邻的协同过滤算法（User Activity and Item Popularity Based Collaborative Filtering Algorithm，UA-BCF）如下。

算法 考虑用户活跃度和项目流行度的基于项目最近邻的协同过滤算法（UA-BCF）

输入：用户-项目评分矩阵 R

输出：用户 u_a 对项目 i^* 的预测评分

Step1 利用式(5)基于用户-项目的评分矩阵 R 计算项目 i^* 与所有其它项目的相关度 $sim(i^*,i)$；

Step2 基于相关度 $sim(i^*,i)$，根据 $sim(i^*,i)$ 的大小进行降序排列并选择用户 u_a 评价过的前 K 个项目作为项目 i^* 的邻居 $Neigh(i^*)$；

Step3 根据用户 u_a 对邻居中项目的评分 $r(u_a,j)(j\in Neigh(i^*))$，利用式(1)加权得到活跃用户对未知项目的预测评分 $\hat{r}(u_a,i^*)$，权重为邻居中项目与活跃项目的相关度 $sim(i^*,j)$。

复杂度分析：传统项目相关度计算方法的复杂度为 $O(N^2)$（其中 N 为项目个数），本文提出的项目相关度计算方法要求预计算用户的活跃度和项目的流行度，计算用户活跃度和项目的流行度的时间复杂度均为 $O(NM)$，因此所提项目相关度计算方法的时间复杂度为 $O(N^2+2NM)$，等价于 $O(N*\max(N,2M))$，其中 M 为用户个数。

4 实验评价

4.1 数据集

本文选择 GroupLens 实验室发布的 MovieLens100K 电影数据集对所提算法的有效性进行验证。MovieLens100K 数据集（简记为 ML100K）由 1997 年到 1998 年 7 个月内在MovieLens网站收集的用户对电影的评分数据构成，包含 943 个用户对 1682 部电影的 100000 条评分记录，每个用户至少对 20 部电影有过评分。该电影数据集的稀疏度为 0.9369。

本文采用 k 折交叉验证方法对 ML100K 进行分割[4]。首先将数据中的所有用户均匀分配到 k 个数据集中，对于其中任一数据集，从每个用户评分记录中随机选择 n 条记录放到测试集中，将该数据集的剩余评分和其它 $k-1$ 个数据集的评分放到训练集中。重复进行 k 次，得到 k 对训练集和测试集，并在所得的 k 对训练集和测试集上计算算法度量指标的均值。本文实验中设 k 为 5，设 n 为 10。

4.2 评价指标

本文利用 MAE，nDCG[13]，Diversity[14] 和 Novelty[15] 4 个指标分别从预测评分精确度、推荐列表排序精确度、推荐列表多样性和推荐新颖性 4 个角度对算法性能进行度量。4 个度量指标的计算公式如下：

$$MAE=\frac{\sum_{i=1}^{N}|p_i-q_i|}{|Testset|} \tag{6}$$

其中，q_i 是测试集中用户的实际评分值，p_i 是对应的算法预测值，$|Testset|$ 是测试集中评分记录的个数，记录总数为 N 和 MAE 值越小，算法性能越好。

$$nDCG@M=\frac{1}{Z_M}\sum_{i=1}^{M}\frac{2^{r_i}-1}{\log_2(i+1)} \tag{7}$$

其中，r_i 是排名第 i 个项目的相关度评分。本实验中，若对应的项目评分高于用户的评分均值，则 r_i 为 1，否则为 0。Z_m 是 $nDCG$ 的标准化因子，本文将 M 设置为 5，表示度量的是长度为 5 的推荐列表的排序精确度。$nDCG$ 值越高，表示推荐列表的排序越精确。

$$Diversity=\frac{\sum_{u,v\in U}diversity_{u,v}(L)}{\frac{1}{2}|U|(|U|-1)} \tag{8}$$

$$diversity_{u,v}(L)=1-\frac{q_{uv}(L)}{L} \tag{9}$$

其中，$diversity_{u,v}(L)$ 是指系统给用户 u 和用户 v 的推荐列表的差异性度量，$q_{uv}(L)$ 是系统给用户 u 和用户 v 的长度为 L 的推荐列表中共同项目的个数，$Diversity$ 是推荐系统的多样性，$|U|$ 为系统中的总用户数。

$$Novelty=\frac{\sum_{u\in U}novelty_u}{|U|} \tag{10}$$

$$novelty_u=\frac{\sum_{i\in L_u} pop(i)}{|L_u|} \tag{11}$$

本文根据推荐列表中项目的平均流行度来评价推荐系统的新颖性，该公式假设流行度越低的产品越能让用户感到新颖。其中 $pop(i)$ 为给用户 u 的推荐列表中项目 i 的流行度，即选择该项目的用户数。

本文将 UA-BCF 算法与基于皮尔逊的项目最近邻方法（实验中简记为 Pearson）[3]、基于加权皮尔逊的项目最近邻方法（WeightedPearson）[9]和基于均值标准化余弦相关度的项目最近邻方法（CosineNormItem）[4] 3 种基准算法进行比较（具体项目相关性计算公式见表 3）。为验证所提项目相关性度量方法的有效性，本文方法与基准方法采取了相同的邻居选择方法和评分加权方法。

本文所有实验代码均用 Matlab R2013a 编写。实验在 Intel Xeon 2.67 GHz CPU 和 8GB 内存的 64 位 Windows Server 2008 平台上运行。

4.3 评价指标

4.3.1 参数敏感性实验

为了验证所提方法的鲁棒性，本文对影响 UA-BCF 算法效果的参数 α 和 β 进行敏感性分析。为了测试参数 β 对算法精度的影响，假设 α 为 2，β 的取值范围为[0.04，0.08]。UA-BCF 算法的平均绝对误差随着邻居规模的变化如图 1 所示。图 1 表明，当取不同数值时，UA-BCF 的平均绝对误差指标比较稳定（在区间[0.76，0.765]之间）。当 β 取 0.05 时，算法的平均绝对误差最小。这说明，当两个项目的流行度差异较大时，不考虑仅对一个项目有评分记录对应的项目流行度差异和用户活跃度，有利于提高算法的评分预测精度。为了测试参数 α 对算法精度的影响，将 β 设置为 0.05，α 的取值范围设为[1，2.25]，算法的平均绝对误差随着邻居规模的变化如图 2 所示。图 2 表明，α 为 1.75 和 2 时算法的平均绝对误差小于 α 为 1 和 1.25 时的平均绝对误差。这说明，针对仅对一个项目有评分的记录，若用户评价了流行度较低的项目而没有评价流行度较高的项目，提高对这类记录的惩罚权重有利于提高评分预测精度。

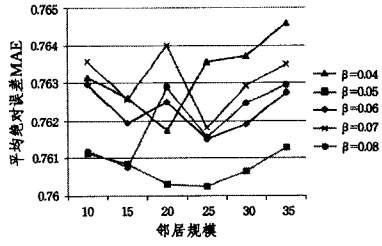

图 1 参数 β 敏感性测试（$\alpha=2$）

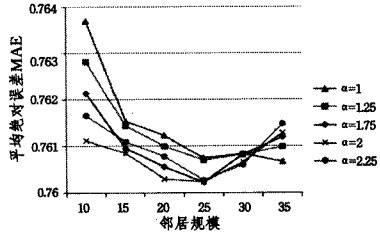

图 2 参数 α 敏感性测试（$\beta=0.05$）

4.3.2 与基准算法的对比实验

由图 1 和图 2 可知，当 α 为 2 且 β 为 0.05 时，UA-BCF 算法可以取得较好的效果。本文后续实验均假设 $\alpha=2$，$\beta=0.05$。本文所提 UA-BCF 算法与基准算法的对比实验如图 3 和图 4 所示。

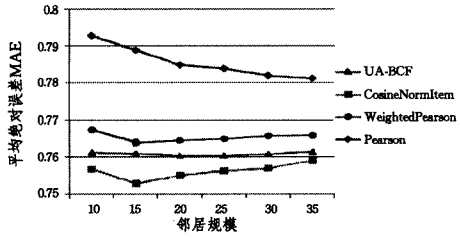

图 3 评分预测精度指标 MAE 对比结果

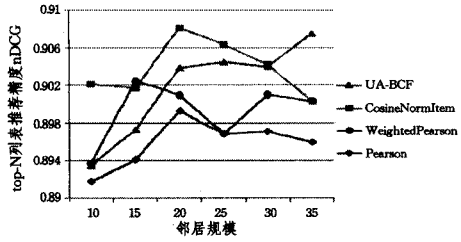

图 4 top-N 推荐列表排序精度指标 nDCG 对比结果

图 3 和图 4 给出了不同邻居规模下各种算法的评分预测精度指标 MAE 值和推荐列表排序精度指标 nDCG 值。在评分预测精度指标 MAE 上，CosineNormItem 方法具有最好的推荐效果，而 Pearson 方法的推荐效果最差。这说明，综合利用仅对一个项目有评分的记录并惩罚具有较少共同评分记录的项目对的相关度有助于提升推荐精度。同时，图 3 和图 4 还表明，当计算项目相关度所依据的邻居规模较少时，本文所提 UA-BCF 方法的效果略低于 CosineNormItem 方法。当计算项目相关度所依据的邻居规模大于一定规模后，UA-BCF 方法与 CosineNormItem 方法的效果基本相同。但是，UA-BCF 方法在 top-N 列表推荐精度指标 nDCG 上取得了更好的效果。如图 4 所示，当邻居规模为 35 时，UA-BCF 方法具有最高的推荐列表精度。

图 5 和图 6 给出了不同推荐列表长度下各种算法的多样性指标 Diversity 和新颖性指标 Novelty 的对比情况。图 5 表明，WeightedPearson 方法的推荐多样性和流行度最差。主要原因在于 WeightedPearson 方法惩罚了具有较少共同评分记录的项目对，使得流行度较低的项目具有较少的相关项目，在推荐时被考虑的概率大幅降低。在 MAE 指标取得最好效果的 CosineNormItem 方法的推荐多样性和新颖性较低，具体原因与 WeightedPearson 方法类似。虽然 Pearson 方法在评分预测精度指标 MAE 和推荐列表排序精度上效果最差（见图 3 和图 4），但是其取得了最优的推荐多样性和流行度。这是因为对于具有较少共同评分记录的项目对，Pearson 方法可以赋予较高的相关度。因此，对于流行度较低的项目，Pearson 方法也可以将其推荐给用户。图 5 和图 6 表明，本文所提的 UA-BCF 方法在推荐多样性和新颖性两方面均取得了较为理想的结果。在保持推荐精度的情况下，UA-BCF 方法的 Diversity 和 Novelty 指标值均优于 CosineNormItem 和 WeightedPearson方法。

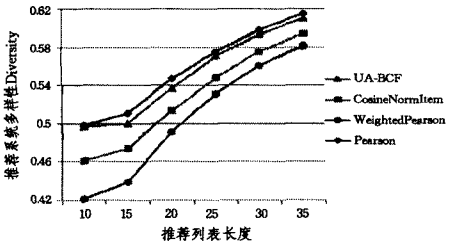

图 5　推荐多样性指标 Diversity 对比结果

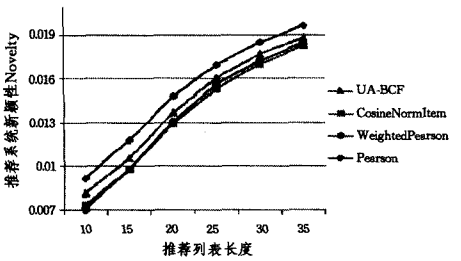

图 6　推荐新颖性指标 Novelty 对比结果

结束语　本文利用用户活跃度和项目流行度信息对现有项目相关性度量方法进行了改进，在保证推荐精度的情况下提高了推荐系统的多样性和新颖性。所提相关性度量方法的基本思想是，对于任意给定的两个项目，若存在较多的仅对两个项目之一有评分的记录，两个项目的相关性会降低。相关性下降程度取决于这些记录对应的用户活跃度和项目流行度。用户活跃度越高，项目相关性下降程度越大，项目流行度差异越小，项目相关性下降程度越大。基于电影数据集 ML100K 的实验结果表明，所提算法可以作为现有基于项目最近邻的协同过滤算法的有效补充。在对多样性和新颖性要求较高的推荐环境中，该算法具有较大的优势。在本文模型中，惩罚权重对相关性度量的效果具有重要作用，如何对惩罚权重进行改进以获得更高的推荐精度是本文后续研究的主要方向。

参 考 文 献

[1] Linden G, Smith B, York J. Amazon. com recommendations: Item-to-item collaborative filtering [J]. IEEE Internet Computing, 2003, 7(1): 76-80

[2] 项亮. 推荐系统实践[M]. 人民邮电出版社, 2012

[3] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C] // Proceedings of the 10th International Conference on World Wide Web. ACM, 2001: 285-295

[4] Ekstrand M D, Ludwig M, Konstan J A, et al. Rethinking the recommender research ecosystem: reproducibility, openness, and LensKit[C] // Proceedings of the Fifth ACM Conference on Recommender Systems. ACM, 2011: 133-140

[5] Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of netnews[C] // Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work. ACM, 1994: 175-186

[6] Zhang Yin, Zhang Bin, Gao Ke-ning, et al. Autonomy Oriented Personalized Tag Recommendation[J]. Acta Electronica Sinica, 2012, 40(12): 2353-2359(in Chinese)
张引，张斌，高克宁，等. 面向自主意识的标签个性化推荐方法研究[J]. 电子学报, 2012, 40(12): 2353-2359

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

[8] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C] // Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc., 1998: 43-52

[9] McLaughlin M R, Herlocker J L. A collaborative filtering algorithm and evaluation metric that accurately model the user experience[C] // Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2004: 329-336

[10] 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

[11] Marlin B M, Zemel R S. Collaborative prediction and ranking with non-random missing data[C] // Proceedings of the third ACM Conference on Recommender Systems. ACM, 2009: 5-12

[12] Little R J A, Rubin D B. Statistical analysis with missing data [M]. John Wiley & Sons, 2014

[13] Järvelin K, Kekäläinen J. Cumulated gain-based evaluation of IR techniques[J]. ACM Transactions on Information Systems (TOIS), 2002, 20(4): 422-446

[14] Zhou T, Kuscsik Z, Liu J G, et al. Solving the apparent diversity-accuracy dilemma of recommender systems[J]. Proceedings of the National Academy of Sciences, 2010, 107(10): 4511-4515

[15] Vargas S, Castells P. Rank and relevance in novelty and diversity metrics for recommender systems[C] // Proceedings of the fifth ACM Conference on Recommender Systems. ACM, 2011: 109-116

(上接第 152 页)

[8] Brin S, Motwani R, Ullman J D, et al. Dynamic itermset counting and implication rules for market basket data[J]. ACM SIGMOD Record, ACM, 1997, 26(2): 255-264

[9] Han Jia-wei, Kamber M. Data Mining: Concepts and Techniques [M]. Beijing: China Machine Press, 2001: 130-160(in Chinese)
韩家炜，坎伯. 数据挖掘：概念与技术[M]. 北京：机械工业出版社，2001：130-160

[10] Agrawal R, Srikant R. Fast algorithm for mining association rules[C] // Proc. 20th Int. Conf. Very Large Data Bases (VLDB). 1994, 1215: 487-499

[11] Van V S, Via J, Santamana I. A Sliding-Window Kernel RLS Algorithm and Its Application to Nonlinear Channel Identification [C] // Proceedings of ICASSP 2006. Toulouse, France, 2006

[12] Wei Ling, Wei Yong-jiang, Gao Chang-yuan. Improved Apriori Algorithm Based on Bigtable and MapReduce[J]. Computer Science, 2015, 42(10): 208-210(in Chinese)
魏玲，魏永江，高长元. 基于 Bigtable 与 MapReduce 的 Apriori 算法改进[J]. 计算机科学，2015，42(10)：208-210，243

[13] Xu Zhang-yan, Liu Mei-ling, Zhang Shi-chao, et al. Three Optimized Methods of Apriori Algorithm[J]. Computer Engineering and Applications, 2004, 40(36): 190-193(in Chinese)
徐章艳，刘美玲，张师超，等. Apriori 算法的 3 种优化方法[J]. 计算机工程与应用，2004，40(36)：190-193