DOI: 10.3969/j.issn.0255-8297.2017.03.006

基于用户和项目的双视角协同过滤推荐方法

程树林[1,2], 张博锋[1], 邹国兵[1]
1. 上海大学 计算机工程与科学学院, 上海 200444
2. 安庆师范大学 计算机与信息学院, 安徽 安庆 246133

摘 要: 传统的协同过滤推荐方法存在单视角信息利用不足、预测精度不高、对数据稀疏性敏感等问题, 为此提出同时考虑相似用户和相似项目的双视角协同过滤推荐方法. 根据辩证的思想, 利用项目内部因子和外部因子生成项目融合相似度, 有效度量了项目相似性和用户相似性, 并解决了双视角协同过滤推荐方法对数据稀疏性敏感的问题. 在标准数据集上多次进行的实验表明, 基于用户和项目的双视角协同过滤推荐方法优于多个典型的协同过滤推荐方法.

关键词: 协同过滤推荐; 双视角; 融合相似度
中图分类号: TP181 **文章编号:** 0255-8297(2017)03-0326-11

Collaborative Filtering Recommendation Based on Double-Perspective of Users and Items

CHENG Shu-lin[1,2], ZHANG Bo-feng[1], ZOU Guo-bing[1]
1. School of Computer and Science Engineering, Shanghai University, Shanghai 200444, China
2. Institute of Computer and Information, Anqing Normal University, Anqing 246133, Anhui Province, China

Abstract: Traditional collaborative filtering (CF) recommendation approach has a serious problems such as insufficient usage of single perspective information, unsatisfactory accuracy and sensitivity to data sparsity. To solve these problems, a CF recommendation method based on double-perspective of users and items is proposed by considering information of similar users and similar items. According to the dialectic principle, fusion similarity of items is given by combination of inner-factors and outer-factors of the item. This way, the item similarity and user similarity can be effectively measured. The measurement is robust against data sparsity in the approach of CF recommendation based on double-perspective of user and item. Several experiments are carried on benchmark datasets. The results show that the proposed CF recommendation method based on double-perspective of users and items outperforms several other typical CF approaches.

Keywords: collaborative filtering recommendation, double-perspective, fusion similarity of item

---

收稿日期: 2016-02-28; **修订日期:** 2016-10-03
基金项目: 国家自然科学基金(No.61303096); 上海市自然科学基金(No.13ZR1454600)资助
作者简介: 程树林, 博士生, 副教授, 研究方向: 个性化推荐, E-mail: chengshulin@shu.edu.cn; 张博锋, 研究员, 博导, 研究方向: 智能信息处理、个性化推荐, E-mail: bfzhang@shu.edu.cn

协同过滤推荐方法分为基于内存和基于模型两类，广泛应用于推荐领域[1-3]. 基于模型的方法是利用机器学习理论建立数学模型实现推荐[3]，基于内存的方法属于启发式方法，包括基于项目的方法 (item-based collaborative filtering, IBCF) 和基于用户的方法 (user-based collaborative filtering, UBCF)[3]两种. 当给定用户项目评分矩阵时，基于项目和基于用户两种方法原理类似，均使用近邻思想搜索出最相似的项目或用户列表进行推荐.

现有基于项目和用户的方法多采用单个视角，即单纯利用项目或用户相关信息进行推荐，在一定程度上取得了较好的推荐效果，如著名的亚马逊图书推荐[4]. 然而，使用评分矩阵中单方面的内嵌信息会导致现有方法出现数据稀疏问题. 文献 [5-9] 利用插值和组合方法来改进协同过滤推荐方法，提高了推荐性能. 在用户项目评分矩阵中，同时考虑项目和用户两方面信息进行协同过滤推荐方法的改进方面的研究较少. 本文主要研究了基于用户和项目双视角的协同过滤推荐方法，并提出基于项目内部和外部因子组合的项目融合相似度方法，缓解用户项目评分矩阵的稀疏性问题.

文献 [10] 研究了协同过滤推荐方法，识别兴趣爱好类似的用户进行项目的协同推荐. 文献 [4,5,11-13] 相继研究了基于用户、项目、模型的协同过滤推荐方法. 为了提高传统的协同过滤推荐方法精度，现有研究主要从 3 个方面进行改进，即相似性度量方法的优化[11,13]、预评分或评分插值[5-6]和组合推荐方法[7-9]等.

相似性度量包括余弦相似度、改进的余弦相似度和相关相似度[11,13]，它对数据集质量比较敏感. 在数据非常稀疏的情况下，该方法所得结果可信度不高[5]. 因此，一些学者针对数据稀疏问题，提出了预评分和评分插值的方法以填充缺失数据. 最简单和直接的方法是使用 0 值或平均分填补缺失评分[5]. 这种方法会导致评分矩阵中存在大量的 0 值或平均分，难以取得较好的效果. 现有文献中具有代表性的评分预测或插值方法有：1）根据相似用户预测评分，寻找目标用户的相似用户，根据相似用户对项目的评分预测目标用户对项目的预评分[5]；2）用目标用户对相似项目的评分进行预评分[6]；3）根据不同问题谨慎选择相应的插值源进行预评分[15]. 这 3 类方法取得了一定效果，但仍然不能克服评分矩阵稀疏问题. 文献 [15] 较全面地分析了利用各种信息源进行插补缺失值，并应用于 SVD 模型改进协同推荐方法. 在实际应用中，应谨慎选择合适的插值源进行缺失值插补，能够在较大程度上提高协同推荐精度. 相应的插值方法主要包括分类方法、回归方法、K-近邻方法、概率分布方法等. 近年来，协同推荐方法也广泛应用于社交网络，利用用户之间的信任和社交关系缓解数据稀疏问题[15,17]，得到了较好的效果，但在有些问题如 MovieLens 推荐中难以获取这些信息，也就无法利用它们来缓解数据稀疏问题，因此本文利用用户项目评分矩阵内含信息来缓解数据稀疏问题. 除了优选相似性度量方法和插补缺失评分方法外，组合推荐也是一种很重要的改进方法，包括以下 3 个方面：1）基于内容过滤的推荐和协同过滤推荐组合[7]；2）基于人口统计信息的推荐技术与协同推荐技术进行结合[8]；3）基于用户和基于项目的预测结果进行线性组合[6]. 组合推荐方法结合了多种方法的优点，其推荐精度得到了较大提高.

目前，协同过滤推荐方法在精度和准确性、大数据下的并行算法和实时推荐等方面还需进一步改善. 本文立足用户和项目 2 个方面的信息，提出双视角协同过滤推荐改进方法来提高评分预测精度和准确性.

1　传统的协同过滤推荐方法

传统的协同过滤推荐方法根据目标用户对未知项目的预测评分，产生基于预测评分倒排的 top-K 推荐列表. 给定 M 个用户和 N 个项目，形成 $M \times N$ 用户项目评分矩阵 R，R 内含有大量的未知评分元素. 首先通过合适的相似度度量方法计算项目或用户之间的相似度，然

后选取 k 个与目标项目或目标用户最为相似的项目或用户预测未知评分，最后为目标用户推荐 top-K 个项目. 基于项目和用户的协同推荐方法分别由式 (1) 和 (2)、式 (3) 和 (4) 给出

$$S^I(k,j)=\frac{\vec{I}_k\cdot\vec{I}_j}{||\vec{I}_k||||\vec{I}_j||} \tag{1}$$

$$\widehat{r}_{i,j}=\overline{r}_{I_j}+\frac{1}{\sum\limits_{k=1}^{K}S^I(k,j)}\sum_{k=1}^{K}S^I(k,j)(r_{i,k}-\overline{r}_{I_k}) \tag{2}$$

$$S^U(k,i)=\frac{\vec{U}_k\cdot\vec{U}_i}{||\vec{U}_k||||\vec{U}_i||} \tag{3}$$

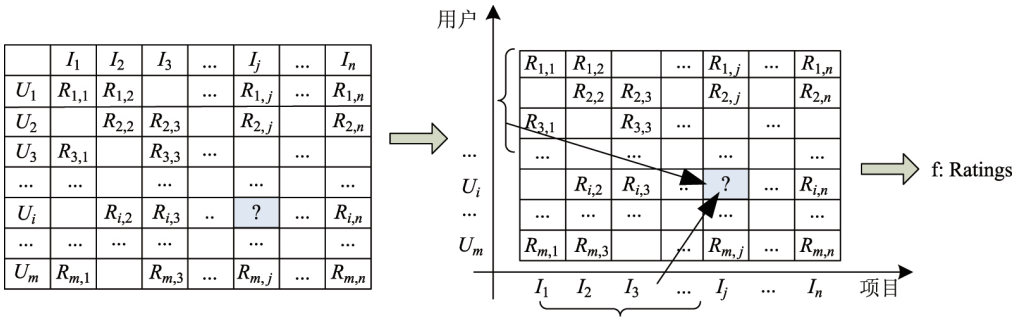
$$\widehat{r}_{i,j}=\overline{r}_{u_i}+\frac{1}{\sum\limits_{k=1}^{K}S^U(k,i)}\sum_{k=1}^{K}S^U(k,i)(r_{k,j}-\overline{r}_{u_k}) \tag{4}$$

式中，$S^I(k,j)$、$S^U(k,j)$、$\vec{I}_k$、$\vec{U}_j$ 分别为项目相似度、用户相似度、项目评分向量、用户评分向量；$r_{k,j}$ 为用户 i 对项目 k 的评分，$\widehat{r}_{i,j}$ 为预测评分，$\overline{r}_{I_j}$ 和 $\overline{r}_{u_k}$ 分别为项目平均评分和用户平均评分，式 (1) 和 (3) 用于计算项目相似度和用户相似度，式 (2) 和 (4) 分别用于实现基于项目和基于用户的未知评分预测.

2 双视角协同过滤推荐方法

2.1 双视角评分预测

传统的基于用户和项目的协同推荐方法仅采用单视角，用户或项目进行推荐不够准确和全面[3]. 若同时考虑相似用户和相似项目进行评分预测，则可以综合二者优点，提高预测精度. 因为相似用户对相似项目进行评分具有更高的可信度，从而为推荐提供更可靠的依据. 本文采用评分重排和相似度融合的方法进行评分预测，其原理如图 1 所示. 首先根据用户相似度和项目相似度重排用户项目评分矩阵，将其映射到一个二维坐标系中，再对待预测的未知评分元素 $R_{i,j}$ (图 1 中问号所对应的元素) 相关的用户和项目，融合用户相似度和项目相似度为一个综合相似度，并根据协同思想进行评分预测.



(a) 原始用户项目评分矩阵
(a) Origin user-item matrix

(b) 二维坐标系中重排的评分矩阵表示
(b) Expression of re-ranked matrix in coordinates

图 1 双视角评分预测原理

Figure 1 Principle of predicting rating based on double-perspective

图 1(a) 为原始用户项目评分矩阵，图 1(b) 为原始用户项目评分矩阵重构后在二维坐标系中的映射关系，每个元素映射到坐标系中的一个点. 与未知评分元素相关的用户为 U_i、项目为 I_j，则重排规则为：首先按照与 U_i 和 I_j 的相似度降序排列，然后选择与用户 U_i 最相似的 top-K 个用户的列表 $U_{\rm ss}$ 和与项目 I_j 最相似的 top-M 个项目的列表 $I_{\rm ss}$，最后根据式 (5) 进行双视角的评分预测.

$$R_{\rm ss}^{\rm p}(i,j)=\frac{\sum\limits_{k\in U_{\rm ss}}\sum\limits_{m\in I_{\rm ss}}S^{\rm SS}(i,j,k,m)R(k,m)}{\sum\limits_{k\in U_{\rm ss}}\sum\limits_{m\in I_{\rm ss}}S^{\rm SS}(i,j,k,m)} \tag{5}$$

式中，$S^{\rm SS}(i,j,k,m)$ 为融合相似用户和相似项目的综合相似度，其计算公式为

$$S^{\rm SS}(i,j,k,m)=\lambda_1 S^U(i,k)+\lambda_2 S^I(j,m) \tag{6}$$

式中，λ_1 和 λ_2 为融合参数，体现相似用户和相似项目在双视角评分预测中的比例，用于协调用户相似度和项目相似度. 本文利用用户相似度和项目相似度的贡献度 (比例) 来确定 λ_1 和 λ_2 的值，且有

$$\lambda_1=\frac{\sum S^U(i,k)}{\sum S^U(i,k)+\sum S^I(j,m)} \tag{7}$$

$$\lambda_2=\frac{\sum S^I(j,m)}{\sum S^U(i,k)+\sum S^I(j,m)} \tag{8}$$

例如，假设用户 U_k 与目标用户 U_i 的相似度为 $S^U(i,k)=0.4$，项目 I_m 与目标项目 I_j 的相似度为 $S^I(j,m)=0.3$，且 $\lambda_1=0.6$，$\lambda_2=0.4$，则综合相似度由式 (7) 和式 (8) 可得 $S^{\rm SS}(i,j,k,m)=0.36$. 可见，综合相似度 $S^{\rm SS}(i,j,k,m)$ 大小介于 $S^U(i,k)$ 和 $S^I(j,m)$ 之间，平衡和协调了用户相似度和项目相似度，体现了用户和项目的双视角协同作用.

2.2　基于内外部相似度融合的数据稀疏缓解方法

基于双视角的评分预测虽然充分利用了相似用户和相似项目信息，但由于评分矩阵存在稀疏问题，若同时要求用户相似和项目相似，会导致相似用户和相似项目的评分数据会更少. 因此，直接实现该方法会导致该方法对数据稀疏性问题更加敏感. 由分析推荐过程可知，相似度计算是寻找相似用户和相似项目的重要环节，其结果直接影响评分预测. 为了解决数据稀疏导致相似度计算不准确的问题，本文提出基于内外部相似度融合的方法来缓解数据稀疏问题.

2.2.1　项目相似性度量

在用户项目评分矩阵中，传统项目间的相似度直接采用由用户评分形成的项目向量计算. 当评分矩阵稀疏且用户之间共同评分元素较少时，项目间的相似性难以准确计算. 实际中，项目间的相似性还受到项目属性的影响. 本文所讨论的项目属性用于说明项目特征，且不同项目属性特征各异. 因此，在度量项目相似性时，需要同时考虑项目的外因用户评分和内因项目特性的影响. 本文将外因产生的相似性称为外部相似性或相似度，可由式 (9) 计算；内因形成的

相似性称为内部相似性或相似度，可由式(10)计算. 即有

$$S^I_{\text{out}}(i,j)=\frac{\vec{I}_i\cdot\vec{I}_j}{||\vec{I}_i||||\vec{I}_j||} \tag{9}$$

$$S^I_{\text{in}}(i,j)=\sum_{k=1}\phi(k)S(\Theta(k),i,j) \tag{10}$$

式中，Θ 为项目特征属性集，$S(\Theta(k),i,j)$ 为项目 i 和项目 j 在属性 k 上的相似度，$\varphi(k)$ 为属性 k 在所有属性中的权重.

为了充分体现内外相似度在项目相似性中的作用，本文设计了一个基于局部稀疏因子的调节参数，即项目评分局部稀疏度. 现有文献中所使用的稀疏因子通常称为全局稀疏因子，其值等于所有未知评分元素数量与所有评分元素数量的比值. 本文的局部稀疏因子与通常的全局稀疏因子不同，它是从局部角度描述了任意两个项目之间共同评分集合的稀疏性.

定义1 项目评分局部稀疏度

设 U^I_i 为对项目 i 进行评分的所有用户形成的集合，U^I_j 为所有对项目 j 进行评分的用户集合，则项目评分局部稀疏度为

$$D^I_{i,j}=\frac{2|U^I_i\cup U^I_j|-(|U^I_i|+|U^I_j|)}{2|U^I_i\cup U^I_j|} \tag{11}$$

在项目评分局部稀疏度下，采用式(12)的Sigmoid函数生成内部相似度和外部相似度融合参数.

$$f(D^I_{i,j})=\begin{cases}\dfrac{1}{1+\mathrm{e}^{-D^I_{i,j}}}, & 0\leqslant D^I_{i,j}<1\\ 1, & D^I_{i,j}=1\end{cases} \tag{12}$$

式中，融合参数 $f(D^I_{i,j})$ 取值范围为0.5和1之间，这样保证最终融合相似度中始终包含项目内部相似度. 内部相似度刻画的是项目属性特征间的相似性，因而在项目相似性度量过程中内部相似度始终产生作用. 当局部稀疏度 $D^I_{i,j}$ 等于0时(极端理想情况下)，两个项目完全由相同的用户进行了评分，则融合 $f(D^I_{i,j})$ 正好等于0.5，也就是说内部相似度和外部相似度融合时权重相同；当局部稀疏度 $D^I_{i,j}$ 等于1时，意味着项目 i 和项目 j 之间没有共同用户评分，则融合参数 $f(D^I_{i,j})$ 为1，也就是说项目 i 和项目 j 间的相似度仅依赖于内部相似度. 项目融合相似度为

$$S^I(i,j)=f(D^I_{i,j})S^I_{\text{in}}(i,j)+(1-f(D^I_{i,j}))S^I_{\text{out}}(i,j) \tag{13}$$

项目融合相似度体现了项目之间内部因子和外部因子在度量项目相似性中的作用，有效解决了数据稀疏问题. 同时，当项目融合相似度仅依赖于内部因子即 $f(D^I_{i,j})=1$ 时，可解决项目冷启动问题. 融合参数 $f(D^I_{i,j})$ 控制和协调内部因子和外部因子在相似性度量中的作用，使得项目相似性度量更加合理.

2.2.2 用户相似性度量

用户兴趣偏好的相似性用于表达用户之间的相似性. 这种相似性恰好反映在用户评分信息中，故可根据用户评分信息之间的相似性来度量用户相似性. 当用户间共同评分数少，存在缺失值较多时，直接利用传统方法(如余弦相似度等)进行计算则误差很大，因此本文在项目融合相似度下先对用户缺失评分进行预评分，再进行相似性度量. 在基于项目融合相似度下

搜索未知评分项目的相似项目集进行预评分，进而利用定义 2 中的广义共同评分项目集来度量用户相似性.

定义 2　广义共同评分项目集

对任意用户 u_i 和 u_j，$I(u_i)$ 为用户 u_i 的评分项目集合，$I(u_j)$ 为用户 u_j 的评分项目集合，$I(u_i)$ 和 $I(u_j)$ 的交集 $I^{\cap}(u_i,u_j) = I(u_i) \cap I(u_j)$ 为用户 u_i 和 u_j 的共同评分项目集，并集 $I^{\cup}(u_i,u_j) = I(u_i) \cup I(u_j)$ 为用户 u_i 和 u_j 的广义共同评分项目集.

设 N_i 为用户 u_i 的在广义共同评分项目集 $I^{\cup}(u_i,u_j)$ 中未评分项目集合，则对任意 $I_p \in N_i$ 的预评分 $\widehat{r}(u_i,I_p)$ 按式 (14) 进行估计

$$\widehat{r}(u_i,I_p) = \overline{r}_{u_i} + \frac{\sum\limits_{I_k \in I'(u_i)} S^I(I_p,I_k)(r(u_i,I_k) - \overline{r}_{u_i})}{\sum\limits_{I_k \in I'(u_i)} S^I(I_p,I_k)} \tag{14}$$

$$I'(u_i) = \{I_k | S^I(I_p,I_k) > \eta, I_k \in I(u_i)\} \tag{15}$$

式中，$\overline{r}_{u_i}$ 为用户 u_i 的平均评分，$S^I(I_p,I_k)$ 为项目融合相似度，$\eta(\eta = 0.1)$ 为阈值参数[18]，根据式 (15) 可以在用户已评分项目集中筛选大于阈值的评分项目. 对每一个未知评分元素预评分后，任意用户 u_i 和 u_j 在广义共同评分项目集下，每个项目都有评分值，要么是已知评分值，要么是预评分值. 最后，用户 u_i 和 u_j 的相似性度量由式 (3) 计算.

基于项目融合相似度的预评分扩展了用户评分集，避免了全局搜索相似项目集，提高了用户之间相似性度量的准确性，缓解了用户评分稀疏性问题. 但该方法不能解决用户冷启动问题，除非利用额外的信息[19]，如社交网络中的用户信任关系或社交关系等，但这些信息在一些经典数据集如 MovieLens 中无法获取. 因此，本文直接采用文献 [16] 给出的用户平均评分或项目平均评分进行填补的方法来解决用户冷启动问题.

2.3　双视角协同推荐算法

双视角协同推荐方法建立在传统协同推荐方法的基础上，综合利用相似用户和相似项目信息进行协同推荐. 假设给定用户项目评分矩阵 R，包含 M 个用户和 N 个项目，采用双视角的协同推荐方法为用户 u_a 推荐 top-K 个项目，其算法如下：

算法 1　双视角协同推荐算法

输入：M, N, R, u_a, K

输出：top-K 的项目列表

S1：离线计算项目间的内部相似度和外部相似度，并进行融合得到项目融合相似度；

S2：度量用户 u_a 与其他用户间的相似性；

S2-1：对用户 u_i，获取 u_a 和 u_i 的广义共同评分项目集 $I^{\cup}(u_i,u_a)$；

S2-2：在 $I^{\cup}(u_i,u_a)$ 内对用户 u_a 和 u_i 的未评分项目分别根据项目融合相似度搜索相似项目进行预评分；

S2-3：根据式 (3) 度量用户相似性.

S3：对用户 u_a 和待预测评分的项目，根据用户相似性和项目相似性重排用户项目评分矩阵；

S4：根据用户相似度和项目相似度计算融合参数 λ_1 和 λ_2，得到综合相似度；

S5：用式 (5) 预测用户 u_a 对未评分项目的评分；

S6：重复执行 S3∼S5，预测各未知评分项目的评分；

S7：降序排列已预测评分的项目，为用户 u_a 推荐 top-K 个项目，算法结束.

3 实 验

为验证本文所提出的双视角协同过滤推荐方法的有效性，选取标准数据集MovieLens和公开数据集LDOS-CoMoDa[20]进行实验，两个数据集上取得了相似的实验结果，本文仅给出MovieLens数据集上的实验结果. MovieLens数据集包含943个用户，1 682部电影即项目，10万项评分数据，评分规模为1~5，全局稀疏度为0.936 95，每个用户至少评分了20部电影.

3.1 实验评价参数

由于双视角的协同推荐方法主要是预测目标用户对未知元素的评分，这里采用式(16)的经典的平均绝对误差(MAE)[21-22]作为评价指标. 该指标反应了预测评分与实际评分之间的平均差异.

$$\mathrm{MAE}=\frac{\sum\limits_{(u,i)\in R_{\mathrm{test}}}|r_{u,i}-\widehat{r}_{u,i}|}{|R_{\mathrm{test}}|} \tag{16}$$

式中，$|R_{\mathrm{test}}|$为测试集中元素数目，平均绝对误差越小，精度越优.

3.2 实验条件

在MovieLens数据集上进行了5种不同规模的实验，取其平均精度为评价指标. 在数据集中随机抽取5组用户，分别包含用户数为100、300、500、700、900，同时抽取与其相关的评分数据和项目. 5组实验数据分别用G1、G3、G5、G7、G9表示，统计信息如表1所示.

表 1 各组实验数据统计信息

Table 1 Statistics information of each group of data

组别	G1	G3	G5	G7	G9
用户	100	300	500	700	900
项目	1 227	1 447	1 563	1 652	1 678
已知评分数据	10 798	32 932	54 701	72 702	95 745
理想评分数据	122 700	414 300	781 500	1 156 400	1 510 200
全局稀疏度	0.912 0	0.920 5	0.930 0	0.937 1	0.936 6

MovieLens数据集中项目是电影，主要包含了项目类别特征信息. 虽然其他特征如情节、导演和演员等对推荐也有作用，但由于这些信息不包含在数据集中，即使有相关信息也需要其他处理方法辅助，如自然语言处理等. MovieLens数据集中的电影项目包括犯罪、科幻、喜剧、惊悚、灾难等20种，用于描述电影所属的题材和类型，刻画了电影的内部固有特征信息，同一部电影可能包含多种类别，这些类别信息即可用于度量其内部相似性. 由于一部电影存在多个类别，且类别之间有主次之分，因此根据这些电影在IMDB网站中提供的顺序进行了排序，序数反映了类别的重要性，采用式(17)的类高斯函数[23]进行量化

$$\mu(g_i,I_j)=r_i/2^{\sqrt{\alpha N_j(r_i-1)}} \tag{17}$$

式中，g_i为电影I_j的类别，N_j为类别数量，r_i为类别g_i的序数，且$1\leqslant r_i\leqslant N_j$. 若某个类别在$I_j$中未出现则以0代替. $\alpha>1$为调节参数，控制各类别在I_j中的差异，$\alpha=1.2$时效果最好[23].

3.3　实验项目 1

本实验主要验证项目融合相似度计算方法的有效性. 项目融合相似度在双视角协同推荐方法中起着很重要的作用，因此首先设计一个小规模的预备实验对其有效性进行验证. 从 MovieLens 中随机选择一半用户和相应的评分形成一个新的数据集，全局稀疏度为 0.932 44，该数据集训练集占 80%，测试集占 20%. 计算项目间的内部相似度和外部相似度得到项目融合相似度，以改进传统的基于项目和基于用户的协同推荐方法并与传统方法进行比较，结果如图 2 所示. 图 2 中 UBCF-I 和 IBCF-I 分别为应用项目融合相似度改进的基于用户和项目的方法. UBCF 和 IBCF 分别表示基于用户推荐的传统方法和基于项目推荐的传统方法. 显然，应用项目融合相似度后，两种协同过滤推荐方法的预测精度都得到了很大改善，充分说明了项目融合相似度增强了项目相似性和用户相似性的度量. 传统方法与改进方法之间预测精度差别较大，这是由于项目综合相似度不仅体现了项目内部相似性的本质特征，而且兼顾了用户评分所产生的项目外部相似性的作用.

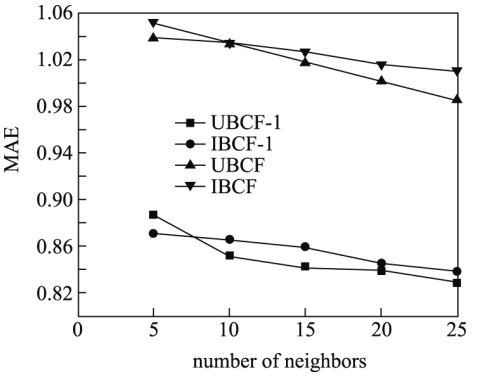


图 2　项目融合相似度下的协同推荐方法与传统协同推荐方法预测精度比较

Figure 2　MAE comparisons between traditional CFs and CFs improved by fusion similarity of item

3.4　实验项目 2

本实验目的是验证和评价双视角协同推荐方法与其他 3 种方法评分预测的精度. 在 G1、G3、G5、G7、G9 这 5 组数据集上利用双视角协同方法进行评分预测，并与传统的基于用户与基于项目的线性组合方法[6]、项目融合相似度下的基于用户的改进方法、基于项目的改进方法进行比较，实验结果如表 2 所示.

表 2　预测精度比较

Table 2　Comparisons of prediction accuracy

组	G1	G3	G5	G7	G9
双视角方法	**0.824**	**0.798**	**0.755**	**0.727**	**0.716**
基于用户的改进方法	0.847	0.838	0.763	0.749	0748
基于项目的改进方法	0.852	0.832	0.786	0.751	0.742
基于用户与基于项目的线性组合方法	0.848	0.829	0.768	0.747	0.753

根据表 2 数据可知，双视角协同推荐方法的预测精度在 5 组实验中均高于其他方法. 当用户数增大时，双视角方法的精度增幅更大，主要原因有两方面：1）双视角方法充分利用了相

似用户和相似项目信息；2) 项目融合相似度对用户相似性和项目相似性度量方法的改进. 项目融合相似度下，基于用户和项目的改进方法也具有较好的预测精度. 项目融合相似度不仅考虑了项目内部相似性而且还考虑了外部相似性. 由于基于用户和项目的线性组合方法是对单个传统方法预测结果的线性组合，单个方法仍然是单视角且未对其进行改进，因此预测精度相对较差.

3.5 实验项目3

本实验验证双视角协同推荐方法与其他几种典型推荐方法对评分数据集全局稀疏度的敏感性. 数据稀疏是推荐领域中不可避免的问题，因为绝大多数用户只能对很少的项目给出评分或评价. 在数据全局稀疏度非常低时，多数方法都能得到良好的推荐效果. 好的推荐方法应对数据全局稀疏度有很好的免疫能力，既使稀疏度很高，也能得到较好的推荐效果. 因此，本实验主要目的是测试在不同数据集全局稀疏度下双视角协同推荐方法对全局稀疏度的敏感性. 通过随机选择与循环逼近的方法选择了5种不同全局稀疏度区间的数据集进行实验. 首先根据稀疏度区间随机产生一定数量的用户数，然后根据用户信息抽取相应的评分数据，进而利用评分数据中涉及的项目信息提取项目集，最后计算数据集全局稀疏度是否满足区间要求. 若不满足则增加或减少一定数量的用户数重复选择过程，直到满足全局稀疏度要求. 最终获取5种不同全局稀疏度区间的数据集，其统计信息如表3所示. 在每个数据集上应用了包括双视角协同推荐方法在内的6种推荐方法 (如表4所示) 进行了10次折叠交叉验证实验，计算其平均绝对误差，结果如图3所示.

表 3 不同全局稀疏度的数据集统计信息

Table 3 Statistics information of dataset with varying global sparsity

全局稀疏度区间	0.4∼0.5	0.5∼0.6	0.6∼0.7	0.7∼0.8	0.8∼0.9
全局稀疏度	0.481 6	0.573 8	0.618 3	0.743 9	0.857 3
用户	137	278	315	552	621
项目	498	512	579	607	1 009
已知评分数	35 368	60 664	69 616	85 810	89 414

表 4 对比方法简称

Table 4 Abbreviations of methods of comparisons

简称	含义
UBCF	传统的基于用户的协同推荐方法
UBCF-I	项目综合相似度下基于用户的协同推荐方法
UIBCF-I	双视角的协同推荐方法
IBCF	传统的基于项目的协同推荐方法
IBCF-I	项目综合相似度下基于项目的协同推荐方法
UI-Linear	基于用户和项目的线性组合推荐方法

传统的协同推荐方法UBCF和IBCF对数据集的全局稀疏度很敏感，而其他4种改进的方法对全局稀疏度免疫力较强. 随着稀疏度增大，传统方法的评分预测精度越来越低，应用项

目融合相似度的改进方法和线性组合方法评分预测偏差不大. 在全局稀疏度较低时，用户具有较多数量的已知评分，4 种改进方法精度差别不大. 当全局稀疏度增大时，线性组合方法精度开始变差，而应用了项目融合相似度的 3 种方法精度略有提高，一方面原因是项目融合相似度发挥作用，另一方面原因是随着用户数的增多，利用项目融合相似度可以找到更多的高质量相似用户. 其中双视角的协同推荐方法表现最优. 显然，在项目融合相似度作用下，充分考虑相似用户和相似项目两方面的信息使得双视角协同推荐方法具有更强的全局稀疏度免疫力.

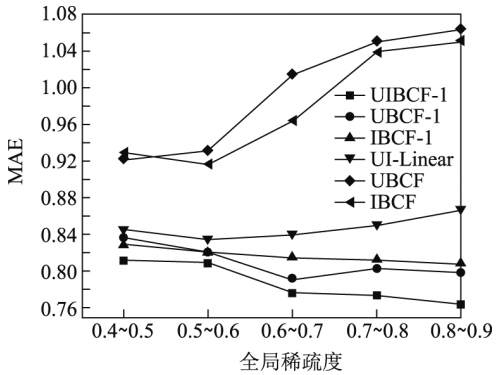


图 3　6 种方法对全局稀疏度敏感性的比较

Figure 3　Comparison of sensitivities to global sparsity of 6 methods

4　结　语

本文同时考虑用户和项目两个视角，利用二者内嵌的信息研究双视角的协同推荐方法，提出基于局部稀疏因子的项目融合相似度改进项目和用户的相似性度量方法. 在项目融合相似度下提出基于用户和项目的双视角协同推荐方法，并在标准数据集上进行了实验. 结果表明，应用项目融合相似度的双视角协同推荐方法很大程度上提高了评分预测精度，且对数据集的全局稀疏度具有较低敏感性，其综合性能优于其他几个典型的协同推荐方法. 接下来将进一步与更多的方法[24]进行实验比较，并考虑参数不同取值以及时间因素的影响，验证双视角协同推荐方法效果和性能.

参考文献:

[1] Ricci F, Rokach L, Shapira B. Introduction to recommender systems handbook [M]. New York: Springer, 2011.

[2] Shi Y, Larson M, Hanjalic A. Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges [J]. ACM Computing Surveys, 2014, 47(1): 1-45.

[3] Park D H, Kim H K, Choi I Y, Kim, J K. A literature review and classification of recommender systems research [J]. Expert Systems with Applications, 2012, 39(11): 10059-10072.

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

[5] 邓爱林，朱扬勇，施伯乐. 基于项目评分预测的协同过滤推荐算法 [J]. 软件学报，2003, 14(9): 1621-1628.
Deng A L, Zhu Y Y, Shi B L. A collaborative filtering recommendation algorithm based on item rating prediction [J]. Journal of Software, 2003, 14(9): 1621-1628 (in Chinese).

[6] Ma H, King I, Lü M R. Effective missing data prediction for collaborative filtering [C]//Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2007: 39-46.

[7] Lu Z, Dou Z, Lian J, Xie X, Yang Q. Content-based collaborative filtering for news topic recommendation [C]//29th AAAI Conference on Artificial Intelligence, 2015: 217-233.

[8] Song R P, Wang B, Huang G M, Liu Q D, Hu R J, Zhang R S. A hybrid recommender algorithm based on an improved similarity method [J]. Applied Mechanics and Materials, 2014, 475: 978-982.

[9] Moin A, Ignat C L. Hybrid weighting schemes for collaborative filtering [D]. Paris: INRIA Nancy, 2014.

[10] Goldberg D, Nichols D, Oki B M, Terry D. Using collaborative filtering to weave an information tapestry [J]. Communications of the ACM, 1992, 35(12): 61-70.

[11] 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.

[12] Deshpande M, Karypis G. Item-based top-n recommendation algorithms [J]. ACM Transactions on Information Systems, 2004, 22(1): 143-177.

[13] Rennie J D M, Srebro N. Fast maximum margin matrix factorization for collaborative prediction [C]//Proceedings of the 22nd International Conference on Machine Learning, 2005: 713-719.

[14] Choi K, Suh Y. A new similarity function for selecting neighbors for each target item in collaborative filtering [J]. Knowledge-Based Systems, 2013, 37: 146-153.

[15] Forsati R, Mahdavi M, Shamsfard M, Sarwat M. Matrix factorization with explicit trust and distrust relationships [J]. ArXiv: 1408. 0325 Ⅵ [cs.SI], 2014.

[16] Ghazanfar M A, Prugel A. The advantage of careful imputation sources in sparse data-environment of recommender systems: Generating improved svd-based recommendations [J]. Informatica, 2013, 37(1): 61-92.

[17] Anand D, Bharadwaj K K. Pruning trust-distrust network via reliability and risk estimates for quality recommendations [J]. Social Network Analysis and Mining, 2013, 3(1): 65-84.

[18] Agarwal V, Bharadwaj K K. A collaborative filtering framework for friends recommendation in social networks based on interaction intensity and adaptive user similarity [J]. Social Network Analysis and Mining, 2013, 3(3): 359-379.

[19] Li W, Ye Z, Xin M, Jin Q. Social recommendation based on trust and influence in SNS environments [J]. Multimedia Tools and Applications, 2015: 1-18.

[20] OdićA, Tkalčič, Tasič J F, Košir A. Predicting and detecting the relevant contextual information in a movie-recommender system [J]. Interacting with Computers, 2013, 25(1): 74-90.

[21] Shani G, Gunawardana A. Evaluating recommendation systems [M]. New York: Springer, 2011: 257-297.

[22] Yang X, Guo Y, Liu Y, Steck H. A survey of collaborative filtering based social recommender systems [J]. Computer Communications, 2014, 41: 1-10.

[23] Zenebe A, Zhou L, Norcio A F. User preferences discovery using fuzzy models [J]. Fuzzy Sets and Systems, 2010, 161(23): 3044-3063.

[24] Wang J, De Vries A P, Reinders M J T. Unifying user-based and item-based collaborative filtering approaches by similarity fusion [C]//Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2006: 501-508.

(编辑：王 雪)