基于用户行为与角色的协同过滤推荐算法①

李幼平[1,2]，尹柱平[2]

[1](桂林航天工业高等专科学校 党委，桂林 541004)
[2](桂林电子科技大学 商学院，桂林 541004)

摘 要：针对传统协同过滤推荐算法中以稀疏评分计算用户相似性可能并不准确的问题，提出以用户行为对应一定分值填补空缺的 I-U 评分矩阵，并以分角色下的权重系数 K 约束用户相似性计算的改进协同过滤推荐算法。实验表明，改进算法的推荐质量更高。

关键词：协同过滤；I-U 评分矩阵；相似性计算；用户行为；用户角色

Collaborative Filtering Recommendation Algorithm Based on Users' Behavior and Roles

LI You-Ping[1,2], YIN Zhu-Ping[2]

[1](Guilin College of Aerospace Technology, Guilin 541004, China)
[2](School of Business, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: There are sparse ratings problem in the traditional CF recommendation algorithm, and based on this sparse ratings will lead to the fact that the similarity may not be accurate. For this reason, a CF algorithm based on fixed I-U ratings matrix, which is given by a certain ratings of user behavior instead of vacancies rating, and weighted coefficient K bases on users' role to constrain the similarity calculation is proposed. Experiments show that the improved algorithm has better recommendation quality.

Key words: collaborative filtering; I-U score matrix; similarity calculation; users' behavior; users' role.

协同过滤推荐算法（collaborative filtering recommendation algorithms）是目前应用和研究最为广泛的推荐技术之一[1]，它根据用户对项的 I-U 评分矩阵来计算用户的相似度，从而产生目标用户的最邻近用户集，通过这一集合中用户对项的打分预测产生目标用户对未知项的打分，取预测打分最高的首选若干项作为结果推荐给用户。它可忽略项目对象本身的内容问题，能应用在可计算的文本领域和非结构化的电影、音乐和图书等复杂对象领域[2]。这一推荐如同现实中的口碑，不局限于用户原有的感兴趣内容，可发现用户可能感兴趣的新内容。

形成用户的最邻近用户集是协同过滤推荐算法中最为关键的一步[3]，但传统协同过滤推荐算法存在稀疏性问题，用户对资源项的打分非常稀疏，以稀疏的打分产生用户间的相似性可能并不准确，从而影响了算法的准确性。故本文提出了一种改进的协同过滤推荐算法。并通过验证实验表明本算法有更高精确度。

1 基于用户行为与角色的推荐算法

1.1 问题的提出

传统协同过滤推荐算法基于以下假设：如果用户对一定项的打分较相似，则他们对其他项的打分也较相似。它是以用户对项打分的 I-U 评分矩阵作为学习用户偏好并产生推荐的基础，但用户的打分常常十分稀疏。

事实上，用户对项的打分作为对资源项的偏好表达，但同时，用户的系统行为也是对偏好的表达。以用户的浏览、点击作为一个基本偏好的筛选过程。以用户的“收藏”操作表明对某一信息资源的关注，并存在针对这一资源的偏好。用户查询则是对自身偏好

① 收稿时间:2011-03-17;收到修改稿时间:2011-04-11

资源的明确查找。在系统中以日志方式记录用户行为，并以 1～4 的分值与上述用户行为对应，以此填补并修正 I-U 评分矩阵。

在得到用户对项的直接打分和通过用户行为填补评分数据后，协同过滤推荐算法将预测用户对未打分的未知项的打分，以此产生推荐。其中，较典型的是 slope one 算法[4]，它是以用户共同打分项的打分情况猜测用户对目标未打分项的打分：假定存在用户 X，Y 和 A 都已对 Item 1 打分，且用户 X，Y 也对 Item2 打分，如表 1：

表 1　用户评分表

Rating / User	Item 1	Item 2
X	5	3
Y	4	3
A	4	?

那么根据 Slope One 算法，用户 A 对 Item2 的打分为：4-[(5-3)+(4-3)]/2=2.5。

但在 slope one 算法中，默认了用户对 Item2 的打分倾向是相似甚至是相同的，显然，在评分数据稀疏时这种倾向并不存在。同时，slope one 算法容易陷入一个逻辑的死循环，计算用户的相似性并预测评分是根据相似用户的评分来计算，即默认了用户之间是相似的，是可允许通过其他任意用户的打分来预测评分。但如果 X、Y 和 A 之间的打分倾向是不一样的，这个式子就不成立了。包括加权 slope one 算法，仍依据这一默认的假设。

故本文认为，在计算用户相似性时，要考虑影响用户偏好或评分倾向的一些重要而稳定的属性。根据这种在用户自身明确而稳定的属性对用户群进行划分，由此产生的用户群称之为角色。因而同一角色下用户具有较好的关联和相似的偏好特征，从而具有相同的打分倾向。

以这一思想，定义约束权重系数 K 来计算用户相似度 sim。相同角色时，K 为 1；不同角色时，取用户之间评分的项的交集比例来作为 K 的值，。在多评分数据下，两用户即使角色不同，如果共同评分项较多，K 也越接近 1，同样可成为相似用户。则权重系数 K 的计算式为：

$$K=\begin{cases}1, \text{相同用户角色}\\ C_{ij}/\left(C_i+C_j\right), \text{不同用户角色}\end{cases} \quad (1)$$

其中，C_{ij} 为用户 i 和 j 共同评分项的交集，C_i+C_j 为用户已评分项的并集。

1.2 基于用户角色和行为的改进算法过程

传统协同过滤推荐算法过程有三个阶段[5]：建立 I-U 评分矩阵，计算用户相似度并产生目标用户的最邻近用户集，预测评分产生推荐项集。基于传统算法，改进算法的过程描述如下：

第一阶段：建立以用户行为填补的修正 I-U 评分矩阵。

统计用户对项的评分形成了 I-U 评分矩阵，它是计算用户相似性的重要数据源。在 m 个用户对 n 个项进行打分的基础上，以用户行为对应一定分值填补部分空缺项的打分，建立修正的 I-U 评分矩阵。

第二阶段：依靠用户角色，基于约束系数 K 的最邻近用户集的形成。

度量用户相似性的方法主要有余弦相似性、相关相似性、以及修正的余弦相似性三种。本文应用基于皮尔逊相关系数（Pearson Correlation）并能度量两组用户评分数据线性关系的相关相似性度量法[6]，具体计算公式为：

$$sim(i,j)=\sum_{I_a\in I_{ij}}(R_{ia}-\bar{R}_I)(R_{ja}-\bar{R}_J)/\sqrt{\sum_{I_a\in I_{ij}}(R_{ia}-\bar{R}_I)^2\times\sum_{I_a\in I_{ij}}(R_{ja}-\bar{R}_J)^2} \quad (2)$$

其中，I_a 为已打分项 a，I_{ij} 为用户 i 和 j 均已打分的项集，R_{ia} 表示用户 i 对项目 a 的打分，R_{ja} 表示用户 j 对项目 a 的评分，$\bar{R}_I$，$\bar{R}_J$ 表示用户 i 和 j 的平均评分。

计算任意用户之间相似约束系数 K 的值，并以式(2)计算用户相似性时加入约束系数 K 进行加权，以 KR_{ia} 代替 R_{ia}，以 KR_{ja} 代替 R_{ja}，得到式：

$$sim(i,j)=\sum_{I_a\in I_{ij}}(KR_{ia}-\bar{R}_I)(KR_{ja}-\bar{R}_J)/\sqrt{\sum_{I_a\in I_{ij}}(KR_{ia}-\bar{R}_I)^2\times\sum_{I_a\in I_{ij}}(KR_{ja}-\bar{R}_J)^2} \quad (3)$$

计算得到用户相关系数后，设定一个阈值 θ，把满足 $sim(i,j)>\theta$ 的相似度以 Top-N 排序，产生最邻近用户集 V。

第三阶段：生成预测评分和推荐项。

由集 V 中用户对当前项 i 的打分，采用平均加权法来计算预测打分并产生推荐结果，则用户 u 对目标项目 i 的预测打分 P_{ui} 计算式为[7]：

$$P_{ui}=\overline{R_u}+[\sum_{v\in V}sim(u,v)\times(R_{vi}-\overline{R_v})/\sum_{v\in V}|sim(u,v)|] \quad (4)$$

其中，$\overline{R_u}$ 为用户 u 对所有项打分的均值，$sim(u,v)$ 表示用户 u，v 的相似度，R_{vi} 表示最近邻用户集中的用户 v 对项 i 的打分，$\overline{R_v}$ 为项 v 的平均打分。

对 P_{ui} 进行 Top-N 排序，推荐前 N 项给用户。

2 改进的CF算法描述

输入：m 个用户对 n 个项的打分，用户行为产生对项的填补打分，I-U 评分矩阵；分角色用户相似度 sim，用户评分项的并集和共同评分项的交集，相似阈值 θ，推荐首选项数 N。

输出：用户的 Top-N 推荐。

算法过程：

1) 根据用户行为以相应分值修正用户的 I-U 评分矩阵。

2) 计算任意用户之间相似约束系数 K 的值，根据式(3)计算用户 i 和 j 的相似度 $sim(i, j)$，比较 $sim(i, j)$ 与 θ 大小，把满足 $sim(i, j) > \theta$ 的相似度以 Top-N 排序，产生最邻近用户集 V。

3) 由集 V 中用户对当前项 i 的评分，根据式(4)计算目标用户 u 对项 i 的预测评分 P_{ui}。

4) 对 P_{ui} 进行 Top-N 排序，推荐前 N 项给用户。

3 验证实验结果

3.1 测试数据集

从 http://www.grouplens.org/node/73 下载 Movie Lens 提供的公共测试数据集 million-ml-data 来验证算法的有效性。该数据集由 6040 个用户对 3900 部电影评价打分产生的约 100 万个评分数据，它含有三个数据表，分别为用户信息表（users）、电影信息表（movies）和评分信息表（ratings）。其中用户信息表包含角色划分的职业类型属性。

为了测试本文提出的基于角色划分的协同过滤推荐算法的有效性，实验共进行两次，分别选择用户信息表中以职业(Occupation)代号为 7 的行政管理者(executive/managerial)和 17 的技术工程师(technician/engineer)的所有用户进行相关数据测试。虽然通过用户的职业属性在一定程度上缩小了测试数据集，但分析在传统算法下的推荐数据情况，满意的推荐结果较集中在相同背景下的用户之间，因而是不影响检验算法的有效性。另值得指出的是，划分用户集能有效降低计算的时耗。

在进行测试实验时，要进行数据的稀疏度计算[8]，低稀疏度的数据产生的预测评分才是可靠的。

$$\text{稀疏度}=1-\text{所有已评分值的项数}/(\text{用户数}\times\text{资源数}) \quad (5)$$

3.2 推荐算法质量的度量指标

本文采用常用的统计精度度量方法中的平均绝对误差 MAE(Mean Absolute Error)作为度量指标。当 MAE 值越小，推荐的质量越好。MAE 是用于计算推荐的预测评分值和用户真实评分值之间的偏离程度[9]，它的计算公式为：

$$MAE=[\sum_{i=1}^{n}|p_i-q_i|]/n \quad (6)$$

其中，p_i 为用户的预测评分值，q_i 为用户的实际评分值，n 为预测项的个数。

3.3 实验结果

通过 Excel 表格和 Access 数据库对数据进行预处理，把数据以八二比例分为两部分，80%部分作为训练 base 集，20%部分作为测试 test 集。预处理后数据样本的格式如下：

A	B	C	D	E
Occupation	usrid	movid	rating	Timestamp
10	1	1193	5	978300760
10	1	661	3	978302109
10	1	914	3	978301968
10	1	3408	4	978300275
10	1	2355	5	978824291
10	1	1197	3	978302268
10	1	1287	5	978302039
10	1	2804	5	978300719
10	1	594	4	978302268
10	1	919	4	978301368
10	1	595	5	978824268

图 1 预处理后数据样本的格式

通过数据预处理和统计得到两个测试数据集：1) 职业代码 7 下的 679 个用户，涉及 3276 部电影，一共有 104561 个评分数据，数据稀疏度为 4.7%。其中用户最少已评电影 20 部，最多 1521 部。2) 职业代码 17 下的 502 个用户，涉及 3196 部电影，共 72816 个评分数据，数据稀疏度为 4.5%。其中用户最少已评电影 20 部，最多 1595 部。并进一步统计各用户的已评项数 n 和用户的评分均值 $\overline{R}$。

对测试数据集 1 和 2 分别进行预测评分的计算并计算 MAE 值，与传统 CF 算法下的 MAE 值比较，得到结果如图 2 和图 3。

两个实验均选取最近邻用户集为 10 到 35 个邻居，间隔为 5，从图 2 与图 3 可看出本文提出的改进协同过滤推荐算法具有较小的 MAE 值，具有更高的推荐质量。

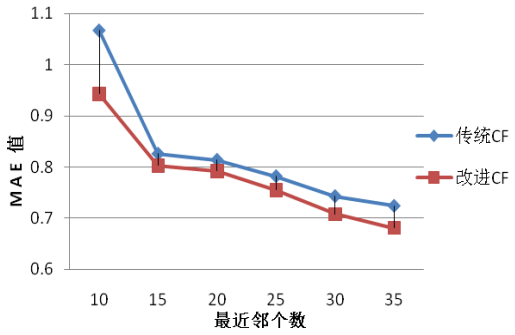

图 2　由测试集 1 得到的 MAE 值对比

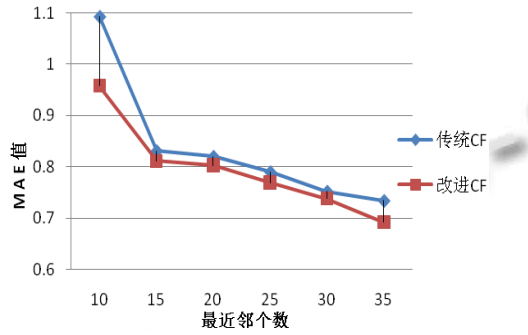

图 3　由测试集 2 得到的 MAE 值对比

4　结论

与传统协同过滤推荐算法比较，本文提出的改进协同过滤推荐算法最大不同的是：从评分数据稀疏问题和产生最近邻的相似性计算出发，讨论了 I-U 评分矩阵的修正问题和仅依靠用户打分是否能成为最近邻的问题，并提出了改进的方法，以用户行为对应分值替代的方式修正了 I-U 评分矩阵并以权重系数 K 约束最近邻用户的计算，使得推荐算法能更好的产生最邻近用户集。实验结果表明，本算法较显著地提高了推荐质量。

参考文献

1 王茜,杨莉云,杨德礼.面向用户偏好的属性值评分分布协同过滤算法.系统工程学报,2010,(4):561−568.

2 黄晓斌.基于协同过滤的数字图书馆推荐系统研究.大学图书馆学报,2006,(1):53−57

3 Kin BM, Li Q, Park CS, et al. A new approach for combining content-based and collaborative filters. Journalof Intelligent Information Systems, 2006,27:79−91.

4 周敏,周继鹏,丁光华.PSL:针对大规模数据应用的并行 Slope One 算法.科学技术与工程,2010,3:711−714.

5 Tian W, Xu J, Pend YQ. CF Improvement Based on Probabilistic Analysis of Discrete Explicit Rating Vector. Proc. of the 2009 First IEEE International Conference on Information Science and Engineering, 2009,9:814−816.

6 Sarwar B, Karypis G, Konstan J, et al. Item-Based Collaborative Filtering Recommendation Algorithms. Proc of the 10th Int'l World Wide Web Conf, 2001,285−295.

7 Breese J, Heckerman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Proc. of the 14th Conference on Uncertainty in Artificial Intelligent. 1998, 43−52.

8 Herlocker JL, Konstan JA, Terveen LG. Evaluating Collaborative Filtering Recommendation System. ACM Trans. on Information System, 2004,22(1):5−53.

9 冯克鹏.基于协同过滤的数字图书馆推荐系统研究.软件导刊,2010,5:16−18.

(上接第 102 页)

pcs. http://www. microsoft. com/whdc /winhec/pres06.mspx. 2006.

3 Panabaker R. Hybrid Hard Disk & ReadyDriveTM technology: Improving performance and power for Windows Vista Mobile PCs. WinHEC, 2006.

4 Kim YJ, Kwon KT, Kim J. Energy-effcient fle placement techniques for heterogeneous mobile storage systems. EMSOFT Conference, 2006.

5 Microsoft. ReadyDrive and hybrid disk. http://www. microsoft.com/whdc/system/sysperf/perfaccel.mspx. 2010.

6 Payer H, Sanvido MAA. Combo Drive: Optimizing Cost and Performance in a Heterogeneous Storage Device. Workshop on Integrating Solid-state Memory into the Storage Hierarchy, 2009.

7 http://traces.cs.umass.edu/index.php/Storage/Storage. 2007.