

一种基于情景的协同过滤推荐算法

李 荟, 谢 强, 丁秋林

(南京航空航天大学 计算机科学与技术学院, 江苏 南京 210016)

摘 要: 针对协同过滤推荐算法中数据极端稀疏所带来的推荐精度低下的问题, 文中提出一种基于情景的协同过滤推荐算法。通过引入项目情景相似度的概念, 基于项目情景相似度改进了用户之间相似度的计算公式, 并将此方法应用至用户离线聚类过程中, 最终利用用户聚类矩阵和用户评分数据产生在线推荐。实验结果表明, 该算法能够在数据稀疏的情况下定位目标用户的最近邻, 一定程度上缓解数据极端稀疏性引起的问题, 并减少系统在线推荐的时间。

关键词: 推荐系统; 情景; 协同过滤; 稀疏性; 聚类

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2014)10-0042-05

doi: 10.3969/j.issn.1673-629X.2014.10.010

A Collaborative Filtering Recommendation Algorithm Based on Scenario

LI Hui, XIE Qiang, DING Qiu-lin

(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics,
Nanjing 210016, China)

Abstract: The extremely sparse data in collaborative filtering recommendation algorithm often causes the decline of recommendation's precision. In order to solve the problem, suggest a new collaborative filtering recommendation algorithm based on scenario. By introducing the concept of item's scenario-similarity, modify the formula of similarity between users is improved based on item scenario similarity, and then the method has been applied in the procedure of user-clustering in the offline phase and recommendation in the online phase is produced with user-clustering matrix and user evaluation data. The experimental results show that the algorithm can locate the nearest neighbor for target in the condition of sparse data, alleviating some sparsity problem and reducing the time of recommendation in the on-line phase.

Key words: recommendation system; scenario; collaborative filtering; sparsity; clustering

1 概 述

随着互联网行业的高速发展, 遍布于其中的各式各样的网络资源也呈现出惊人的增长速度, 如何在众多的资源中方便快捷地获取所需的信息引起了人们越来越多的关注, 在这样的背景下, 推荐系统应运而生。推荐系统最早应用的领域是电子商务, 特别是自 20 世纪 90 年代以来, 在各国政府对电子商务的大力推动下, 推荐系统显示出了巨大的商业价值。推荐系统拥有能够帮助将浏览者转变为购买者和提高交叉销售能力等优势^[1], 帮助人们挑选感兴趣或者有价值的书籍^[2]、网页、电影、音乐^[3]等等。故目前几乎所有的电

子商务系统都部署了不同形式的推荐系统, 如 Amazon、eBay、Alibaba 等。

为了实现快速而精准的推荐, 研究者们提出了多种推荐方法, 如基于内容的推荐方法、协同过滤推荐方法、混合推荐方法等等。其中, 协同过滤推荐方法是最成功也是应用面最广泛的技术之一^[4], 该方法的出现大大促进了整个推荐领域的发展。

目前大部分的协同过滤推荐方法都是通过对某一用户未评分的项目进行预测评分并以此作为主要依据向该用户进行推荐。传统的基于用户协同过滤推荐算法将预测评分的基础建立在与目标用户相似的其他真

收稿日期: 2013-11-22

修回日期: 2014-03-03

网络出版时间: 2014-07-28

基金项目: 江苏省产学研联合创新资金项目 (SBY201320423)

作者简介: 李 荟 (1989-), 男, 硕士研究生, 研究方向为知识工程、信息集成; 谢 强, 博士, 副教授, 研究方向为信息安全、人机交互; 丁秋林, 博士, 教授, 博导, 研究方向为企业信息化、信息系统集成。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140728.1222.011.html>

实评分记录之上,这些相似用户往往指的是兴趣和口味相似的用户^[5],因此这些相似的用户在给相同的项目进行评分的时候往往会给出相似的结果。

但是协同过滤推荐算法在实际应用的过程中还存在一些问题,如稀疏性、冷启动、扩展性问题等^[6]。为了进一步提高基于用户协同过滤推荐质量,越来越多的学者们对此提出改进。文献[7]中将用户-项目评分矩阵转换为用户-类别矩阵以此来降低评分矩阵的稀疏性。文献[8]中把相似度支持度作为最近邻度量,并提出了能在维持推荐精确度前提下的 k 近邻策略。文献[9]中将用户与用户间的信任关系等加以利用,并在此基础上辅助产生推荐,也在一定程度上解决了稀疏性问题。文献[10]中利用用户对不同项目类型的多个评分相似度来计算用户对未评分项目的预测评分。文献[11]中利用用户对项目的关注的相似性来修正原始相似性计算,综合考虑用户对项目的关注和用户评价对推荐的影响。

传统的用户相似度算法未能考虑到项目与项目之间的内在联系,故文中提出将项目的使用情景也作为用户相似度计算的一部分,并以此展开进行用户聚类。实验证明,该改良算法能够有效地克服传统协同过滤推荐方法的数据稀疏导致的推荐质量下降问题,提高了推荐的准确度。

2 基于项目情景的用户相似度

2.1 传统的用户相似度

协同过滤算法的思路是基于用户的观点或评分对目标用户产生推荐列表,如果一个用户群对某些项目的评分比较相似,则认为他们的喜好是相似的,因此他们对其他项目的评分也会相似。假设推荐数据用三元组表示为

定义 1:数据源 $T = (U, I, R)$ 。

其中, $U = \{u_1, u_2, \dots, u_m\}$ 表示用户集合; $I = \{i_1, i_2, \dots, i_n\}$ 表示项目集合; $R = R(m, n)$ 表示用户 - 项目评分矩阵,寻找目标用户的最近邻正是通过用户 - 项目评分矩阵计算而来,如图 1 所示。

	i_1	...	i_k	...	i_n
u_1	$r(1,1)$...	$r(1,k)$...	$r(1,n)$
...
u_j	$r(j,1)$...	$r(j,k)$...	$r(j,n)$
...
u_m	$r(m,1)$...	$r(m,k)$...	$r(m,n)$

图 1 用户评分矩阵图

用 u, v 表示用户, $\text{sim}(u, v)$ 表示用户相似性。传统的用户相似度计算一般分两种:

(1) 余弦相似性。

$$\text{sim}(u, v) = \cos(\vec{u}, \vec{v}) = \frac{\sum_{i \in I_{u,v}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in I_u} r_{u,i}^2} \sqrt{\sum_{i \in I_v} r_{v,i}^2}} \tag{1}$$

每个用户的所有项目评分被看作 n 维项目空间里的一个向量,用户间的相似度可以用余弦夹角来度量。 I_u, I_v 表示用户 u, v 的评分项目集合。

(2) 相关相似性。

$$\text{sim}(u, v) = \frac{\sum_{c \in I_{u,v}} (r_{u,c} - \bar{r}_u)(r_{v,c} - \bar{r}_v)}{\sqrt{\sum_{c \in I_u} (r_{u,c} - \bar{r}_u)^2} \sqrt{\sum_{c \in I_v} (r_{v,c} - \bar{r}_v)^2}} \tag{2}$$

相关相似性根据用户共同的评分的项目评分记录来计算 Pearson 相关系数,并以此作为用户与用户之间的相似度。用户 u, v 共同评分的项目集合用 $I_{u,v}$ 表示; $r_{u,c}$ 和 $r_{v,c}$ 表示用户 u 和 v 对项目 c 的评分; \bar{r}_u 和 \bar{r}_v 分别表示用户 u 和 v 对项目的平均评分。

2.2 用户项目情景相似度

传统的协同过滤算法仅靠单一的共同评分为基础来计算用户相似度并不是很合理。因为这个单一的项目评分相似度是基于用户对项目的历史评分记录计算得出,并没有考虑到项目之间的内在联系,所以传统的相似度计算只能反映出用户评分记录的相似性,而不能很好地判断用户真实的喜好相似度,所以在一定程度上降低了推荐质量。

情景信息指的是实体的行为或时间事件的发生影响的上下文信息或是应用场景。情景与情景之间的联系很大程度上反映了项目与项目之间的联系。把情景内容集合表示为:

定义 2: $I_S = \bigcup_{i=1}^k S_i$ 表示情景中的项目集合,其中, $S_i = \{i_{i1}, i_{i2}, \dots, i_{in_i}\}, \dots, S_n = \{i_{n1}, i_{n2}, \dots, i_{nn_i}\}$ 。

S_n 表示第 n 个情景中包含的项目集合。例如,假设有用户 u 和 v ,没有共同评分项目, u 的项目评分数据稀疏,其中只包含项目 i_1, i_2 , v 的项目评分稀疏,只有 i_3, i_4 ,如图 2 所示。

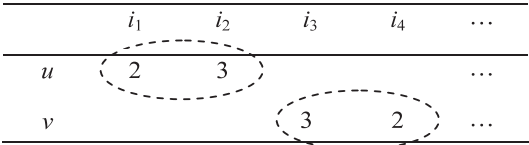


图 2 用户 u, v 无共同评分项目示意图

这时,如果按照传统的用户相似性算法计算,用户 u, v 之间是没有相似性的,若 u, v 的评分项目情景相似度很高,那么说明 u, v 互相之间是有推荐能力的,传统的用户相似度计算方法是不准确的。假设把情景节点用树形结构模型来表示,如图 3 所示。

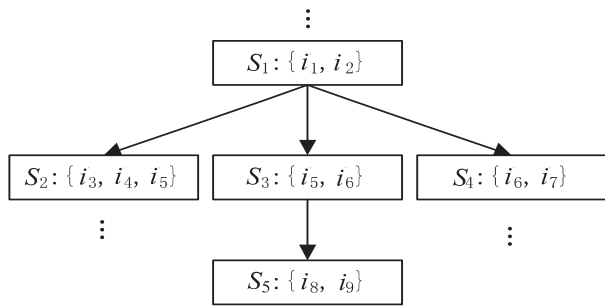


图 3 情景节点树形图

在此,提出一种项目情景相似度计算方法。

定义 3: NPS (Nearest Path Scenario, 最近路径情景)。NPS(i_j, i_k) = (S_m, S_n) 表示分别包含项目 i_j 和 i_k 的最近情景节点。例如: NPS(i_8, i_6) = (S_5, S_3)。

定义 4: LCA_I(S_m, S_n) 表示在情景结构中, 节点 S_m 与 S_n 的最近公共祖先 (Least Common Ancestors)。

至此, 项目 i_j 和 i_k 的情景相似度的计算公式为:

$$\text{sim}_s(i_j, i_k) = \frac{2\text{depth}(\text{LCA}_I(\text{NPS}(i_j, i_k)))}{\text{depth}(S_m) + \text{depth}(S_n)} \quad (3)$$

由公式可知, $\text{sim}_s(i_j, i_k) \in [0, 1]$ 。任意两个项目的情景相似度均可由公式(3)得出。

2.3 改进的用户相似度算法

传统的用户相似性计算过分依赖于用户项目评分数据, 在用户项目评分数据过于稀疏或者用户之间共同评分交集很少的情况下, 不能有效度量用户间的相似程度, 从而影响推荐质量。为此, 文中给出一种改进的结合项目情景相似度的方法来计算用户相似度。

设用户 u 和用户 v 评分项目数量为 m 和 n , 用户 u 和 v 的兴趣模型为 $I_u = \{(p_i, s(p_i)) \mid 1 \leq i \leq m\}$ 、 $I_v = \{(q_j, s(q_j)) \mid 1 \leq j \leq n\}$ 。其中, p_i 和 $s(p_i)$ 表示用户 u 的评分项目及评分值, q_j 和 $s(q_j)$ 表示用户 v 的评分项目及评分值。若将用户的兴趣模型映射到向量空间上, 则可以得出用户 u 和用户 v 的兴趣特征向量:

$$I_u = \sum_{i=1}^m (s(p_i) - \overline{s(p)}) \mathbf{P}_i \quad (4)$$

$$I_v = \sum_{j=1}^n (s(q_j) - \overline{s(q)}) \mathbf{Q}_j \quad (5)$$

式中, $\overline{s(p)}$, $\overline{s(q)}$ 表示用户 u, v 对项目的平均评分; $\mathbf{P}_i, \mathbf{Q}_j$ 分别表示项目 p_i 和 q_j 的评分向量。

根据传统的相似度计算方法, 当 $p_i \neq q_j$ 时, $\mathbf{P}_i \cdot \mathbf{Q}_j$ 为正交向量, $\mathbf{P}_i \cdot \mathbf{Q}_j = 0$; 当 $p_i = q_j$ 时, $\mathbf{P}_i \cdot \mathbf{Q}_j = 1$ 。根据公式(3), 有 $\mathbf{P}_i \cdot \mathbf{Q}_j = \text{sim}_s(i_j, i_k)$ 。由此可得:

$$I_u \cdot I_v = \sum_{j=1}^m \sum_{k=1}^n (s(p_j) - \overline{s(p)}) (s(q_k) - \overline{s(q)}) \cdot \text{sim}_s(i_j, i_k) \quad (6)$$

$$I_u \cdot I_u = \sum_{j=1}^m \sum_{k=1}^n (s(p_j) - \overline{s(p)}) (s(q_k) - \overline{s(q)}) \cdot \text{sim}_s(i_j, i_k) \quad (7)$$

所以当 $p_i \neq q_j$ 时, 即用户 u, v 之间没有共同评分项目的情况下, 依然可以基于项目间的情景相似度计算出非零用户相似度:

$$\text{sim}(u, v) = \frac{I_u \cdot I_v}{\sqrt{I_u \cdot I_u} \sqrt{I_v \cdot I_v}} \quad (8)$$

3 基于情景的协同过滤推荐

传统的基于用户的协同过滤算法在海量数据中寻找目标用户近邻的过程的执行效率日益低下, 为了提高在线执行效率, 文中的协同过滤算法将分为离线聚类类和在线推荐两个部分, 从而达到缩小在线查找目标用户近邻的搜索空间, 节省在线推荐时处理数据的时间, 提高系统推送效率的目的。

3.1 离线用户聚类

目前有很多聚类方法, 文中选取一种应用比较广泛的 k -means 聚类思想对数据源中的用户进行聚类^[12]。给出数据源 T 并给出给出聚类数目 k , 经过 k -means 算法计算得出用户聚类矩阵, 具体算法如下。

算法 1: 离线用户聚类算法。

输入: 数据源 $T(U, I, R)$ 和聚类数目 k ;

输出: 用户聚类矩阵 $\mathbf{D}(m, k)$ 。

(1) 预处理产生 k 个原始聚类中心;

(2) repeat

(3) for each u_i in U

(4) 计算 u_i 与各聚类中心的 $\text{sim}(u_i, 1), \text{sim}(u_i, 2), \dots, \text{sim}(u_i, k)$

(5) if ($\max(\text{sim}(u_i, 1), \text{sim}(u_i, 2), \dots, \text{sim}(u_i, k)) > \text{阈值}$)

(6) 将 u_i 划为该聚类;

(7) 更改聚类中心用户, 重新计算聚类中心;

(8) until 收敛, 聚类中心不再发生明显变化。

$\mathbf{D}(m, k)$ 为用户聚类矩阵, 其中元素 d_{ij} 表示用户 u_i 对聚类中心 j 的 $\text{sim}(u_i, j)$, 计算方法见公式(8)。

3.2 在线寻找近邻

在 3.1 节中离线聚类的基础上, 在线推荐算法首先计算待推荐用户与各聚类中心的相似度, 然后与用户聚类矩阵 $\mathbf{D}(m, k)$ 做匹配计算, 从而确定待推荐用户的 TOP- N 最近邻。

算法 2: 在线确定 TOP- N 最近邻。

输入: 带推荐用户 u_i , 用户聚类矩阵 $\mathbf{D}(m, k)$;

输出: TOP- N 最近邻。

(1) 计算待推荐用户 u_i 与聚类中心的相似度, 获得相似度向量 $\vec{s} = (\text{sim}(u_i, 1), \text{sim}(u_i, 2), \dots, \text{sim}(u_i, k))$;

(2) 计算相似度向量 \vec{s} 与 $\mathbf{D}(m, k)$ 中各个用户的欧式距离;

(3) 在步骤(2)中计算得出的欧式距离中选取最小的前 N 个用户作为最近邻。

3.3 产生推荐

在算法 2 中得出带推荐用户的 TOP- N 个最近邻后,组成目标推荐用户的最近邻居集 $N_u = \{u_1, u_2, \dots, u_n\}$,以相似度降序排列,再根据最近邻居集用户的项目评分情况对待推荐用户进行预测评分计算。待推荐用户 u 对项目实例 i 的预测评分 $P_{u,i}$ 计算公式如下:

$$P_{u,i} = \bar{r}_u + \frac{\sum_{v \in N_u} \text{sim}(u, v) \times (r_{v,i} - \bar{r}_v)}{\sum_{v \in N_u} |\text{sim}(u, v)|} \quad (9)$$

其中, \bar{r}_u 和 \bar{r}_v 表示用户 u 和用户 v 对资源项目的平均评分; $r_{v,i}$ 表示用户 v 对资源项目 i 的评分。

4 实验结果及分析

4.1 实验数据集

由于文中情景用户相似度的计算需要结合实际的应用背景,故采用某航空研究所的知识共享平台的全所区日志数据作为实验数据集。对数据进行预处理后从中选择 68 395 条评分数据作为实验数据集,该数据集包括 926 个用户和 1 745 个资源项目,并将其中 80% 作为训练集,其余作测试集。为了验证算法在用户评分矩阵稀疏情况下的运行效率,引入稀疏等级的概念^[13]。稀疏等级 ψ 的含义是用户评分矩阵中未评分的项目在整体数据集中所占的比例。文中所取的实验数据集的稀疏等级为:

$$\psi = 1 - \frac{68\ 395}{926 \times 1\ 745} = 95.767\%$$

可见用户评分矩阵还是相当稀疏的。

4.2 评估标准

实验中采用广泛使用的平均绝对偏差 (Mean Absolute Error, MAE) 作为推荐算法准确性的评估标准。MAE 通过计算预测用户评分与实际用户评分之间的偏差来衡量算法的准确性,MAE 值越小,算法推荐质量越高^[14]。假设算法对 N 个资源项目的预测评分向量为 $\{p_1, p_2, \dots, p_N\}$,对应用户的实际评分向量为 $\{r_1, r_2, \dots, r_N\}$,则 MAE 的计算公式如下:

$$\text{MAE} = \frac{\sum_{i=1}^N |p_i - r_i|}{N} \quad (10)$$

4.3 实验结果

为了测试文中算法 (Proposed CFR) 的推荐效率和执行效率,将传统的相关相似协同过滤推荐算法 (Corr-based CFR) 和余弦相似性协同过滤推荐算法 (Cosine-based CFR) 与文中算法做对比。

(1) 聚类数目对算法的影响。

固定用户最近邻居数,聚类数目从 10 到 50 递增,间隔为 10,计算 Proposed CFR 算法的 MAE,实验结果如图 4 所示

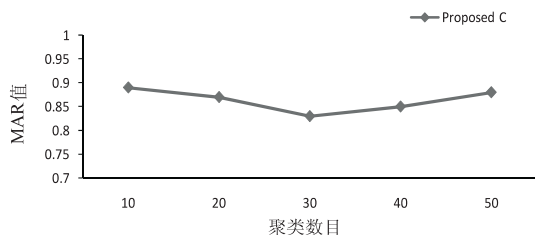


图 4 Proposed CFR 算法的 MAE

由图 4 可知,在用户邻居数固定为 $n = 40$ 的情况下,聚类数目的变化对整体的推荐精度影响不明显,当聚类数目 k 选取 30 左右的情况下,推荐效率较好。

(2) 与传统算法推荐精度的对比。

固定聚类数目 $k = 30$,最近邻居数从 10 到 60 递增,间隔为 10,分别计算 Proposed CFR 与 Corr-based CFR、Cosine-based CFR 并做对比,结果如图 5 所示。

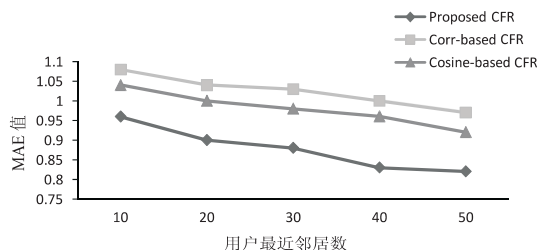


图 5 与传统算法对比图

由图 5 可知,文中提出的算法在最近邻居数递增过程中,均比传统的推荐算法有着更高的推荐精度。

(3) 在线推荐时间对比。

将分别应用 Proposed CFR 和 Corr-based CFR 思想的推荐系统做实时推荐时间的检测,最近邻居数从 10 递增到 50,间隔为 10,分别计算两者推荐时间,实验结果如图 6 所示。

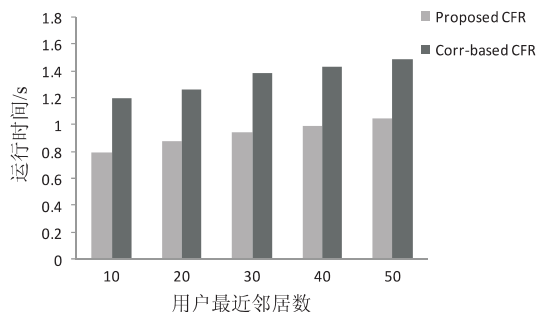


图 6 实时推送时间对比图

由图 6 可知,在事先对用户进行离线聚类的前提下,在线推荐的时间会大大缩短。

4.4 实验结论

通过 4.3 节的实验结果可知,文中提出的改进算法思想能够在用户评分矩阵稀疏的情况下,体现出较好的推荐精度,且经过离线聚类之后,使在线寻找用户

最近邻居的范围和时间缩短,大大降低了在线实时推荐的执行时间。

5 结束语

针对传统过滤协同算法在推荐精度和稀疏性方面所存在的一些问题,提出了一种基于情景的协同过滤推荐改进算法。实验结果表明该算法能在一定程度上提高推荐精度和推荐时间,目前还存在一些改进空间,比如在结合用户个性化方面还需要做进一步的研究。

参考文献:

- [1] Schafer J B, Konstan J A, Riedl J. E-commerce recommendation applications[J]. Data Mining and Knowledge Discovery, 2001, 5(1-2): 115-153.
- [2] Su Xiaoyun, Khoshgoftaar T M. A survey of collaborative filtering techniques[J]. Advances in Artificial Intelligence, 2009, 2009: 421-425.
- [3] Linden G, Smith B, York J. Amazon.com recommendations: item-to-item collaborative filtering[J]. IEEE Internet Computing, 2003, 7(1): 76-80.
- [4] 游文, 叶水生. 电子商务推荐系统中的协同过滤推荐[J]. 计算机技术与发展, 2006, 16(9): 70-72.
- [5] 许海玲, 吴潇, 李晓东, 等. 互联网推荐系统比较研究[J]. 软件学报, 2009, 20(2): 350-362.
- [6] Adomavicius G, Tuzhilin A. Towards the next generation of

recommender systems: a survey of the state-of-the-art and possible extensions[J]. IEEE Trans on Knowledge and Data Engineering, 2005, 17(6): 734-749.

- [7] Zeng Chun, Xing Chunxiao, Zhou Lizhu. Similarity measure and instance selection for collaborative filtering[C]//Proc of the 12th international conference on World Wide Web. New York, NY: ACM Press, 2003: 652-658.
- [8] 罗辛, 欧阳元新, 熊璋, 等. 通过相似度支持度优化基于 K 近邻的协同过滤算法[J]. 计算机学报, 2010, 33(8): 1437-1445.
- [9] Pitsilis G, Knapklog S J. Social trust as a solution to address sparsity-inherent problems of recommender systems[C]//Proc of ACM recommender system workshop on recommender system & the social web. [s. l.]: [s. n.], 2009: 33-40.
- [10] 范波, 程久军. 用户间多相似度协同过滤推荐算法[J]. 计算机科学, 2012, 39(1): 23-26.
- [11] 查文琴, 梁昌勇, 曹镭. 基于用户聚类的协同过滤推荐方法[J]. 计算机技术与发展, 2009, 19(6): 69-71.
- [12] 毛国君. 数据挖掘原理与算法[M]. 北京: 清华大学出版社, 2007.
- [13] 陈健, 印鉴. 基于影响集的协作过滤推荐算法[J]. 软件学报, 2007, 18(7): 1685-1694.
- [14] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithm[C]//Proc of the 10th international World Wide Web conference. New York: ACM Press, 2001: 285-295.

(上接第 41 页)

参考文献:

- [1] Wang Xiaofeng, Feng Weijia, Liu Qingjie, et al. Calibration research on fish-eye lens[C]//Proc of 2010 IEEE international conference on information and automation. Harbin: IEEE, 2010: 385-390.
- [2] 贾云得, 吕宏静, 徐岸, 等. 一种鱼眼镜头成像立体视觉系统的标定方法[J]. 计算机学报, 2000, 23(11): 1215-1219.
- [3] Kanatani K. Calibration of ultrawide fisheye lens cameras by eigenvalue minimization[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(4): 813-822.
- [4] Carroll R, Agrawal M, Agarwala A. Optimizing content-preserving projections for wide-angle images[J]. ACM Transactions on Graphics, 2009, 28(3): 1-9.
- [5] Wei Jin, Li Chenfeng, Hu Shimin, et al. Fisheye video correction[J]. IEEE Transactions on Visualization and Computer Graphics, 2012, 18(10): 1771-1783.
- [6] 英向华, 胡占义. 一种基于球面透视投影约束的鱼眼镜头校正方法[J]. 计算机学报, 2003, 26(12): 1702-1708.
- [7] 黄有度, 苏化明. 大视角鱼眼镜头图像中的漫游[J]. 工程

图学学报, 2006, 27(5): 44-49.

- [8] 周辉, 罗飞, 李慧娟, 等. 基于柱面模型的鱼眼影像校正方法的研究[J]. 计算机应用, 2008, 28(10): 2664-2666.
- [9] Zhu Haijiang, Wang Xuejing, Yi Chong. An elliptical function model for fisheye camera correction[C]//Proc of 2011 9th world congress on intelligent control and automation. Taipei: IEEE, 2011: 248-253.
- [10] Mundhenk T N, Rivett M J, Liao Xiaoqun, et al. Techniques for fisheye lens calibration using a minimal number of measurements[C]//Proc of intelligent systems and smart manufacturing. [s. l.]: International Society for Optics and Photonics, 2000: 181-190.
- [11] 杨玲, 成运. 应用经纬映射的鱼眼图像校正设计方法[J]. 工程图学学报, 2010, 31(6): 19-22.
- [12] 唐俊. 鱼眼图像轮廓提取算法研究[J]. 微机发展(现更名: 计算机技术与发展), 2004, 14(10): 9-10.
- [13] 江芳, 袁纵横. 鱼眼图像轮廓提取方法的对比[J]. 计算机系统应用, 2011, 20(4): 214-218.
- [14] 杨丹, 游磊, 张小洪, 等. 基于区域生长的鱼眼图像轮廓提取算法[J]. 计算机工程, 2010, 36(8): 217-218.

一种基于情景的协同过滤推荐算法

作者：[李荟](#)，[谢强](#)，[丁秋林](#)，[LI Hui](#)，[XIE Qiang](#)，[DING Qiu-lin](#)

作者单位：[南京航空航天大学 计算机科学与技术学院, 江苏 南京, 210016](#)

刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2014(10)

引用本文格式：[李荟](#).[谢强](#).[丁秋林](#).[LI Hui](#).[XIE Qiang](#).[DING Qiu-lin](#) 一种基于情景的协同过滤推荐算法[期刊论文]
]-[计算机技术与发展](#) 2014(10)