第 53 卷 第 4 期 厦门大学学报(自然科学版) Vol. 53 No. 4
2014 年 7 月 Journal of Xiamen University (Natural Science) Jul. 2014
doi:10.6043/j.issn.0438-0479.2014.04.006

一种基于蚁群算法的用户浏览路径推荐方法

刘晋佩,曾建平*
(厦门大学信息科学与技术学院,福建 厦门 361005)

摘要:在协同过滤算法的基础上,结合仿生学蚁群算法设计出一种新型的推荐算法.该算法模拟蚂蚁觅食原理,将用户视为"蚂蚁",目标商品视为"食物",利用蚂蚁之间通过信息素的交流来预测用户下一步将要浏览的商品项目.从标准数据集 MovieLens 上的测试结果表明,相比于传统的协同过滤算法,该算法可有效减少由数据集稀疏带来的问题,提高推荐系统的推荐质量.

关键词:协同过滤;蚁群算法;推荐系统
中图分类号:TP 391 **文献标志码**:A **文章编号**:0438-0479(2014)04-0465-04

随着计算机技术和网络的迅速发展,互联网的信息量呈爆炸式增长,信息过载成为互联网发展所面临的主要挑战.尤其在电子商务领域,信息过载问题更加突出,推荐系统正是解决该问题的有效手段之一.在电子商务推荐系统中[1],协同过滤算法是应用最为广泛的方法,它具有推荐效果好,实现和维护代价低等优点,成为最成功的个性化推荐技术之一.Xerox PARC 研究中心提供的 Tapestry[2]被认为是第 1 个协同过滤推荐系统,该系统用于过滤 E-mail 信息和 Usent 文章.由于协同过滤技术是基于用户对项目的评分数据,随着电子商务系统中用户和项目的快速增长,数据稀疏程度不断扩大,从而造成整个推荐系统的准确度不断降低.此外,推荐系统中新用户和新商品的加入还会造成系统无法完成推荐的冷启动问题.为了解决上述问题,需要探索新的推荐算法.

1 预备知识

Dorigo 等[3]在 1991 年提出了蚁群算法,此后在他的其他著作中详细地阐述了该算法的基本原理和数学模型.蚁群算法是种群进化算法中的一种,具有良好的协作和搜索能力.同时还是一种本质并行的算法,能够迅速发现最优解.因此,蚁群算法非常适合应用于推荐系统中解决由数据集稀疏带来的问题.

收稿日期:2013-08-27
*通信作者:jpzeng@xmu.edu.cn

根据蚁群算法[4],一个完整的推荐系统可以看作是由蚂蚁所代表的用户以及蚂蚁所要寻找的"食物"所代表的商品项目构成,因此,有如下假设:

1)将用户看作是推荐系统中的蚂蚁,蚂蚁之间通过信息素进行交流;

2)将推荐系统中的目标商品项目看作蚂蚁要寻找的"食物";

3)蚂蚁和食物之间存在很多非目标商品项目,将他们视为蚂蚁觅食路径上的节点,用户浏览这些商品项目就等同于蚂蚁访问这些节点,并留下信息素.这里信息素为用户对商品项目的评分.

设推荐系统中用户的总数为 M,商品项目总数为 N,记用户 i 对商品项目 j 的原始评分为 a'_{ij},实时评分为 a''_{ij},最终评分为 a_{ij}.

用户间相似度的计算通常是利用用户对商品的评分信息来计算得到的[5],其中 Pearson 相似性和余弦相似性是最常用的方法,由于 Pearson 相似性考虑了不同用户的评分喜好,因此可以提高寻找邻居用户的准确度,本文计算用户间的相似度所采用的便是 Pearson 相似性算法.定义用户 i_1 和 i_2 共同打过分的商品项目集合为 $S_{i_1 i_2}$,则用户 i_1 和 i_2 之间的相似度 $\mathrm{sim}(i_1,i_2)$ 定义为:

$$\mathrm{sim}(i_1,i_2)=\frac{\sum_{s\in S_{i_1 i_2}}(a'_{i_1 s}-\overline{a'_{i_1}})(a'_{i_2 s}-\overline{a'_{i_2}})}{\sqrt{\sum_{s\in S_{i_1 i_2}}(a'_{i_1 s}-\overline{a'_{i_1}})^2}\sqrt{\sum_{s\in S_{i_1 i_2}}(a'_{i_2 s}-\overline{a'_{i_2}})^2}},\quad(1)$$

其中,$\overline{a'_{i_1}}$ 和 $\overline{a'_{i_2}}$ 分别表示用户 i_1 和用户 i_2 对商品项目的平均评分.

目标用户 u 的最近邻居集合用 S_u 表示[6],在 S_u 中的用户是按与目标用户的相似度从高到低排列的,则目标用户 u 对未评分项目 j 的预测评分 P_{uj} 为:

$$P_{uj}=\overline{a'_u}+\frac{\sum_{s\in S_u}\mathrm{sim}(u,s)\times(a'_{si}-\overline{a'_s})}{\sum_{s\in S_u}(|\mathrm{sim}(u,s)|)}. \tag{2}$$

综上,用户 i 对项目 j 的最终评分[7]可表示为:

$$a_{ij}=\begin{cases}a''_{ij}, & \text{初次登陆的用户},\\ \dfrac{a'_{ij}+a''_{ij}}{2}, & \text{非初次登陆的用户},a'_{ij}\neq 0,\\ \dfrac{P_{uj}+a''_{ij}}{2}, & \text{非初次登陆的用户},a'_{ij}=0.\end{cases} \tag{3}$$

2 基于蚁群算法的推荐系统

2.1 转移概率函数的建立

用户在浏览商品时会根据浏览路径上信息素的强弱来决定下一个将要浏览的商品,这里用禁忌表集合 tabu 来记录用户浏览过的商品项目,集合 $D=\{1,2,\cdots,N\}$ 为推荐系统中项目的集合,用户从项目 j_1 转移到项目 j_2 的转移概率为:

$$P_{j_1j_2}=\begin{cases}\dfrac{[a_{ij_1}]^\alpha[\eta_{j_1j_2}]^\beta}{\sum_{k\in\mathrm{allowed}}[a_{ik}]^\alpha[\eta_{j_1k}]^\beta}, & j_2\in\mathrm{allowed},\\ 0, & \text{其他}.\end{cases} \tag{4}$$

其中,allowed=D−tabu 表示用户下一步可选择的商品集,$\eta_{j_1j_2}$ 为能见度,它表示用户从项目 j_1 转移到项目 j_2 的期望程度,可根据某种启发式函数来确定,具体形式将在下文中给出. 参数 α 和 β 分别表示信息素强度和能见度在转移概率中的相对重要性. α 越大,表示用户更倾向于选择最近邻居集中用户选择过的商品. β 越大,表示用户更倾向于选择与当前浏览项目相似的商品.

2.2 启发式函数的设计

启发式函数的设计来源于两部分,一部分来自于不同用户对同一项目的评分,为保持项目的原始属性,用项目 j 的原始评分向量 $C_j=(a'_{1j},a'_{2j},\cdots,a'_{Mj})$ 来表示项目的评价属性值,项目 j_1 和 j_2 的相似性由余弦相似性公式来表示:

$$\cos(C_{j_1},C_{j_2})=\frac{\sum_{i=1}^{M}a'_{ij_1}a'_{ij_2}}{\sqrt{\sum_{i=1}^{M}(a'_{ij_1})^2}\sqrt{\sum_{i=1}^{M}(a'_{ij_2})^2}}, \tag{5}$$

余弦值越大说明 I_{j_1} 和 I_{j_2} 越相似.

启发式函数的另一部分来自项目本身的属性值. 设 I 为 m 个具有 l 维属性的项目数据集合 $I=\{I_i\mid I_i=(I_{i_1},I_{i_2},\cdots,I_{il}),i=1,2,\cdots,m\}$,其中 I_{ij} 指项目 i 的第 j 个属性值,则项目 j_1 和 j_2 间的相似度由欧氏距离公式来表示:

$$d(I_{j_1},I_{j_2})=\sqrt{\sum_{k=1}^{l}(I_{j_1k}-I_{j_2k})^2}, \tag{6}$$

I_{j_1} 和 I_{j_2} 的欧氏距离越小说明 I_{j_1} 和 I_{j_2} 越相似.

综上,启发式函数 $\eta_{j_1j_2}$ 就可以表示为:

$$\eta_{j_1j_2}=\begin{cases}\dfrac{\cos(C_{j_1},C_{j_2})}{d(I_{j_1},I_{j_2})}, & d(I_{j_1},I_{j_2})\neq 0\\ \cos(C_{j_1},C_{j_2}), & d(I_{j_1},I_{j_2})=0\end{cases}, \tag{7}$$

$\eta_{j_1j_2}$ 值越大,表明用户从项目 j_1 转移到项目 j_2 的期望程度越大,反之越小.

2.3 项目评分更新规则

蚂蚁在它行走的过程中会在其运动轨迹上留下信息素,信息素的强弱会影响到其他蚂蚁的行为,如果某条运动轨迹上的信息素越强,那么其他蚂蚁选择该轨迹的概率也就越大,当有蚂蚁从这条轨迹上经过时还会继续增加该轨迹上信息素的强度,当然,如果随后该轨迹上没有蚂蚁经过,信息素的强度就会逐渐减弱. 这里信息素的更新实质上就是项目评分的更新,推荐系统每做一次推荐或遍历完所有的项目都会依据用户的反应,更新项目的评分,更新规则如下所示:

$$a_{ij}(\tau+1)=(1-\rho)\cdot a_{ij}(\tau)+\xi, \tag{8}$$

其中,$\rho(0\leqslant\rho<1)$ 表示信息素挥发因子,$1-\rho$ 为信息素残留因子,ξ 为本次遍历过程中的信息素增量.

信息素残留因子表示不同蚂蚁间相互影响的关系[8]:当信息素残留因子过大时,之前搜索过的路径被再次搜索的可能性会增大,降低了算法的随机性和全局搜索能力;虽然通过减小信息素残留因子可以提高算法的随机性和全局搜索能力,但又会降低算法的收敛速度. ξ 的初值为 0,如果用户浏览并购买推荐项目时,会使项目评分增加,即 $\xi>0$;如果用户未浏览或浏览后未购买推荐项目时,$\xi=0$.

3 实验结果与分析

3.1 数据集

实验采用由美国 Minnesota 大学的 GroupLens 研究小组创建并维护的 MovieLens 数据集,该数据集由 943 名用户对1 682部电影的100 000条评分记录所

组成,其中,任一用户都至少对 20 部电影打过评分,评分范围是 1～5,5 表示"非常喜欢",1 表示"不喜欢".评分的稀疏等级为 $\psi=1-100\ 000/(943\times1\ 682)=0.\ 93\ 695$,可见,该数据集的评分矩阵是非常稀疏的.实验随机地把数据按 8∶2的比例分为训练集和测试集,训练集用来产生实验结果,测试集用来测试实验结果的性能.

3.2　推荐质量的评价标准

这里采用分类准确度作为评价该算法优劣的标准[9],分类准确度表示的是推荐系统能否正确预测用户是否喜欢某个商品的能力.网站在提供推荐服务时,一般给用户一个个性化推荐列表,长度为 L.根据预测评分对训练集上的商品进行排序,记 $R(u)$为用户 u 在训练集上的最可能喜欢商品列表,$T(u)$为用户 u 在测试集上的最喜欢商品列表,T_u 为用户 u 在测试集上喜欢的商品数目,N_{pt} 为用户 u 在 $R(u)$和 $T(u)$上共有的商品数目.那么对用户 u,其推荐结果的准确率 $P_u(L)$(precision)和召回率 $R_u(L)$(recall)分别表示为:

$$P_u(L)=\frac{N_{pt}}{L},\tag{9}$$

$$R_u(L)=\frac{N_{pt}}{T_u},\tag{10}$$

这里,$P_u(L)$表示系统所推荐的 L 个商品中用户真正喜欢的商品所占的比例,$R_u(L)$表示一个用户真正喜欢的商品能被推荐出来的概率.

3.3　实验结果分析

实验使用 5 对数据集进行测试,取 5 次实验的平均值作为实验的结果.实验中当取 $\rho=0.\ 3$,$\xi=1.\ 5$,$\alpha=1.\ 8$,$\beta=1.\ 3$ 时,取得较好的结果.为了验证本文提出的算法的有效性[10],以 ItemRank 和 Maximum-frequency(MaxF)推荐算法作为对照,推荐长度从 10 增加到 60,间隔为 10,然后与本文提出的算法作比较,实验结果如表 1 所示.

将表 1 中的数据转换成图的形式,实验结果如图 1 所示.

由于 MaxF 算法只是将推荐系统内的项按被用户所浏览的次数进行简单的降序排序,将用户没有看过且排序靠前的项推荐给用户,没有考虑项目的评价属性和项目自身的属性,而 ItemRank 算法在将用户-项目评分二部图转换为相关图时会导致用户评分信息的丢失.本文提出的算法在运用蚁群算法强大的协作和搜索能力时,启发式函数的设计考虑了项目的评价属性和项目自身的属性,提高了寻找相似项目的准确度,同时保留了协同过滤算法来寻找目标用户的最近邻居集.从图 1 中可以看出,随着推荐长度的增加,3 种算法的准确率均降低,召回率增加,但本文提出的算法不论准确率还是召回率都比 MaxF 和 ItemRank 算法有了明显提高,在推荐长度为 40 时本文提出的算法的准确率和召回率分别为 27.14%和 31.82%,比 MaxF 算法分别高出了 10.82%和 10.62%,比 ItemRank 算法分别高出了 9.91%和 9.23%,证明了算法的有效性.

4　结　论

本文提出一种基于蚁群算法的电子商务推荐系统.该方法可有效地减少由数据集稀疏带来的问题,提高推荐系统的推荐质量,且通过设置相关参数解决新用户的冷启动问题.然而,蚁群算法中各参数的取值尚无严格的理论依据,至今还没有确定参数取值的一般方法,需要依靠大量的重复性实验寻找较优的结果.此外,由于推荐系统中的用户-评分矩阵是一个高维度的庞大矩阵[11],且数据稀疏度极高,目前只能通过扫描评分矩阵来进行查询,并且蚁群算法的引入也

表 1　不同推荐长度下各算法的准确率和召回率

Tab.1　Precision and recall under the different length of recommended list

长度	准确率			召回率		
	本文算法	ItemRank	MaxF	本文算法	ItemRank	MaxF
10	0.320 0	0.240 0	0.221 3	0.116 0	0.122 1	0.121 9
20	0.300 0	0.215 9	0.182 3	0.196 7	0.173 1	0.152 8
30	0.266 7	0.186 9	0.173 0	0.243 6	0.182 4	0.172 9
40	0.271 4	0.172 3	0.163 2	0.318 2	0.225 9	0.212 0
50	0.240 0	0.163 0	0.151 3	0.343 2	0.271 0	0.256 9
60	0.200 6	0.153 8	0.147 7	0.343 8	0.293 2	0.283 8

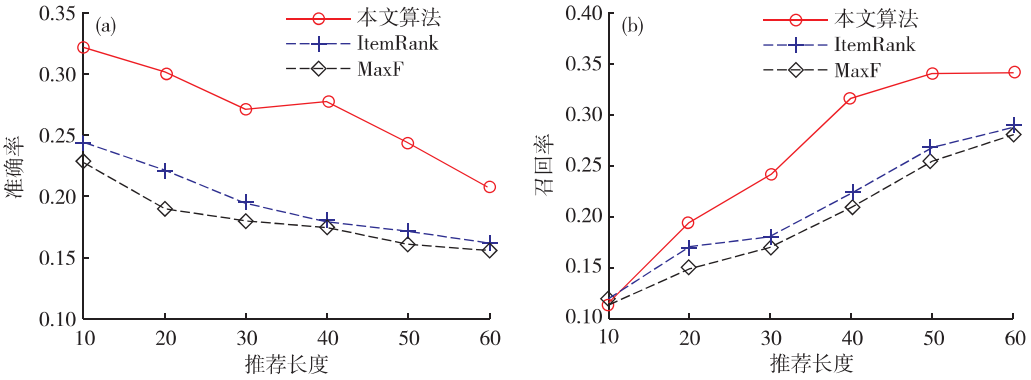

图 1 3 种算法在 MovieLense 数据集上的准确率(a)和召回率(b)

Fig. 1 Three algorithms performances of the MovieLens data set: (a) precision, (b) recall.

提高了扫面时间的复杂度. 因此,未来可以通过建立多维的动态索引结构,从而提高推荐系统的实时性.

参考文献:

[1] 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. New York, USA: ACM, 2001: 285-295.

[2] Deshpande M, Karypis G. Item-based top-N recommendation algorithms[J]. ACM Transactions on Information Systems (TOIS), 2004, 2(1): 143-177.

[3] Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents[J]. Systems Man and Cybernetics, Part B: Cybernetics, 1996, 26(1): 29-41.

[4] **王晗,夏自谦. 基于蚁群算法和浏览路径的推荐算法研究[J]. 中国科技信息**, 2009, 33(7): 103-104.

[5] **吴月萍,王娜,马良. 基于蚁群算法的协同过滤推荐系统的研究[J]. 计算机技术与发展**, 2010, 21(10): 73-76.

[6] **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.

[7] **周玉妮. 基于蚁群算法的移动商务个性化推荐体系研究[D]. 南京:南京邮电大学**, 2012.

[8] **詹士昌,徐婕,吴俊. 蚁群算法中有关参数的最优选择[J]. 科技通报**, 2003, 19(5): 381-386.

[9] **朱郁筱,吕琳媛. 推荐系统评价指标综述[J]. 电子科技大学学报**, 2012, 41(2): 163-175.

[10] Zhang Y, Wu J, Zhuang Y. Random walk models for top-N recommendation task[J]. Journal of Zhejiang University Science A, 2009, 10(7): 927-936.

[11] **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.

A Recommendation Approach to Users′ Browsing Path Based on Ant Algorithm

LIU Jin-pei, ZENG Jian-ping*

(School of Information Science and Engineering, Xiamen University, Xiamen 361005, China)

Abstract: A new recommendation approach is proposed combining ant algorithm and the collaborative filtering in this paper. According to the principle of ant foraging, the user is considered as "ant", the target commodity as "food", then using pheromones between the ants communication to predict the next commodity to be browsed. The benchmark MovieLens is applied in the experiment. Results shows that this method can efficiently alleviate the dataset sparsity problem, and provide better recommendation results than collaborative filtering algorithms.

Key words: collaborative filtering; ant algorithm; recommendation system