第 40 卷第 5 期
2010 年 9 月

东南大学学报(自然科学版)
JOURNAL OF SOUTHEAST UNIVERSITY (Natural Science Edition)

Vol. 40 No. 5
Sept. 2010

doi:10.3969/j.issn.1001-0505.2010.05.007

一种综合用户和项目因素的协同过滤推荐算法

黄裕洋　　金远平

(东南大学计算机科学与工程学院, 南京 210096)

摘要: 针对用户评分数据极端稀疏情况下传统协同过滤推荐算法的不足, 提出了一种综合用户和项目因素的最近邻协同过滤推荐(HCFR)算法. 该算法首先以一种改进的相似性度量方法(ISM)为基础, 根据当前评分数据的稀疏情况, 动态调节相似度的计算值, 真实地反映彼此之间的相似性. 然后, 在产生推荐时综合考虑用户和项目的影响因素, 分别计算目标用户和目标项目的最近邻集合. 最后, 根据评分数据的稀疏情况, 自适应地调节目标用户和目标项目的最近邻对最终推荐结果的影响权重, 并给出推荐结果. 实验结果表明, 与传统的只基于用户或基于项目的推荐算法相比, HCFR算法在用户评分数据极端稀疏情况下仍能显著地提高推荐系统的推荐质量.

关键词: 协同过滤推荐; 数据稀疏; 相似性; 评分预测

中图分类号: TP312　　**文献标志码**: A　　**文章编号**: 1001-0505(2010)05-0917-05

Collaborative filtering recommendation algorithm based on both user and item

Huang Yuyang　　Jin Yuanping

(School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract: To solve the shortcomings of the traditional collaborative filtering recommendation algorithms in the situation of extreme sparsity of user's rating data, a hybrid collaborative filtering recommendation (HCFR) algorithm for the nearest neighbors based on users and items is proposed. First, on the basis of correlation similarity, this algorithm adopts an improved similarity measure method (ISM) which can dynamically adjust the value of similarity according to the current state of sparse rating data and truly reflect the real situation. Then, in the process of generating recommendation results, both user factors and item factors are considered and the nearest neighbor sets of the active user and the active item are obtained. Finally, according to the sparsity of the user's rating data, different self-adaptive influence weights of the neighbor sets of the active user and the active item are adjusted, and the final recommendation results are obtained. The experimental results show that compared with the traditional recommendation algorithms which are only based on user or item, the HCFR algorithm can effectively improve the recommendation quality even in the situation of extreme sparsity of user's rating data.

Key words: collaborative filtering recommendation; data sparsity; similarity; rating prediction

最近邻协同过滤推荐是个性化推荐中研究和应用最多的一种技术, 已经被广泛地应用于电子商务的各个领域. 常用的最近邻协同过滤推荐算法可分为基于用户和基于项目 2 种. 为了找到目标用户真正感兴趣的内容, 基于用户的推荐算法需要首先找到与此用户有相似兴趣

收稿日期: 2010-01-21　作者简介: 黄裕洋(1986—), 男, 硕士生; 金远平(联系人), 男, 教授, ypjin@seu.edu.cn
基金项目: 国家自然科学基金资助项目(60973023).
引文格式: 黄裕洋, 金远平. 一种综合用户和项目因素的协同过滤推荐算法[J]. 东南大学学报: 自然科学版, 2010, 40(5): 917-921. [doi:10.3969/j.issn.1001-0505.2010.05.007]

的其他用户,然后将他们感兴趣的内容推荐给此用户[1].随着用户数量的增多,这种算法的计算量呈线性增长,性能变差,且不能对推荐结果提供很好的解释.2001年,Sarwar等[2]提出了一种基于项目的协同过滤推荐算法.该算法假设当大部分用户对一些项目的评分比较相似时,当前用户对这些项目的评分也是相似的,因此可以先计算出目标项目的最近邻集合,然后根据目标项目的邻居项目评分值来预测该项目的评分.

数据稀疏性是最近邻协同过滤推荐中的一个重要问题.在电子商务网站等实际应用中,用户评分项目一般不会超过总数的1%[3].评分数据极度稀疏,直接导致用户之间或者项目之间的相似度计算不准确,从而影响了推荐精度.针对这一不足,研究者们开展了很多相关的改进工作.Sarwar等[4]提出通过奇异值分解(SVD)减少项目空间的维数,使用户在减少的项目空间上对每一个项目均有评分;这种方法的缺点在于,当维数很高时难以保证其降维效果.各种聚类(clustering)技术[5-8]被研究者们应用于离线数据的预处理,与在线实时推荐技术相结合可以降低数据的稀疏性;然而,聚类技术的使用限制了评分数据矩阵的多样性.邓爱林等[9]提出了一种基于项目评分预测的协同过滤推荐算法;李聪等[10]对此算法进行了进一步改进,只对有推荐能力的用户采用领域最近邻方法进行评分预测,从而在一定程度上降低了数据稀疏性带来的不良影响.Ma等[11]提出了一种同时基于目标用户和目标项目的最近邻集合来产生推荐的方法,用一个调节因子λ来控制两者对最终推荐结果的影响权重,但λ需要根据经验来手动设定,往往不能得到最佳的推荐效果.

在上述研究的基础上,本文提出了一种新的最近邻协同过滤推荐(HCFR)算法.该算法能够更加真实地反映用户或者项目之间的相似性,在进行组合推荐时综合考虑了用户和项目的影响因素.实验结果表明,该算法能有效提高推荐质量,产生较好的推荐效果.

1 HCFR算法

HCFR算法可分为3个阶段:① 构建用户-项目评分矩阵;② 计算目标用户和目标项目的最近邻集合;③ 产生推荐结果.算法流程图如图1所示.

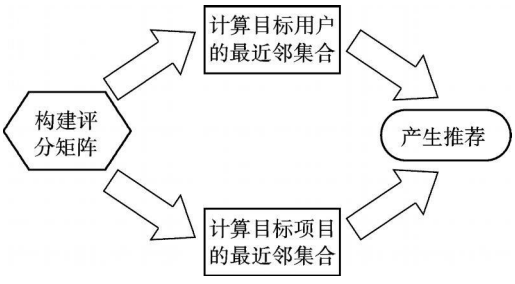

图1 HCFR算法流程图

作为算法的输入数据,用户-项目评分矩阵通常可表述为一个 $m\times n$ 的矩阵 R,其中 m 为用户数,n 为项目数,矩阵元素 r_{ij} 表示第 i 个用户对第 j 个项目的评分值.评分值越高,用户对该项目的认可度就越高.

1.1 相似性度量方法的改进

1.1.1 传统的相似性度量方法

传统的相似性度量方法主要有3种:余弦(cosine)相似性度量方法(CSM)、修正的余弦(adjusted cosine)相似性度量方法(ACSM)和相关(correlation)相似性度量方法[12].这些方法都是基于对象向量、在对象属性之间进行严格匹配的.

在基于用户的推荐算法中,余弦相似性度量方法采用向量的余弦夹角度量彼此之间的相似性,忽略了用户评分的统计特征.修正的余弦相似性度量方法在余弦相似性基础上,考虑了不同用户的评分尺度问题,减去了用户对项目的平均评分;该方法更多体现的是用户之间的相关性而非相似性[13].相关相似性度量方法根据双方共同评分的项目进行相似性度量.如果用户间的所有评分项目均为共同评分项目,那么相似相关性和修正的余弦相似性是等同的.用户对共同评分项目的评分能够体现出用户间的相似程度,计算公式如下:

$$S(a,u)=\frac{\sum_{k\in I_a\cap I_u}(r_{a,k}-\bar{r}_a)(r_{u,k}-\bar{r}_u)}{\sqrt{\sum_{k\in I_a\cap I_u}(r_{a,k}-\bar{r}_a)^2}\sqrt{\sum_{k\in I_a\cap I_u}(r_{u,k}-\bar{r}_u)^2}} \tag{1}$$

式中,$S(a,u)$ 表示用户 a 与用户 u 的相似度;I_a 和 I_u 分别表示用户 a 和用户 u 中含有评分值的项目的集合;$I_a\cap I_u$ 表示用户 a 和用户 u 共同评分的项目的集合;$\bar{r}_a$ 和 $\bar{r}_u$ 分别表示用户 a 和用户 u 对项目的平均评分值;$r_{a,k}$ 和 $r_{u,k}$ 分别表示用户 a 和用户 u 对项目 k 的评分值.

与用户相似度的定义类似,在基于项目的推荐算法中,可通过比较对同一个项目存在评分值的用户集合来计算相似性,即

$$S(i,j)=\frac{\sum_{u\in U_i\cap U_j}(r_{u,i}-\bar{r}_i)(r_{u,j}-\bar{r}_j)}{\sqrt{\sum_{u\in U_i\cup U_j}(r_{u,i}-\bar{r}_i)^2}\sqrt{\sum_{u\in U_i\cup U_j}(r_{u,j}-\bar{r}_j)^2}} \tag{2}$$

式中, $S(i,j)$表示项目 i和项目 j的相似度; U_i和 U_j分别表示项目 i和项目 j中存在评分值的用户集合; $\bar{r}_i$表示项目 i的平均被评分值; $r_{u,i}$和 $r_{u,j}$表示用户 u对项目 i和 j的评分值.

由此可知, $S(a,u)\in[0,1]$, $S(i,j)\in[0,1]$. 相似性公式的计算值越大表明用户之间或者项目之间越相似.

1.1.2　改进的相似性度量方法

在用户评分数据较多的情况下, 传统的相关相似性度量方法一般都能取得不错的度量效果. 但在实际应用中, 用户评分数据往往极度稀疏, 传统的相关相似性度量方法不能准确地反映数据稀疏对相似性度量的影响. 例如, 当 $S(a,u)$一定时, 如果用户 a和用户 u之间共同评分的项目很少, 则在不考虑其他条件 (如项目子类别划分等)的情况下, 可以认为两者之间的相似性也相对较小. 因此, 仅利用式 (1)得到的计算值来设定相似度是不合理的.

McLaughlin等[14]提出, 针对数据稀疏情况, 可以通过增加一个权重因子 θ来改进相似度的计算结果, 即

$$S_\theta(a,u)=\frac{\max(|I_a\cap I_u|,\theta)}{\theta}S(a,u) \tag{3}$$

式 (3)可以克服评分项目过少所带来的不利影响, 但是 θ需要根据经验手动设定. 如果 $|I_a\cap I_u|>\theta$, 则 $S_\theta(a,u)>1$; 在 θ设置不佳的情况下, $S_\theta(a,u)$甚至可以达到 2或 3. 因此, 权重因子的设置会较大程度地影响相似性的度量.

本文提出的改进的相似性度量方法 (ISM)能根据数据的稀疏状况自适应地调节相似度的大小, 更准确地反映数据稀疏情况对相似性计算的影响. 将用户之间共同评分的项目比例作为度量相似性的一个辅助因素, 对式 (1)进行了如下改进:

$$S_1(a,u)=(1-\chi)+\chi\frac{|I_a\cap I_u|}{|I_a\cup I_u|}S(a,u) \tag{4}$$

式中, χ表示权重指数, 且 $\chi\in(0,1]$.

当 $S(a,u)$一定时, 用户 a和用户 u之间共同评分的项目比例越小, 则其相似性也越小; 反之, 如果其共同评分的项目比例越高, 则其相似性也越大. 通过改变权重指数 χ可以调整相似度的值空间, χ越大则值空间的增长速度越快. χ的设置会影响到算法的推荐效果, 因此应该根据不同的推荐系统动态调整 χ从而实现推荐效果的优化.

类似地, 基于项目的 ISM计算公式为

$$S_1(i,j)=(1-\mu)+\mu\frac{|U_i\cap U_j|}{|U_i\cup U_j|}S(i,j) \tag{5}$$

式中, μ表示可手动设定的权重指数, 且 $\mu\in(0,1]$.

1.2　最近邻集合的计算

对于每个未知评分数据 $r_{u,i}$, 最近邻居用户的集合可表示为

$$M(u)=\{u_n|S_1(u_n,u)>\alpha,\ u_n\neq u\} \tag{6}$$

最近邻居项目的集合可表示为

$$M(i)=\{i_n|S_1(i_n,i)>\beta,\ i_n\neq i\} \tag{7}$$

式中, u_n表示用户 u的最近邻居用户; i_n表示项目 i的最近邻居项目; α和 β分别表示确定邻居用户和邻居项目数目的阈值.

1.3　推荐结果的产生

推荐结果的产生是通过预测目标用户对未评分项目的评分实现的, 可分为单一推荐和多项推荐. 当使用单一协同过滤推荐方法时, 若某个未评分项目的预测评分大于等于设定的阈值, 则推荐系统可以向目标用户推荐该项目; 若预测评分小于阈值, 则不向目标用户推荐该项目. 当使用多项协同过滤推荐方法时, 可通过选择其中排在前面 N位的预测评分所对应的项目向目标用户进行推荐.

在计算某一未知数据 $r_{u,i}$的预测评分时, 单方面考虑用户或者项目的因素往往不能得到最准确的预测结果. 本文综合考虑了用户和项目的影响因素, 通过加权计算得到预测评分, 计算公式为

$$P(r_{u,i})=\sigma\left[\bar{r}_u+\frac{\sum_{u_n\in S(u)}S_1(u_n,u)(r_{u_n,i}-\bar{r}_{u_n})}{\sum_{u_n\in M(u)}S_1(u_n,u)}\right]+(1-\sigma)\left[\bar{r}_i+\frac{\sum_{i_n\in S(i)}S_1(i_n,i)(r_{u,i_n}-\bar{r}_{i_n})}{\sum_{i_n\in M(i)}S_1(i_n,i)}\right] \tag{8}$$

式中, $P(r_{u,i})$表示用户 u对项目 i的预测评分值; σ表示用户和项目的影响因素的比例调节因子; $\bar{r}_u$表示用户 u对项目 i的平均评分值; $\bar{r}_i$表示项目 i的平均被评分值.

假设 I为所有项目的集合, U为所有用户的集合, 令 $\sigma_1=|I_u|/|I|$, $\sigma_2=|U_i|/|U|$, 则

$$\sigma=\begin{cases}1 & M(u)\neq\varnothing,\ M(i)=\varnothing\\ \dfrac{\sigma_1}{\sigma_1+\sigma_2} & M(u)\neq\varnothing,\ M(i)\neq\varnothing\\ 0 & M(u)=\varnothing,\ M(i)\neq\varnothing\end{cases} \tag{9}$$

由式(9)可知,随着目标用户和目标项目的评分数目占总的用户和项目集合数目比例的变化,σ的值会随之变化,因此,推荐结果中邻居用户和邻居项目的影响权重也会变化.当 $M(u)\neq\varnothing$且 $M(i)=\varnothing$时,$\sigma=1$,表明此时没有可供推荐的邻居项目信息,只需要考虑邻居用户集合的影响因素.当 $M(u)=\varnothing$且 $M(i)\neq\varnothing$时,$\sigma=0$,表明此时没有可供推荐的邻居用户信息,只需要考虑邻居项目集合的影响因素.当 $M(u)=\varnothing$且 $M(i)=\varnothing$时,没有可供推荐的信息,此时用户 u对项目 i的预测评分值可表示为

$$P(r_{u,i})=\sigma r_u+(1-\sigma)r_i \tag{10}$$

式中,σ的值可以手动确定,也可以考虑通过式(9)进行自适应确定.

2 实验结果及分析

2.1 数据集

实验采用的数据集是目前衡量推荐算法质量时比较常用的 MovieLens数据集,由美国明尼苏达大学 GroupLens研究小组创建并维护.从用户评分数据库中选择 2.01×10^4 条评分数据,其中包含了930个用户对 440部电影的评分,且每个用户至少对 20部电影进行了评分,评分值范围为 1~5.实验中采用整个实验数据集的 80%作为训练集,剩余的 20%作为测试集,则该数据集的稀疏等级为

$$\Psi=\frac{1-2.01\times10^4}{930\times440}=0.95087$$

由此可见,这个数据集的评分矩阵是相当稀疏的.

2.2 度量标准

在统计精度度量方法中,可以采用平均绝对误差(MAE)来直观地度量推荐质量.因此,本文采用 MAE作为度量标准,通过计算用户对项目的预测评分与实际评分之间的偏差来度量预测的准确性.MAE的值越小,则推荐质量越高.假设算法对 W个项目预测的评分集合为 $\{p_1,p_2,\cdots,p_W\}$,对应的实际用户评分集合为 $\{q_1,q_2,\cdots,q_W\}$,则算法的 MAE可表示为

$$M=\frac{\sum_{i=1}^{W}|p_i-q_i|}{W} \tag{11}$$

2.3 相似性度量方法的比较

采用传统的基于用户的最近邻协同过滤推荐算法,从 10个最近邻开始逐步递增,对本文提出的改进的相似性度量方法、传统的余弦相似性度量方法以及修正的余弦相似性度量方法进行了比较.此时,权重指数 $\chi=\mu=0.7$.实验结果如图 2所示.

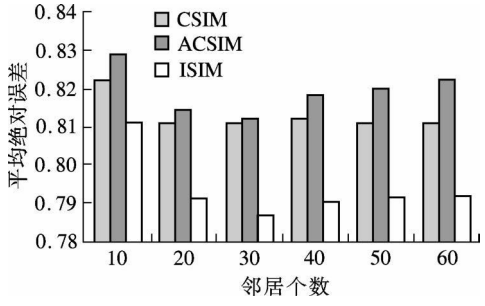

图 2 相似性度量方法的比较

由图 2可以看出,与传统的余弦相似性度量方法和修正的余弦相似性度量方法相比较,本文提出的改进的相似性度量方法在不同的邻居个数下都能取得较小的平均绝对误差,因此该方法能更好地度量相似性.

2.4 最近邻协同过滤推荐算法的比较

为了验证本文提出的最近邻协同过滤推荐算法(HCFR)的有效性,将该算法与传统的基于用户的最近邻协同过滤推荐算法(UCFR)以及文献[8]提出的基于项目的最近邻协同过滤推荐算法(ICFR)进行了比较,权重指数 $\chi=\mu=0.7$.比较结果如图 3所示.

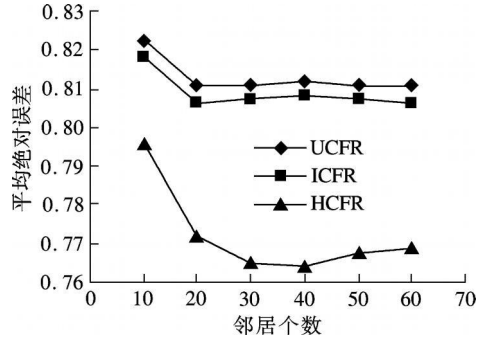

图 3 最近邻协同过滤推荐算法的比较

由图 3可以看出,当邻居数目递增时,HCFR算法的平均绝对误差明显低于传统的 2种算法,因此该算法可以有效地提高推荐质量.

3 结语

针对最近邻协同过滤推荐算法中的数据稀疏性问题,提出了一种综合用户和项目因素的优化推荐算法.它对相关相似性度量方法进行了改进,有效地解决了评分数据极端稀疏情况下传统相似性度量方法存在的不足;在产生推荐时,综合考虑用户和项目的影响因素,分别计算目标用户和目标项

目的最近邻集合，根据评分数据的稀疏情况动态调节目标用户和目标项目的最近邻对最终推荐结果的影响权重.实验结果表明，该算法能较好地提高推荐质量，改善推荐效果.然而，本文算法中的权重指数 λ和 μ是手动设定的，且随着评分矩阵数据量的膨胀和维数的增加，对算法的复杂度和可扩展性的要求也进一步提高.因此，下一步的研究重点是在保证推荐质量的同时，寻找一种更好的数据资源增量维护策略以及提高实时推荐效率的方法.

参考文献 (References)

[1] Deshpande M, Karypis G. Item-based top-N recommendation algorithms[J]. ACM Trans Information System, 2004, **22**(1): 143-177.

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

[3] Sun Xiaohua, Kong Fansheng, Ye Song. A comparison of several algorithms for collaborative filtering in startup stage[C]//Proceedings of the 2005 IEEE International Conference on Networking, Sensing and Controlling. Los Alamitos, CA, USA, 2005: 25-28.

[4] Sarwar B M, Karypis G. Application of dimensionality reduction in recommender system: a case study[C]//Proceedings of ACM Web KDD Workshop on Web Mining for E-commerce. New York, USA, 2000: 114-121.

[5] Gong Songjie, Ye Hongwu. Joining user clustering and item based collaborative filtering in personalized recommendation services[C]//Proceedings of the 2009 International Conference on Industrial and Information Systems. Haikou, China, 2009: 149-151.

[6] Braak Paul, Abdullah Noraswaliza, Xu Yue. Improving the performance of collaborative filtering recommender systems through user profile clustering[C]//Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology. Milan, Italy, 2009: 147-150.

[7] Xue G R, Lin C, Yang Q, et al. Scalable collaborative filtering using cluster-based smoothing[C]//Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Salvador, Brazil, 2005: 114-121.

[8] Wang J, de Vries A P, Reinders M J. 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. Washington DC, USA, 2006: 501-508.

[9] 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 Ailin, Zhu Yangyong, Shi Bole. A collaborative filtering recommendation algorithm based on item rating prediction[J]. Journal of Software, 2003, **14**(9): 1621-628. (in Chinese)

[10] 李聪，梁昌勇，马丽. 基于领域最近邻的协同过滤推荐算法[J]. 计算机研究与发展，2008, 45(9): 1532-1538.
Li Cong, Liang Changyong, Ma Li. A collaborative filtering recommendation algorithm based on domain nearest neighbor[J]. Journal of Computer Research and Development, 2008, **45**(9): 1532-1538. (in Chinese)

[11] Ma Hao, King Irwin, Lyu Michael. Effective missing data prediction for collaborative filtering[C]//Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Amsterdam, Netherlands, 2007: 39-46.

[12] 周军锋，汤显，郭景峰. 一种优化的协同过滤推荐算法[J]. 计算机研究与发展，2004, 41(10): 1842-1847.
Zhou Junfeng, Tang Xian, Guo Jingfeng. An optimized collaborative filtering recommendation algorithm[J]. Journal of Computer Research and Development, 2004, **41**(10): 1842-1847. (in Chinese)

[13] Tao Yufei, Yi Ke, Sheng Cheng, et al. Quality and efficiency in high dimensional nearest neighbor search[C]//Proceedings of the 35th SIGMOD International Conference on Management of Data. Rhode Island, USA, 2009: 563-576.

[14] 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. Sheffield, UK, 2004: 329-336.