·人工智能及识别技术· 文章编号:1000-3428(2016)01-0199-04 文献标识码:A 中图分类号:TP18

基于相似度拓展与兴趣度缩放的协同过滤算法

夏平平,帅建梅

(中国科学技术大学自动化系,合肥 230027)

摘 要:现有的协同过滤算法未考虑用户浏览记录中用户对项目的潜在厌恶信息,忽视新老用户对不同流行度项目的兴趣差异。为此,提出一种改进的协同过滤算法。从用户浏览记录中提取用户对项目的潜在厌恶信息,计算项目之间被用户厌恶的相似度,将其与项目之间被用户喜欢的相似度结合,得到项目的综合相似度。在此基础上用偏好因子对用户的兴趣度进行缩放,该因子能够反映新老用户对不同流行度项目的倾向性。实验结果表明,该算法在不明显增加时空复杂度的前提下,可有效提高推荐准确率、召回率和覆盖率。

关键词:协同过滤;潜在厌恶信息;偏好因子;相似度拓展;兴趣度缩放

中文引用格式:夏平平,帅建梅. 基于相似度拓展与兴趣度缩放的协同过滤算法[J]. 计算机工程,2016,42(1):199-202,209.

英文引用格式:Xia Pingping,Shuai Jianmei. Collaborative Filtering Algorithm Based on Similarity Extension and Interest Degree Scaling[J]. Computer Engineering,2016,42(1):199-202,209.

Collaborative Filtering Algorithm Based on Similarity Extension and Interest Degree Scaling

XIA Pingping,SHUAI Jianmei

(Department of Automation,University of Science and Technology of China,Hefei 230027,China)

【Abstract】 The existing Collaborative Filtering(CF) recommendation algorithms are not taken into account the latent information of users' aversion to items in the users' browsing history and the difference between old users' interests and new users' interests in items of different popularity. To solve the problem,this paper proposes a collaborative filtering algorithm based on similarity extension and interest degree scaling. It extracts the latent information of users' aversion to items from the users' browsing history to calculate the items' similarity of user' aversion,and combines it with items' similarity of users' favorite to get the integrated similarity of items. On the basis of that,the algorithm scales the users' interest degree to the items by preference factor which can reflect the difference between new users' interests and old users' interests in both popular items and unpopular items. Experimental results show that this algorithm improves the recommendation precision,recall rate and coverage,without obviously increasing in time-space complexity.

【Key words】 Collaborative Filtering(CF);latent aversion information;preference factor;similarity extension;interest degree scaling

DOI:10.3969/j.issn.1000-3428.2016.01.035

1 概述

协同过滤推荐算法是当下最流行的个性化推荐算法,它主要分为基于模型的协同过滤、基于图的协同过滤和基于邻域的协同过滤 3 类方法[1]。其中基于邻域的协同过滤因其时空复杂度相对较低和可解释性强,应用最为广泛。

现有的协同过滤算法只考虑用户对项目的喜好信息,而忽视了用户浏览记录中用户对项目的潜在厌恶信息以及新老用户对不同流行度项目的兴趣差异。实验分析表明:新用户倾向于浏览热门的项目;老用户会逐渐开始浏览冷门的项目,用户越活跃,越倾向于浏览冷门的项目[2]。

针对上述问题,本文在基于邻域的协同过滤算法基础上,考虑用户潜在厌恶信息和新老用户兴趣差异,提出一种基于相似度拓展和兴趣度缩放的协同过滤算法。

2 已有基于邻域的协同过滤算法

在实际应用中,基于邻域的协同过滤算法一般

作者简介:夏平平(1991-),男,硕士研究生,主研方向为数据挖掘、推荐算法;帅建梅,高级工程师。

收稿日期:2015-01-28 修回日期:2015-02-28 E-mail:xpp8@mail.ustc.edu.cn

采用 TOP-N 的推荐方式,输入用户的浏览记录,计算并预测用户对项目的兴趣度,最后输出用户兴趣度最大的 N 个项目作为推荐结果。

2.1 经典方法

基于邻域的方法根据求邻居集的不同,分为基于用户的协同过滤(User-based Collaborative Filtering,UserCF)和基于项目的协同过滤(Item-based Collaborative Filtering,ItemCF)[3]。UserCF 通过分析用户的浏览记录,计算用户间的相似度 $s_{ij}=\frac{|N(u)\cap N(v)|}{\sqrt{|N(u)|\times|N(v)|}}$,得到与用户有相似浏览记录的用户邻居集,然后推荐邻居们看过而自己没看过的项目给用户,其中,$|N(u)|$ 为用户 u 浏览过的项目数目。ItemCF 通过分析项目的被浏览记录,计算项目间的相似度 $s_{ij}=\frac{|N(i)\cap N(j)|}{\sqrt{|N(i)\times|N(j)|}}$,得到与其有相似浏览记录的项目邻居集,然后推荐与用户看过的项目相似的项目给用户,其中,$|N(i)|$ 为浏览过项目 i 的用户数目。

2.2 现有的改进方法

人们对基于邻域的协同过滤方法进行了一些改进,主要分为算法改进和混合改进 2 种方式。

算法改进是针对基于邻域的协同过滤算法本身进行改进。如文献[4]提出的 UserCF-IIF 通过将用户兴趣相似度公式中的分子除以 $\ln(1+|N(i)|)$,惩罚 2 个用户共同兴趣列表中的热门项目对他们相似度的影响;提出的 ItemCF-IUF 通过将项目相似度公式中的分子除以 $\ln(1+|N(u)|)$,惩罚浏览过2 个项目的用户列表中活跃用户对它们相似度的影响;文献[5]提出的 ItemCF-Norm 通过将项目相似度矩阵按最大值归一化,提高推荐结果的准确度和覆盖率。

混合改进是将基于邻域的协同过滤算法与其他推荐算法进行混合使用,常用的方式有与内容过滤混合使用、使用不同推荐算法进行预填充和 2 种基于邻域的协同过滤算法混合使用等。与内容过滤混合使用可以充分利用内容信息来提高推荐效果:如文献[6]提出在基于用户的协同过滤基础上进行内容过滤,文献[7]提出使用标签内容代替浏览记录来计算用户相似度,文献[8]提出在项目分类信息基础上使用带时间加权信息的基于项目的协同过滤算法,但是实际中内容信息需要大量的人工标注,同时也增加了算法时空复杂度,应用范围有限。使用不同推荐算法进行预填充可以减少用户项目矩阵的稀疏度,从而提高推荐效果:如文献[9]提出在项目的协同过滤预填充的基础上进行基于用户的协同过滤,文献[10]提出在基于项目分类预填充的基础上进行基于项目的协同过滤,但是这样做也明显增加了算法的时空复杂度。2 种基于邻域的协同过滤算法混合使用,充分利用了用户相似信息和项目相似信息来提高推荐效果:如文献[11]提出使用将项目相似度和用户相似度叠加的综合相似度来计算邻居集,文献[12]提出分别使用基于项目和基于用户的协同过滤求得预测评分再进行加权,但是这么做不仅增加了算法时空复杂度,而且也影响了基于邻域的协同过滤算法的推荐列表更新实时性。总体来说,混合改进虽然提升推荐效果,却大大增加了系统的时空复杂度,而且往往实时性和可解释性都会减弱。

3 协同过滤算法

基于相似度拓展和兴趣度缩放的协同过滤算法(Collaborative Filtering algorithm based on Similarity Extension and Interest Degree Scaling,ItemCF-SEIDS),是以基于邻域的协同过滤算法为框架,输入用户的浏览记录,考虑用户的潜在厌恶信息,通过拓展的综合相似度计算出用户对项目的兴趣度;再考虑新老用户对不同流行度项目的兴趣差异,用偏好因子将用户兴趣度进行缩放,最后输出用户没浏览过的项目中用户兴趣度最大的 N 个项目作为推荐结果。

3.1 算法框架选择

在协同过滤算法中,基于邻域的方法时空复杂度相对较低,都可以很容易地实时更新推荐列表。但 ItemCF 推荐用户用过的项目的类似项目给用户,侧重于维系用户的历史兴趣,比 UserCF 个性化更强,推荐理由更有说服力。同时 ItemCF 在用户数目庞大时,它的算法时空复杂度要远低于 UserCF。

综合考虑,本文算法以 ItemCF 作为基本框架。用户数目为 M,项目数目为 N,邻居数目为 k,记录总数为 T,则 ItemCF-SEIDS 离线计算空间复杂度为 $O(N^2)$,离线计算时间复杂度为 $O(N^2(T/N))$,使用倒排索引[13]的时间复杂度降为 $O(M(T/M)^2)$,推荐列表在线更新时间复杂度为 $O(k(T/M))$。它的推荐理由和 ItemCF 一样。

3.2 考虑用户潜在厌恶信息进行的相似度拓展

对于每个用户来说,看过项目则默认用户喜欢它,用户没看过该项目则可能有 2 种情况:不喜欢或者没接触到。用户没看的项目流行度越高,则用户接触过它的概率越大,用户不喜欢它的可能性也越大;用户的浏览记录越多,则用户接触到其厌恶的项目的可能性越大。

所以,首先将用户没看过的项目按流行度由高到低排序,按顺序挑选出与用户浏览记录数目相当的项目添加到用户的厌恶记录;然后以此计算项目被用户厌恶的相似度,该相似度与项目流行度和用户浏览记录数目相关;最后与项目被用户喜欢的相

似度结合得到项目的综合相似度,再将其标准化。被厌恶相似度从与被喜好相似度相反的方向拓展了相似度计算依据。详细步骤见算法 1。

算法 1　基于潜在厌恶信息的相似度拓展算法

输入　用户-项目矩阵 $\boldsymbol{R}$

输出　项目综合相似度矩阵 $\boldsymbol{W'}$

(1)获得用户-项目矩阵 $\boldsymbol{R}$。$\boldsymbol{R}$ 如表 1 所示,r_{ui} 为用户 u 对项目 i 的浏览记录,浏览过则 $r_{ui}=1$,否则 $r_{ui}=0$。

表 1　用户-项目矩阵

User	Item1	Item2	…	Item$N$
User1	1	1	…	0
User2	0	1	…	1
…	…	…	r_{ui}	…
User$M$	1	1	…	1

(2)计算项目的被喜好相似度 s_{ij}。被喜好相似度采用文献[4]的计算方法,项目 i 与项目 j 被喜好相似度 s_{ij} 计算公式如下:

$$s_{ij}=\frac{\sum_{u\in N(i)\cap N(j)}1/\ln(1+|N(u)|)}{\sqrt{|N(i)|\times|N(j)|}} \tag{1}$$

其中,$N(u)$ 为用户 u 的浏览记录;$N(i)$ 为浏览过项目 i 的用户集合。

(3)计算厌恶信息可信度系数 τ_{ui}。首先将所有项目按流行度由高到低排序,得到厌恶候选表 PT。然后从 PT 中按顺序取出与用户记录数量相仿的用户没看过的项目,构建用户厌恶记录表 HT。用户浏览记录越多,项目的流行度越高,相应的厌恶信息越可信。厌恶信息可信度系数 τ_{ui} 计算公式如下:

$$\tau_u=\sqrt{\frac{\log(1+|N(u)|)}{\log(1+\max_u(|N(u)|))}} \tag{2}$$

$$\tau_i=\sqrt{\frac{\log(1+|N(i)|)}{\log(1+\max_i(|N(i)|))}} \tag{3}$$

$$\tau_{ui}=\tau_i\times\tau_i \tag{4}$$

(4)计算项目的被厌恶相似度 v_{ij}:

$$v_{ij}=\frac{\sum_{u\in N'(i)\cap N'(j)}\sqrt{\tau_{ui}\times\tau_{uj}}}{\sqrt{|N'(i)|\times|N'(j)|}} \tag{5}$$

其中,$N'(i)$ 为厌恶项目 i 的用户集合。

(5)计算项目 i 与项目 j 的综合相似度 w_{ij}:

$$w_{ij}=s_{ij}+v_{ij} \tag{6}$$

(6)综合相似度 w_{ij} 归一化。因为热门类里的项目相似度往往比其他类内的相似度高,所以对项目综合相似度进行标准化。这一步的作用是消减热门类项目的推荐能力。归一化后的综合相似度 w'_{ij} 计算公式如下:

$$w'_{ij}=\frac{w_{ij}}{max_j(w_{ij})} \tag{7}$$

其中,$w'_{ij}\neq w'_{ji}$。

3.3　考虑用户对项目偏好因子的兴趣度缩放

新用户倾向于浏览热门的项目,老用户会逐渐开始浏览冷门的项目,用户越活跃,越倾向于浏览冷门的项目。根据这条规律,提出一个偏好因子概念,该因子能够反映新用户和老用户对热门项目和冷门项目的不同倾向性:用户的浏览记录数目 $|N(u)|$ 占用户平均浏览记录数目 $\overline{|N_{\text{user}}|}$ 的比例表示用户的新老程度,使用该项目的用户数目 $|N(i)|$ 占项目平均被使用次数 $\overline{|N_{\text{item}}|}$ 的比例表示项目的流行程度,两比例之和等于 2 时,该用户对该项目的偏好因子 α_{ui} 值为 1;当比例之和小于 2 或者大于 2 时,偏好因子 α_{ui} 值小于 1,比例之和与 1 之差的绝对值越大,偏好因子越小。得到偏好因子后,将用户对项目的兴趣度乘上偏好因子 α_{ui} 作为改进后的兴趣度,从而对每个用户的兴趣度进行了针对性的缩放。详细步骤见算法 2。

算法 2　基于偏好因子的兴趣度缩放算法

输入　项目的综合相似度矩阵 $\boldsymbol{W'}$ 和用户-项目矩阵 $\boldsymbol{R}$

输出　用户兴趣度最大的 N 个项目

(1)计算用户 u 对项目 i 的偏好因子 α_{ui}:

$$\alpha_{ui}=(1-\alpha)+\frac{\alpha}{\exp(|\frac{|N(u)|}{\overline{|N_{\text{user}}|}}+\frac{|N(i)|}{\overline{|N_{\text{item}}|}}-2|)} \tag{8}$$

其中,α 为偏好置信度参数;$\overline{|N_{\text{user}}|}$ 为用户平均浏览记录数目;$\overline{|N_{\text{item}}|}$ 为项目平均被浏览次数。

(2)计算项目 j 传递给项目 i 的用户兴趣度 q_{ij}:

$$q_{ij}=\begin{cases}w'_{ji}\times r_{uj} & j\in N(u),i\in S(j,k)\\ 0 & \text{else}\end{cases} \tag{9}$$

其中,$S(j,k)$ 为与项目 j 最相似的 k 个项目集合。

(3)计算用户 u 对项目 i 兴趣度 p_{ui}:

$$p_{ui}=\alpha_{ui}\times\sum_{j\in N(u)}q_{ij} \tag{10}$$

(4)选取用户兴趣度最大的 N 个项目推荐给用户。

4　实验与结果分析

4.1　数据集

实验采用标准测试数据集——MoviesLens (http://movilens.umn.edu/),其中包括 6 040 位用户对 3 900 部电影的 10^6 条评分记录详细信息。在该数据集中,每个用户至少有 20 条评分记录。数据预处理时,将用户有评分的电影加到用户的播放记录中。

4.2　评估标准

实验采用 10 折交叉验证法,将实验数据集分为 10 份,每次取 1 份不同的数据作为测试集,剩余 9 份作

为训练集,取 10 次实验结果的平均值作为最终结果。

对参与比较的算法分别实验并比较它们的推荐指标:准确率计算公式为式(11),召回率计算公式为式(12),覆盖率计算公式为式(13)。

$$Precision = \frac{\sum_{u \in U} | R(u) \cap T(u) |}{\sum_{u \in U} | R(u) |} \tag{11}$$

$$Recall = \frac{\sum_{u \in U} | R(u) \cap T(u) |}{\sum_{u \in U} | T(u) |} \tag{12}$$

$$Coverage = \frac{\cup_{u \in U} R(u)}{I} \tag{13}$$

其中,$R(u)$为用户 u 推荐列表;$T(u)$为测试集中用户 u 的实际点播记录集合;U 为用户集合;I 为项目集合。准确率和召回率越大说明推荐越准确,覆盖率越大说明算法发现项目长尾[14]的能力越强。

4.3 结果分析

4.3.1 调节参数实验

偏好置信度参数 α 是为了调整偏好因子对用户兴趣度的影响。邻居数目 k 取 80,当 α 在[0,1]间取值时,分析其对算法推荐效果的影响。

结果如图 1 所示,ItemCF-SEIDS 算法准确率和召回率在 $\alpha = 0.51$ 左右最大,覆盖率随着 α 增大而增大。综合来看,$\alpha = 0.51$ 左右推荐效果最好。

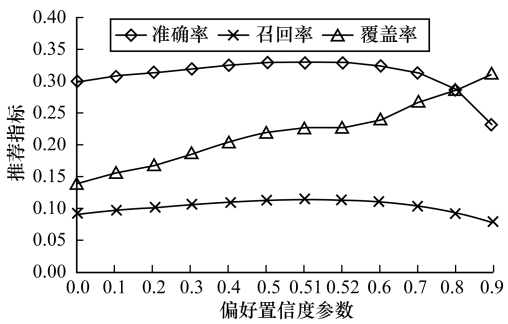

图 1 邻居数目为 80 时 α 对算法的影响

4.3.2 推荐效果实验

本节将 ItemCF-SEIDS 算法、ItemCF 以及 ItemCF 系列改进算法 ItemCF-IUF 和 ItemCF-Norm 在不同邻居数目条件下的推荐效果进行了对比。其中,ItemCF-SEIDS 算法的偏好置信度参数 α 取上节讨论的最优值 0.51。结果如图 2 ~ 图 4 所示。图 2 ~ 图 4 分别表明,ItemCF-SEIDS 算法在不同的邻居数目情况下,准确率、召回率和覆盖率比其他3 种算法都要高。其中覆盖率提升最为明显,从而有效减弱了推荐列表的长尾效应。本文实验推荐列表长度为 10,经验证算法的推荐效果随着推荐列表长度的增加,优势更加明显。限于篇幅,该对比实验未列出。

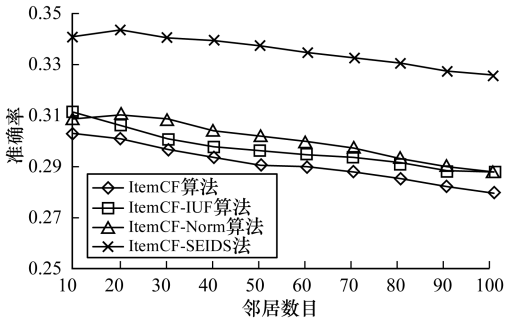

图 2 算法准确率对比

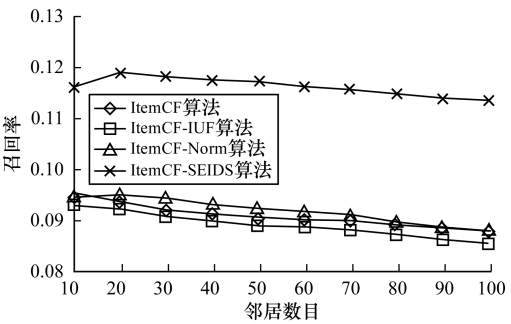

图 3 算法召回率对比

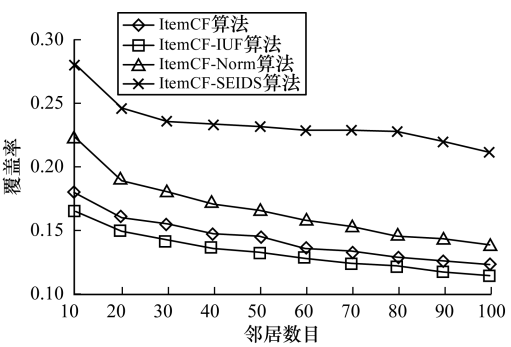

图 4 算法覆盖率对比

5 结束语

ItemCF-SEIDS 算法在基于项目的协同过滤算法框架基础上,针对用户潜在厌恶信息和新老用户兴趣差异性问题进行了改进。与其他改进算法或者混合算法相比,它在保证时间空间复杂度、更新实时性与经典基于项目的协同过滤算法基本相同的前提下,明显提高了推荐结果准确率、召回率和覆盖率,同时保留了基于邻域方法的可解释特性。ItemCF-SEIDS 算法还可以应用在混合推荐算法中,代替其对应的基于项目的协同过滤算法部分,其推荐效果将能得到更好的提升,同时算法时空复杂度也不会有明显增加。进一步地,可以把评分信息和项目流行度随时间变化的信息应用到改进点中,推荐效果能获得较好的提升。

(下转第 209 页)

方便了参数的理解和设置。通过此自适应机制,本文算法在运行时可以根据数据的分布情况,自动调节 DBSCAN 参数的值,从而发现更好的自然簇。实验结果表明,OS-DBSCAN 可以发现任意形状、大小和密度的簇,并且对簇内不均匀、簇间重叠有一定的聚类能力,同时提高聚类算法的准确率。下一步工作将对算法进行改进,使其能通过分析数据自身的情况自动选择合适的簇类个数 k,并将增量数据动态增加到数据集中,避免重复计算整个 *distTable*,降低运行时间。

参考文献

[1] 孙吉贵,刘　杰,赵连宇. 聚类算法研究[J]. 软件学报,2008,19(1):48-61.

[2] Chen M S,Han Jiawei,Yu P S,et al. Data Mining:An Overview from a Database Perspective [J]. IEEE Transactions on Knowledge and Data Engineering,1996,8(6):866-883.

[3] Jiang Hua,Li Jing,Yi Shenghe,et al. A New Hybrid Method Based on Partitioning-based DBSCAN and Ant Clustering[J]. Expert Systems with Applications,2011,38(8):9373-9381.

[4] MacQueen J. Some Methods for Classification and Analysis of Multivariate Observations[C]//Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley,USA:University of California Press,1967:281-297.

[5] Ester M,Kriegel H P,Sander J,et al. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise [C]//Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. Portland,USA:AAAI Press,1996:226-231.

[6] Liu Peng,Zhou Dong,Wu Naijun. VDBSCAN:Varied Density Based Spatial Clustering of Applications with Noise[C]//Proceedings of International Conference on Service Systems and Service Management. Washington D. C.,USA:IEEE Press,2007:1-4.

[7] 杨亚军,张坤龙,杨晓科. 基于变化密度的自适应空间聚类方法研究[J]. 计算机工程,2014,40(8):58-63,69.

[8] 陈　刚,刘秉权,吴　岩. 一种基于高斯分布的自适应 DBSCAN 算法[J]. 微电子学与计算机,2013,30(3):27-30.

[9] 汪　中,刘贵全,陈恩红. 一种优化初始中心点的 K-means 算法[J]. 模式识别与人工智能,2009,22(2):299-304.

[10] Karypis G,Han E H,Kumar V. Chameleon:Hierarchical Clustering Using Dynamic Modeling [J]. Computer,1999,32(8):68-75.

[11] 马儒宁,王秀丽,丁军娣. 多层核心集凝聚算法[J]. 软件学报,2013,24(3):490-506.

编辑　金胡考

(上接第 202 页)

参考文献

[1] 刘建国,周　涛,汪秉宏. 个性化推荐系统的研究进展[J]. 自然科学进展,2009,19(1):1-4.

[2] 项　亮. 推荐系统实践[M]. 北京:人民邮电出版社,2012.

[3] 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 Press,2001:285-295.

[4] Breese J S,Heckerman D,Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering [C]//Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. [S. 1.]:Morgan Kaufmann Publishers Inc.,1998:43-52.

[5] Karypis G. Evaluation of Item-based Top-n Recommendation Algorithms [C]//Proceedings of the 20th International Conference on Information and Knowledge Management. New York,USA:ACM Press,2001:247-254.

[6] 陈天昊,帅建梅,朱　明. 一种基于协作过滤的电影推荐方法[J]. 计算机工程,2014,40(1):55-58,62.

[7] 陈博文,刘功申,张浩霖,等. 融合标签传播和信任扩散的个性化推荐方法[J]. 计算机工程,2014,40(12):33-38.

[8] 韦素云,业　宁,杨旭兵. 结合项目类别和动态时间加权的协同过滤算法[J]. 计算机工程,2014,40(6):206-210.

[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.

[10] 熊忠阳,刘　芹,张玉芳,等. 基于项目分类的协同过滤改进算法[J]. 计算机应用研究,2012,29(2):493-496.

[11] 王　茜,段双艳. 一种改进的基于二部图网络结构的推荐算法[J]. 计算机应用研究,2013,30(3):771-774.

[12] 高全力,高　岭,杨建锋,等. 融合影响因子的加权协同过滤算法[J]. 计算机工程,2014,40(8):38-42.

[13] Wikipedia. 倒排索引[EB/OL]. (2013-03-12). http://zh.wikipedia.org/wiki/%E5%80%92%E6%8E%92%E7%B4%A2%E5%BC%95.

[14] Wikipedia. 长尾效应[EB/OL]. (2015-01-11). http://zh.wikipedia.org/wiki/%E9%95%BF%E5%B0%BE.

编辑　索书志