**引入兴趣稳定性的时间敏感协同过滤算法**

1.山东理工大学 计算机科学与技术学院,山东 淄博 255049

[College of Computer Science and Technology](https://www.baidu.com/link?url=6_Wjpm8Po16wMbB6r00ZEOMqx9gdC_xUNIo7PrEckDVCk2iltkPpTJkuffqt3nOV2m3Dg3uxIrINyWGrdbo5Bp4BuU3r_lQM6Mr0b4K8huBIac3cuf1tpGDsCuz9mD7Dqa_uvTPStURabxgmVlf3LyRNfGpnW5LHIEEodde9QCi&wd=&eqid=e618908600056d820000000458817ed8),Shandong University of Technology, Zibo 255049,China

.Collaborative Filtering Algorithm

**Abstract**：

**Key words:**time-sensitive; stability of interest; prevalence of item; collaborative filtering; popularity bias

摘 要：针对物品流行偏置现象,将物品流行度引入到用户兴趣中建模，提出了基于物品流行度的用户兴趣特征相似度模型。针对传统模型没有考虑到用户兴趣稳定性和难以实时捕获用户兴趣问题,在计算用户兴趣相似度过程中引入时间敏感和兴趣稳定性，提出了引入时间敏感的用户兴趣稳定性的相似度模型。最后，融合权重因子将两种相似度模型线性加权，提出了一种新颖的算法模型。实验表明, 该算法比传统协同过滤算法在平均绝对误差（MAE）和均方根误差（RMSE）上均有明显降低的同时也能挖掘出长尾物品，缓解了物品流行偏置现象。

**关键词** ：时间敏感；兴趣稳定性；物品流行度；协同过滤；流行偏置

**1 引言**

《中国互联网络发展状况统计报告》 显示,截至2015年6月，中国互联网网民规模达6.68亿[1]。伴随着信息技术飞速发展，信息量也发生着爆炸式的增长，信息过载问题也日益突出。在海量信息中用户很难发现自己需要的，感兴趣的部分。而解决这一问题的技术通常有搜索引擎，信息过滤技术，推荐系统等。但搜索引擎主要是在用户有明确需求的情况下,通过用户输入关键字的方式挖掘出用户需求或者用户感兴趣的部分。推荐系统是在用户没有明确需求的情况下，通过分析用户的历史行为数据给用户进行兴趣建模，然后给用户进行推荐。

协同过滤推荐算法主要分为两类:基于用户的协同过滤和基于物品的协同过滤[2]。基于物品的协同过滤推算法的思想是根据用户的历史行为数据分析计算得出用户的行为偏好，据此用户偏好给用户做出推荐。这种推荐算法前提假设是，在未来的一段时间内用户的兴趣偏好是不变化的。基于用户的协同过滤推荐算法的思想是根据用户A的行为数据计算出和用户A具有相似偏好的邻居用户，然后将具有相似偏好的邻居用户感兴趣的物品且用户A还没有发现的物品推荐给用户A。可见，不管是哪一种协同过滤推荐算法，其核心就是相似度的计算。

针对协同过滤技术中的动态性问题，文献[3]加权时间权重和资源相似度的数据权重，提出了适应用户兴趣的推荐算法。文献[4]提出了一种动态推荐技术。文献[5]随着移动设备蓬勃发展，提出了一种SUCM模型来学习细粒度的用户偏好给用户做推荐。

针对协同过滤技术中的可扩展性问题传统的解决思路一般有两种。第一种思路是以聚类技术为代表的。文献[6]提出为每个聚类设置一个代表用户，通过计算和代表用户的相似性来选择近邻用户集合。文献[7]通过深度学习方法来解决可扩展性问题。文献[8]提出了一种协作深度学习的分层次贝叶斯网络来解决数据稀疏等问题。

第二种思路为通过对评分矩阵降维来解决可扩展性问题[9]。文献[10]提出可利用统计学中的主成分分析技术来实现评分矩阵降维。

针对冷启动问题，文献[11]提出了运用近邻度，影响力和普及性三方面综合考虑用户评分对用户之间相似度的影响。文献[12]提出了一种新型的启发式相似度计算模型，可以缓解用户冷启动问题，提高了用户之间的相似度。文献[13][14][15]通过使用自编码方式来解决冷启动问题，提高topN推荐质量。文献[16]利用社交网络，分析社交群体中的强弱关系，提出一种EM算法来提高推荐质量。文献[17]利用深度学习技术，将内容与用户之间的交互生成一个学习函数，将用户偏好转换成排名给用户推荐。

针对物品流行偏执问题，文献[18]通过改变推荐结果的流行度分布来缓解的。而本文通过将所有物品流行度进行“装箱化”，然后根据用户实际评分去映射“箱子”，映射完成之后，生成一个三维的基于物品流行度的用户兴趣特征向量-“向量化”，然后计算用户之间的相似度。

**2方法介绍**

一般物品流行度越高，用户会对它越感兴趣，即用户兴趣与物品的流行度有关。本文将物品的流行度作为用户兴趣特征来构建模型，提出了基于物品流行度的用户兴趣特征相似度模型。本文认为用户短期兴趣不一定是不变的，这需要根据用户兴趣的稳定性来确定，若用户兴趣稳定，那么随着时间的流逝，用户兴趣不变化，若用户兴趣不稳定，则兴趣会发生变化。本文从兴趣度和兴趣稳定性从发,提出了一种时间敏感的用户兴趣稳定性的协同过滤推荐算法。

**2.1 基于物品流行度的用户兴趣特征相似度模型**

**定义1**（物品的流行度定义）所有用户对物品的评价次数之和与物品总数的比值。公式如下：

 （1）

其中，表示有多少个用户对物品进行了评价，表示物品总数。

算法描述：

（1）取三个物品流行度区间，，。

（2）用公式计算用户已经评价的所有物品的流行度。

（3）将步骤（2）中用户计算的所有物品的流行度先装箱然后映射到步骤（1）。

装箱伪代码为：

 1 





















说明：，，，，，是根据实验取得的阈值数据。

映射伪代码为：









































（4）将步骤（3）中生成的用户特征向量采用余弦相似度公式来计算用户A和用户B的基于物品流行度的用户兴趣特征相似度。公式描述如下:

（2）

上述模型称为Item\_pop\_sim模型，简称IPS模型。

**2.2 引入兴趣稳定性的时间敏感相似度模型**

在实际应用过程中，用户的兴趣通常具有易变性，不仅与用户对物品评分数值有关，还与物品的流行度有关，这两者加权构成了用户的兴趣度。用户兴趣度定义如下：

**定义2**（用户的兴趣度）是由用户对个物品的兴趣向量组成的向量集合。

**定义3**（用户的第个物品兴趣向量）用户评价的第个物品的实际评分值与满分值的比值再加权该物品的流行度。公式描述如下：

 （3）

（4）

其中，表示用户对第个项目的兴趣程度，表示用户对物品的打分值，表示该物品的满分值，表示公式（1）中定义的物品的流行度，表示所有物品的最大流行度，表示所有物品的最小流行度。，表示参数，且，该参数可以通过实验验证得出。同样，使用余弦相似度来计算用户和用户之间的兴趣相似度，公式描述如下：

 （5）

该模型通过引入带权值的物品流行度可以有效的缓解用户打分偏置的问题。

**定义4**（用户的兴趣稳定性）用户评价的所有物品评分数值的方差。公式描述如下：

 （6）

 （7）

其中，表示用户评价的第个物品的评分值，为用户评价的所有物品的总数，为用户评价的所有物品评分值的平均值。方差的大小来衡量用户兴趣的稳定性，即方差越小用户兴趣越稳定。

用户兴趣的影响因素：自身，时间和环境。现实中，用户的兴趣往往不是一成不变的，可能受自身因素的影响，周围环境，亲朋好友影响，兴趣会随着时间的流逝潜移默化，很早之前的兴趣可能会逐渐淡忘或者消失。如果用户的兴趣是比较稳定的，本文认为用户随着时间的变化兴趣不会有太大的变化。因此从影响用户兴趣的因素出发，给出了时间敏感的表征。

**定义5**（时间敏感）在两个用户兴趣稳定的基础上，两用户对物品的评分时间越相近则用户间的兴趣相似度就越高，即用户的兴趣相似度对时间敏感。

综上，本文提出了一种引入用户兴趣稳定性的时间敏感的相似度计算模型。公式描述如下：

 （8）

其中，，，和分别表示用户和用户分别去中心化后的评分方差，是指评分方差的中值，通过图7（横坐标表示用户ID，纵坐标表示用户评分方差）分析可以得出，方差大部分在0.5到1.5之间，所以，表示实验取得的参数，其中，。表示用户和用户共同评分的物品交集，和分别表示用户和用户对物品的评价时间。表示实验取得的参数，其中，。和表示用户和用户对物品评分值的平方和。上述公式（8）中的模型称为Stability\_Time\_Sim模型，简称为STS模型。

**2.3 两种相似度模型的融合**

上面介绍了两种相似度模型各有各的优点，其中基于物品流行度的用户兴趣特征相似度模型可以有效的缓解物品偏置问题，引入时间敏感的用户兴趣稳定性的相似度模型可以实时抓住用户兴趣，挖掘长尾物品，提高推荐系统的新颖度。所以为了使推荐效果更好，我们将两种模型进行线性加权，提出了函数模型。公式描述如下：

（9）

上述模型简称模型，模型综合考虑了用户兴趣的稳定性，时间敏感情况，以及物品流行度等因素。综合这些因素进行建模，实验表明，加权后的相似度模型在推荐质量上有明显提高。

**3实验设计及分析**

**3.1 实验数据集**

本文实验所用的数据集是美国Minnesota大学GroupLens小组开发的MovieLens站点所提供的数据集。MovieLens建立于1997年，是一个基于Web的推荐系统，目前，该站点提供三种不同数量级的数据集，分别为：943个用户对1682部电影的10万条评分的数据；6040个用户对3900部电影评分的100万条数据；71567个用户对10681部电影做出的1000万条评分数据。

本文实验采用了943个用户对1682部电影的10万条评分数据集，其中每个用户至少对20部电影进行了评价。该数据集的稀疏等级为，本文将实验数据集划分为训练集和测试集，其中训练集占80%，测试集占20%。

**3.2 评价标准**

本文采用的推荐质量的评价标准分别是平均绝对误差（Mean Absolute Error， 简称MAE）和均方根误差（Root Mean Squared Error，简称RMSE）[19]。平均绝对误差(MAE)公式如下：

 （10）

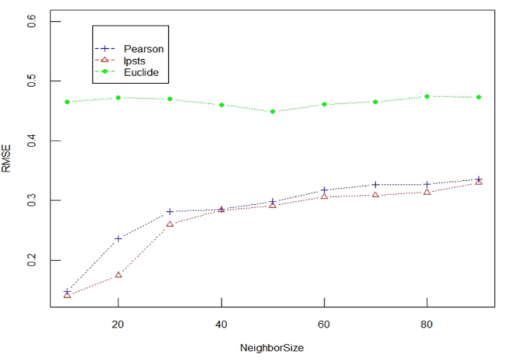
均方根误差（RMSE）公式如下：

 （11）

**3.3 实验结果**

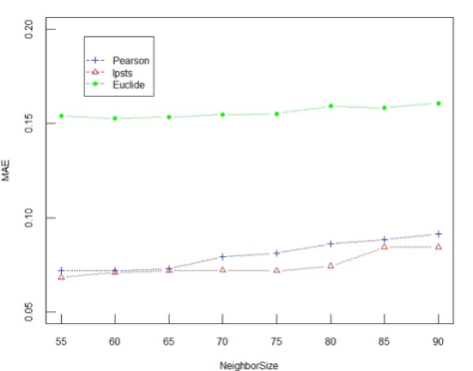
实验比较了模型与Pearson（皮尔森）模型以及Euclide（欧几里得）模型在推荐质量上的差异，第一步先考查当参数和参数固定时，随着邻居数量逐渐增多，三种模型在RMSE和MAE两个评价标准上的效果。第二步固定邻居数量，考查当参数和参数以及参数按照相应的步长逐渐增长时，三种模型在RMSE和MAE两个评价标准上的效果。

**3.3.1** **模型实验结果**



**图1**  IPSTS模型与Pearson模型以及Euclide模型RMSE对比图

图1（横坐标表示选取的邻居数量，纵坐标表示均方根误差）显示出取不同邻居时，各个相似度计算模型对RMSE的影响。其中邻居数量分别为10，20，30，40，50，60，70，80，90。分析得出，模型计算的均方根误差（RMSE）比Euclide和Pearson模型计算的均方根误差（RMSE）都要低（例如：当邻居数量为20时，本文提出的模型比Pearson模型要低6%左右，比Euclide模型要低30%左右）。所以，降低了误差，提高了推荐质量。

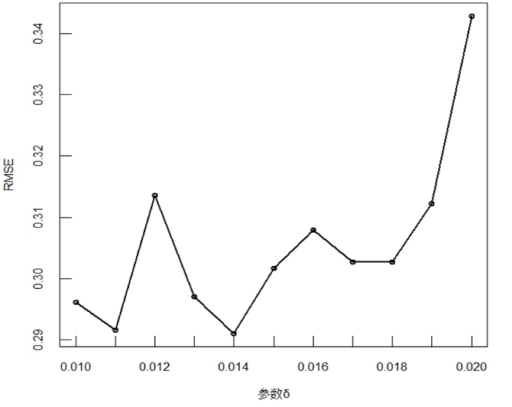


**图2** IPSTS模型与Pearson模型以及Euclide模型MAE对比图

图2（横坐标表示选取的邻居数量，纵坐标表示平均绝对误差）显示出取不同邻居时各个相似度计算模型对MAE的影响折线图。其中邻居数量分别为55，60，65，70，75，80，85，90。分析得出，模型计算的平均绝对误差（MAE）比Euclide和Pearson模型计算的平均绝对误差（MAE）都要低（例如：当邻居数量为80时，模型比Pearson模型要低1%左右，比Euclide模型要低8%左右）。所以，提高了推荐质量。

### **3.3.2 参数对****模型的影响**

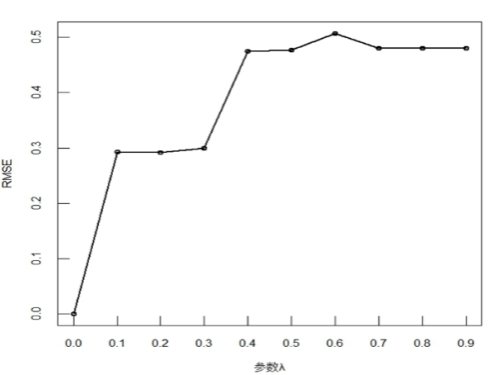
在实验中，参数的选取是至关重要的，为了让实验达到满意的效果，同时充分测试算法的健壮性和适用性，更为了优化算法，本节对取值选取进行了实验，为了尽可能的排除其他因素干扰，仅仅检验对模型的影响，我们取邻居数量为50，参数取0.2时，检验随着的递增，模型在RMSE评价标准上的效果。



**图3** 参数变化对IPSTS模型的影响

### **3.3.3 参数对****模型的影响**

同时，为了检验权重参数对模型的影响，结合上一小节实验结果我们取邻居数量为50，参数取0.014时，检查随着的增长，模型在RMSE评价标准上的效果。

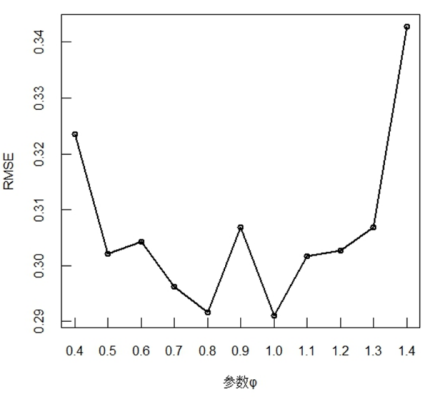


**图4** 参数变化对IPSTS模型的影响

图4（横坐标表示选取的参数，纵坐标表示均方根误差）显示，从0.1开始，随着的增长，RMSE是先减少后增长，然后增长到一定的数值后再下降，之后，随着的增长，RMSE基本上趋于缓和。由图4可知，当取0.2时，RMSE最小。所以，在模型中，参数取0.2最优。

### **3.3.4 参数****对****模型的影响**

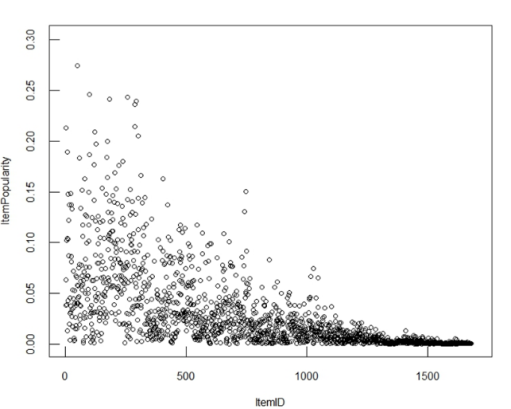
为了检验参数对模型的影响，结合上一节实验，邻居数量固定为50，取0.2，取0.014。



**图5** 参数对IPSTS模型的影响

图5（横坐标表示选取的参数，纵坐标表示均方根误差）显示，当取1.0时，RMSE最小。因此，在模型中，参数取1.0最优。

### **3.3.5** **模型对长尾物品的影响**



**图6**  物品ID与物品流行度散点图

由图6可知，这些物品存在长尾现象。其中，物品ID越靠前，流行度越高，ItemID在1000之后的为长尾物品。为检验模型对长尾物品的挖掘能力，实验如下：随机选取三个用户，对三个用户分别进行top5（推荐前5个）推荐，用户ID分别为80，800，888。userID为需要进行推荐的用户编号，RecommenderSize=5为推荐列表大小，NeighborSize（邻居数量）=90。Pearson模型中RMSE=0.33560，模型中RMSE=0.32965。

表1 userID为80时， Pearson模型与IPSTS模型推荐的物品比较

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 推荐的物品ID | | | | | |
| Pearson | 113 | 945 | 224 | 30 | 408 |
| IPSTS | 634 | 1467 | 1189 | 114 | 115 |

表1（ItemID为推荐的物品编号）显示，模型在降低了RMSE（均方根误差）的基础上，提高了推荐质量的同时，能够挖掘出物品编号为1467,1189的长尾物品给用户进行推荐，对长尾现象起到了一定的缓解，可以给用户带来惊喜。

表2 userID为800时， Pearson模型与IPSTS模型推荐的物品比较

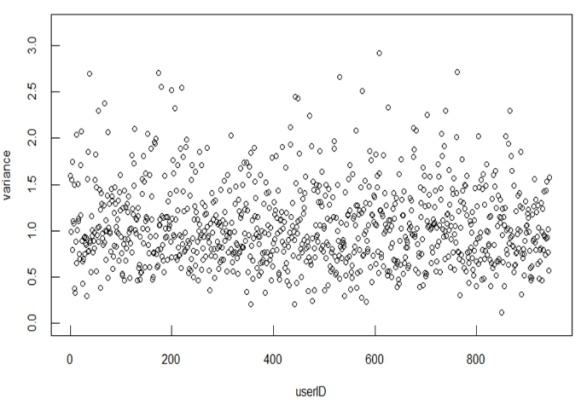
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 推荐的物品ID | | | | | |
| Pearson | 113 | 30 | 656 | 493 | 64 |
| IPSTS | 1159 | 1121 | 524 | 1103 | 114 |

同样，表2（ItemID为推荐的物品编号）显示，采用Pearson模型推荐的物品列表没有长尾物品推荐出来，而模型在降低了RMSE（均方根误差）基础上，挖掘出了1159，1121，1103三个长尾物品进行推荐。不仅缓解了长尾现象，而且挖掘出的长尾物品能够给用户带来新鲜感。

表3 userID为888时,Pearson模型与IPSTS模型所推荐物品的比较

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 推荐的物品ID | | | | | |
| Pearson | 641 | 30 | 113 | 522 | 656 |
| IPSTS | 1467 | 1368 | 1512 | 615 | 169 |

表3（ItemID为推荐的物品编号）显示，在给编号为888的用户进行推荐时，模型在降低了RMSE（均方根误差）的基础上，给用户做出的推荐策略为1467， 1368，1512，615，169。其中的前三个物品为长尾物品。但是，Pearson模型不仅均方根误差比模型要高，而且给出的推荐策略也没有挖掘出长尾物品。因此，模型对长尾现象有所缓解，能够挖掘出长尾物品给用户进行推荐。



**图7**  用户ID与用户评分方差散点图

**4 结束语**

推荐系统是在用户没有明确需求的情况下，从海量的数据中帮助用户寻找感兴趣的信息，进而以合适的方式给用户展现和推荐。本文围绕降低预测评分误差以及挖掘长尾物品，缓解物品流行偏置问题，引入兴趣稳定性，同时关注时间敏感因子，构建了引入兴趣稳定性的时间敏感相似度融合模型。实验表明该模型能够实现用户兴趣的实时动态推荐，且进一步提高了推荐质量。然而，影响用户兴趣的因素还包括外部环境因素等其它关键因素，随着移动设备的蓬勃发展和定位系统的成熟，可以将这些因素引入到模型中。其次，融合后的模型是线性加权的，下一步还可以考虑构建一个非线性模型进行推荐。

**参考文献**

1. 佚名. 第36次《中国互联网络发展状况统计报告》[J]. 网络传播, 2015(7):54-59.
2. Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of netnews[J]. Working Paper Series, 2015:175--186.
3. 邢春晓,高凤荣,战思南.适应用户兴趣变化的协同过滤推荐算法[J].计算机研究与发展,2007.44(2): 296-301.
4. 项亮. 动态推荐系统关键技术研究[D].北京:中国科学院自动化研究所,2011.
5. Liu B, Wu Y, Gong N Z, et al. Structural Analysis of User Choices for Mobile App Recommendation[J]. Acm Transactions on Knowledge Discovery from Data, 2016, 11(2):17.
6. Rashid A M, Lam S K, Karypis G, et al. ClustKNN: a highly scalable hybrid model-& memory-based CF algorithm[C]// The Workshop on in Proc of Webkdd. 2006.
7. Elkahky A M, Song Y, He X. A Multi-View Deep Learning Approach for Cross Domain User Modeling inRecommendationSystems[C]//TheInternationalConference. 2015:278-288.
8. Wang H, Wang N, Yeung D Y. Collaborative Deep Learning for Recommender Systems[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2014:1235-1244.
9. Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, Usa, August. 2008:426-434.
10. Kim D, Yum B J. Collaborative filtering based on iterative principal component analysis[J]. Expert Systems with Applications An International Journal, 2005, 28(4):823-830.
11. Ahn H J. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem[J]. Information Sciences, 2008, 178(1):37-51.
12. Liu H, Hu Z, Mian A, et al. A new user similarity model to improve the accuracy of collaborative filtering [J]. Knowledge-Based Systems, 2014, 56(3):156-166.
13. Li S, Kawale J, Fu Y. Deep Collaborative Filtering via Marginalized Denoising Auto-encoder[C]// The, ACM International. 2015:811-820.
14. Wu Y, Dubois C, Zheng A X, et al. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems[C]// ACM International Conference on Web Search and Data Mining. ACM, 2016:153-162.
15. Strub F, Mary J, Gaudel R. Hybrid Recommender System based on Autoencoders[J]. 2016:11-16.
16. Wang X, Lu W, Ester M, et al. Social Recommendation with Strong and Weak Ties[C]// The, ACM International. 2016:5-14.
17. Vuurens J B P, Larson M, De Vries A P. Exploring Deep Space: Learning Personalized Ranking in a Semantic Space[J]. 2016:23-28.
18. Adomavicius G, Kwon Y O. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques[J]. Knowledge & Data Engineering IEEE Transactions on, 2012, 24(5):896-911.
19. Herlocker J L. Evaluating collaborative filtering recommender systems[J]. Acm Transactions on Information Systems, 2004, 22(1):5-53.