

2022

基于时间序列的用户协同过滤推荐算法

Team：我想试一试



1

本算法基于Python 3.9.12开发

2

分析数据采用上海市图书馆2019年全年用户与借阅数据，其中书目数据354万余条、读者数据10万余条、以上两者所产生的流通数据623万余条。

3

构建基于时间序列的用户协同过滤图书推荐算法。

为了减小部分偏差数据的影响、并减小计算量，对同一读者多次借阅同一书籍的行为记作一次，并去除归还记录数据（即不考虑借阅时间长短对最终推荐度的影响），最终得到1955509条数据并纳入模型计算，数据如下图所示。

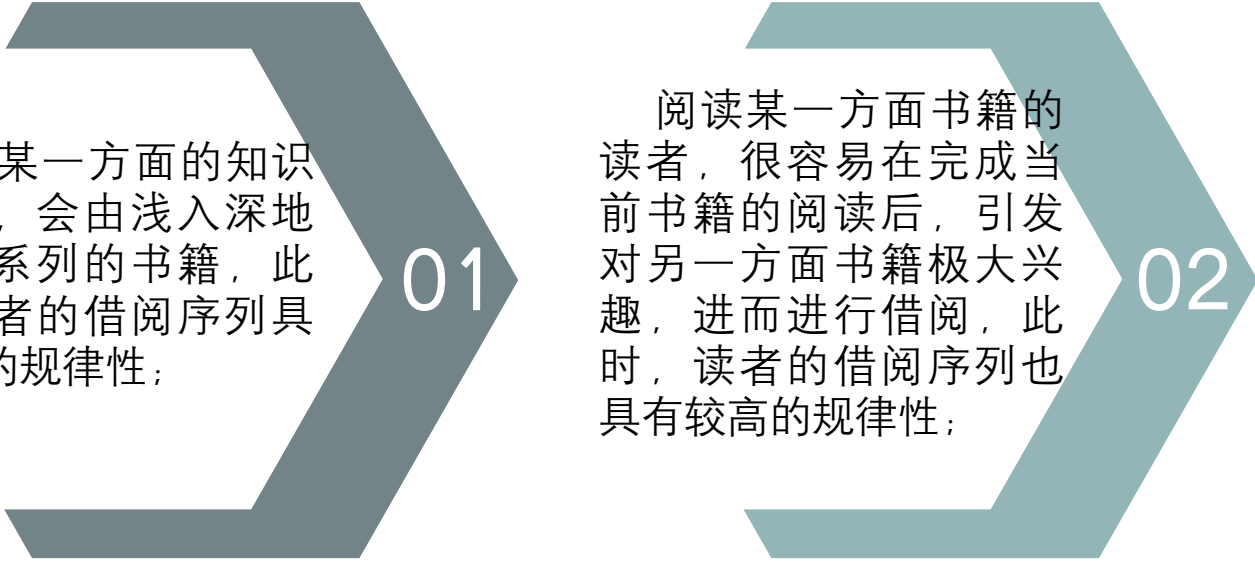
	date	time	location	action	book_id	borrower_id
0	20190101	160500	G17	CKO	395845	8164
1	20190101	160500	G17	CKO	343676	8164
2	20190101	160500	G17	CKO	350823	8164
3	20190101	160500	G17	CKO	369924	8164
4	20190101	160500	G17	CKO	390827	8164
...
1955504	20191203	150300	G12	CKO	268395	59101
1955505	20191203	150300	G12	CKO	341874	59101
1955506	20191203	150300	G12	CKO	428123	59101
1955507	20191203	151700	J125	CKO	218917	5315
1955508	20191203	151700	J125	CKO	91597	5315

1955509 rows × 6 columns

传统的协同过滤算法分为两类：基于用户的(User-based)协同过滤、基于物品的 (Item-based)协同过滤。这两种方法分别围绕用户、物品，首先计算用户之间、物品之间的相似度，继而根据相似度评估结果找到用户或物品的最近邻，再通过最近邻实现物品的推荐。

然而，上述传统算法在计算临近关系时，仅从集合的角度出发，而不会考虑时间、空间的因素带来的影响。而本文基于传统的基于用户的协同过滤算法，增加**借阅行为的时间序列因素**，改进得到考虑时间序列的基于用户的协同过滤推荐算法。

从实际生活角度出发，在许多场景中，借阅的书籍都能找到时间序列上的规律及合理性，例如：

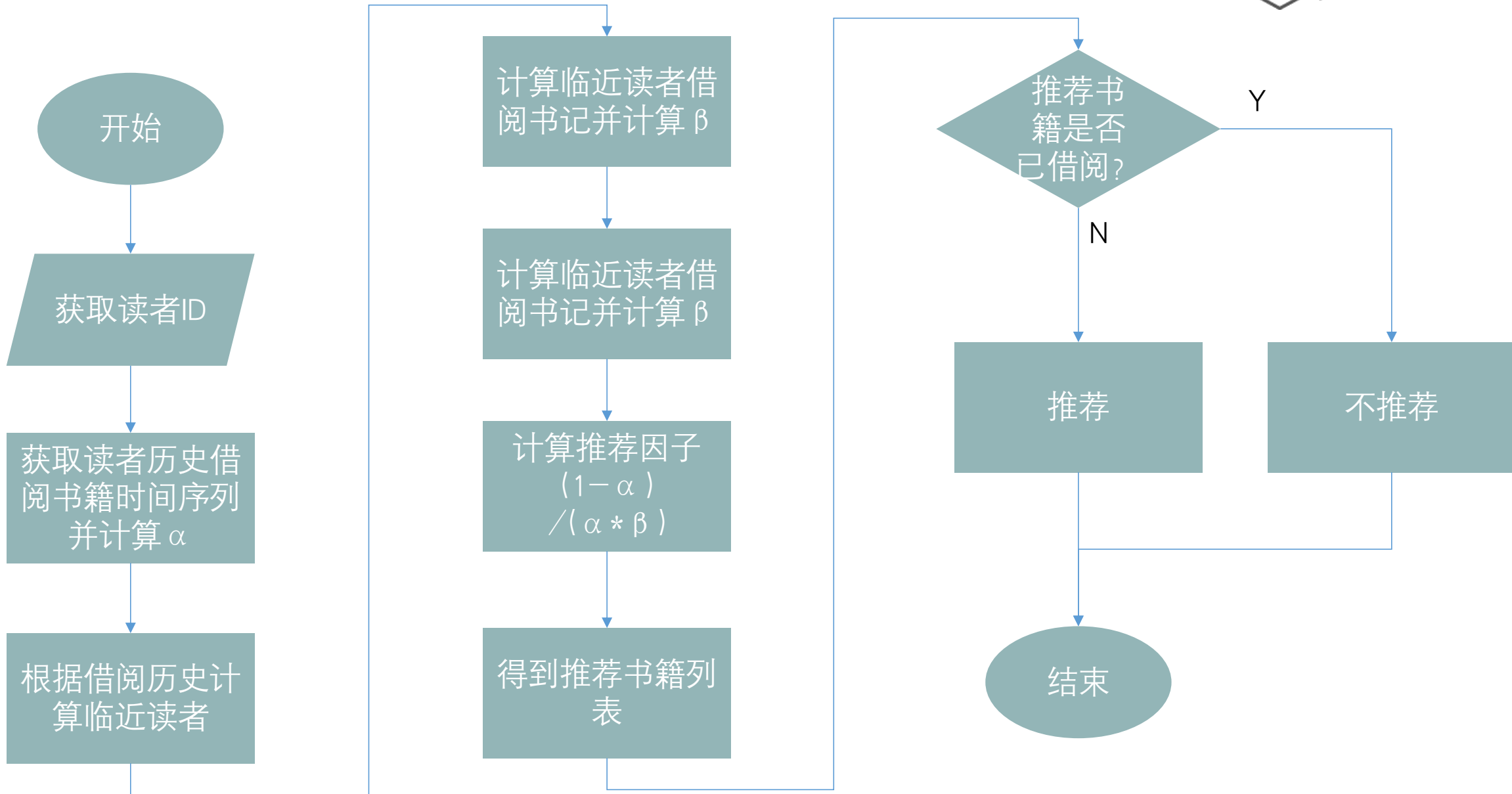


01

学习某一方面的知识的读者，会由浅入深地阅读一系列的书籍，此时，读者的借阅序列具有很高的规律性；

02

阅读某一方面书籍的读者，很容易在完成当前书籍的阅读后，引发对另一方面书籍极大兴趣，进而进行借阅，此时，读者的借阅序列也具有较高的规律性；



(1) 获取读者ID

(2) 计算读者借阅历史时间序列:

定义 a_i 为用户借阅书籍历史序列中第 i 本书推荐时间序列因子, 其计算公式为:

$$a_i = \frac{book_i - time_{min}}{time_{max} - time_{min}}$$

(3) 计算临近读者借阅书籍序列:

定义 β_{ij} 为用户借阅书籍历史序列中第 i 本书临近用户借阅历史序列中第 j 本书推荐时间序列因子, 其计算公式为:

$$\beta_{ij} = \frac{|book_j - book_i|}{time_{max} - time_{min}}$$

(4) 计算推荐因子:

定义 p_j 为最终推荐因子, 其计算公式为:

$$p_j = \sum \frac{1 - a_j}{a_j \beta_j}$$

(5) 得到推荐书籍列表:

根据最终的书籍列表及推荐因子进行排序, 推荐因子较高者为优先推荐书籍。

(6) 剔除已借阅书籍:

剔除列表中用户已经借阅过的书籍, 按推荐因子得到最终推荐列表。

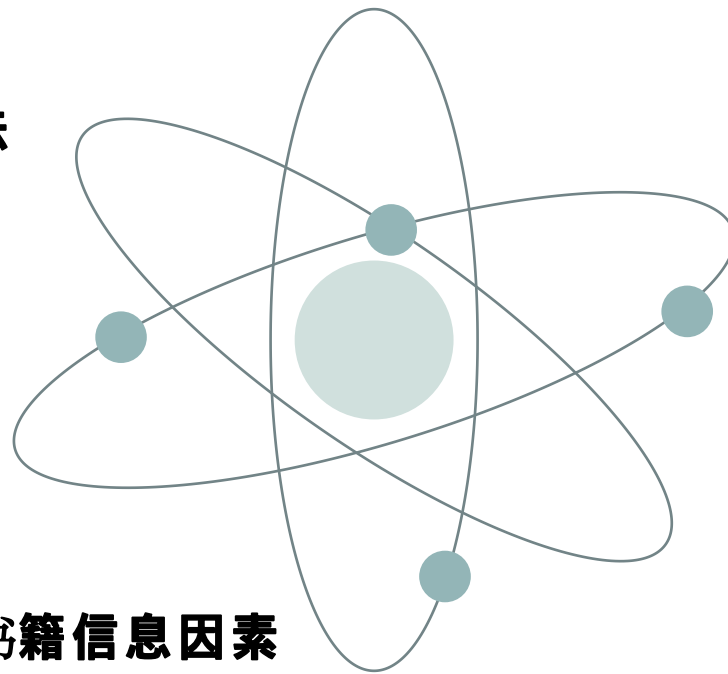
(7) 得到结果。

优化计算方法

由于协同过滤所需运算量较大，目前求邻近集算法有待优化，以提高计算速度。

增加书籍信息因素

目前推荐算法只考虑用户行为带来的影响，后续需要增加书籍因素。



考虑更多借阅行为因素

目前只考虑结束时间点、未考虑归还时间点以及总借阅时间带来的兴趣度影响。

.....
.....



1

首先，非常感谢组委会举办的本次比赛以及在过程中给予的支持，也很感激能对我给予这么高的评价与认可。

2

于个人而言，这次比赛也是对自己一次挑战，能够成功参赛并得到认可，对自己也是极大的激励。

3

从能力上讲，本次比赛带给我一次不错的锻炼机会，能够在数据处理、算法设计、程序设计等多个角度锻炼自身。

01

参赛选择

参赛最大的阻碍是自己，一个人也可以是一支队伍。

02

数据处理

数据库很庞大，数据质量也不算高，选择一个合适的角度切入是个不错的思路。

03

算法选取

合适的、能用好的才是好算法，不要一味追求过于高级的算法。

04

.....

.....

2022

感谢您的聆听

Team：我想试一试

