A Prediction Recommendation Algorithm Research Toward to the Latest Published Article

Tao Liu [1, a], Zuo Liu [1,b]

[1] North China Electric Power University, 071003 Baoding, China

[a]taoliu@ncepu.edu.cn, [b]1291224355@qq.com

Keywords: information exploded, recommendation algorithm, information retrieval, data mining,machine learning

Abstract: with the development of the Internet, people have entered an era of information explosion. A good user experience system which to screen and filter massive informationcan and to show the most interesting information in front of the readers greatly save the user's time.This paper presents a new algorithm based on the latest published articles of WeChat public platform.Sort the just published articles through the analysis of the author and the articles themselves potential. Recommend the most explosive potential articles which were going to have a high reading number to 10w+ to the public.The development of recommendation system is closely related to the problems and challenges it faces. It is still a hot research topic in information retrieval, data mining and machine learning.

1 Introduction

With the rapid development of the Internet, people entered the era of information explosion. In order to solve the contradiction between the complex needs of users and the huge data，the personalized recommendation system came into being, which is used more and more widely [1]. Personalized recommendation technology provides a variety of resources for users by studying the user's preferences and interests[2]. A good user experience system can predicted explosion point for user intelligently and display them in front of the user.

Occurrences of the search engines solve the problem of information filtering to a certain degree but not enough[3]. Search engines require users to actively provide keyword to filter vast amounts of information. Screening effect of the search engine will be greatly reduced when the user can not accurately describe their needs, and it is not an easy process that transformed user’s own needs and intentions into a keyword[4].

While the WeChat was only launched in 2011 by Tencent, but as a new media, its powerful influence has triggered widespread concern of academia and industry[5]. As one of the most popular social networking platform to push information, WeChat public platform to generate hundreds of millions of data by the user per day[6]. Massive data is a huge social benefits. Under this background, be able to providing intelligent recommendation for public from overload articles has become a challenging issue.

The problem to be solved in this article is proposing a recommendation algorithm named TNAPR(Toward to New Article Predicted Recommendation). For new articles just published,by analyzing their existing reading number and liking number or several other data, combined with analysis of the current article’s author, predict and recommend articles those reading number can be achieved 10w+. Section 1 is introduction which describes the recommended algorithm research background. Section2 describes the new recommendation algorithm named TNAPR. Section3 is the evaluate of the recommendation algorithm. Section4 is the full text summary. Article data using

real data crawling from the WeChat public platform.

2 The overall framework of TNAPR

For a just released article, whether it can become a cexplosion point article considers the impact the author of the article named *author_score*. The impact factor of the author is author's influce which named *author_score* and the probablity of hot article which named *ehot_rate*. The *author_score* depend on *read_num* and *like_num* of the previous article of the current author. The *ehot_rate* impact factor is more extensive as WeChat public platform's subscriptions named *sub_num*, the release time and the read number of every article named *release_time* and *read_num*. There will be analyzed in detail below.

Each new article predicted scores represented by *recommend_score*. The *recommend_score* is defined as the equation (1):

$$recommend\_score = auther\_score \times ehot\_rate \tag{1}$$

Wherein each symbol and the correspondence between influencing factors are as follows:

Table 1 Correspondence between the symbols of formulas and impact factor

Formula symbol	Impact factor
recommend_score	the recommend score of the current article
auther_score	author's influce
ehot_rate	the probablity of hot article

2.1 The author's influence

The thought of this strategy is the author of the influential ranking,who is the current article's author of the public platform of WeChat. On the influence of rank is to score each author. Each author has a score which named *author_score*, the greater the *author_score*, the higher the score, the greater the influence author. For *author_score* we were the fllowing conclusions:

Conclusion A:The reading numbers of each article of the current author is a contribution factor;

Conclusion B:The liking numbers of each article of the current author is a contribution factor also;

Conclusion C:The *author_score* inversely proportional to the all total number of articles of the current author.

Obviously, conclusion A and conclusion B are correct. The reading number and the liking number are two different factors. Compare with the liking number, the reading number has a greater persuasion, so that the weight of the liking number is relatively large. The greater the contribution factor, shows that the author more influential;

Conclusion C is based on:If the author has published a large number of articles,the reading numbers and the liking numbers of each article can contribuate to the current author's total reading numbers and total liking numbers even if they are only a small number,so that for the same total reading number or the same total liking number, the less the published article, the more influence;
So each author's influence score is shown as fllow equtio(2):

$$author\_score = \frac{\sum_{k \in \{author\_articles\}} i \times read\_num_k + j \times like\_num_k}{author\_article\_count} \tag{2}$$

2.2 The probablity of hot article

The hot article rate named *ehot_rate* is popular articles proportion in the all articles of the current author,the same as the number of popular articles and the number of the total articles ratio. Every article has a heat rate flag *ehot_flag*, and the heat rate flag of popular articles is 1, the heat

rate flag of on-popular articles is 0 in contrast. Therefore the definition formula of the *ehot_flag* is as follow eqaution(3). Where S is a constant, is the critical value of *ehot_flag* of each article:

$$ehot\_rate = \frac{\sum_{k \in \{author\_articles\}} ehot\_flag_k}{author\_article\_count} \tag{3}$$

Among them, the heat rate flag based on equation(4):

$$hot\_flag = \begin{cases} 1 & score > S \\ 0 & score \le S \end{cases} \tag{4}$$

For each article's score, we have the following few conclusions:

Conclusion A: The *score* inversely proportional to the time named *release_time* of the current published articles.

Conclusion B: The *score* inversely proportional to the article's public-owned number named *sub_num*.

Conclusion C: The *score* proportional to the reading number of the current article named *read_num*.

Conclusion D: The *score* proportional to the liking number of the current article named *like_num*.

Conclusion A is based on: Usually as time goes on, the articles of the public platform of WeChat get more and more attention, thus the earlier published article, if a greater amount of reading it, the more propagation factor.

Conclusion B is based on: If the public subscription number is large enough, even though it is common for an article hits are great. So that the same amount of reading the article, if the number of public-owned subscriptions smaller the factor has spread.

Conclusion C and D is a very natural conclusion, the greater the amount of reading or the number of thumbs, have proved more articles propagation factor.

So the *score* of each article can be calculated by equation(5):

$$score = \frac{read\_num \times like\_num}{sub\_num \times release\_time} \tag{5}$$

3 The simulation experiment and performance analysis

3.1The simulation experiment environment Settings

The experiments run on basis of python and MySQL. MySQL is an open source of relational data management system, which is characterized by fast and flexible enough to support the simulation experiments carried out. To testing the efficiency of the algorithm of this paper, the experimental environment is true and random. The simulation completely use the real data which is crawling from WeChat public platform. For the purpose of reflecting the efficiency of the algorithm in a real environment, set up a test environment and experiment as follows:

1)Apply for an account as a test number, and focus on 64 accounts of WeChat public platform. Experimental data all from a real social network.

2)Training data are collected from WeChat public platform within a month moreover test data set is the most active in the training set data to ensuring the effectiveness of data.

3)Write reptiles with python to obtaining the required data set then store in a MySQL database.

4)Measure the effectiveness of the algorithm by comparing statistical data and evaluation standard.

Visibly experiment maximize simulation and describe the application scenarios of real social networking by testing the real historical data of representative WeChat public platforms.

3.2 Getting test data

We strapped a Trojan that we'd have written on the phone, which can hook the data that WeChat app with WeChat server interaction out and sent to the remote server processing and storage without affecting WeChat app normal circumstances.

Implement an automated data acquisition system need to use 4 main tools as fllows:

1)Trojan: The Trojan is strapped on the phone and can hook the data that WeChat app with WeChat server interaction out without affecting WeChat app normal circumstances.

2)Monkeyrunner: Control your phone clicking and automatic sliding operation.

3)Robot: Simulation of WeChat web client automatically orwarded data to the WeChat app to helping monkeyrunner to clicking.

4)Data Handler: process data that be acquired by Trojan and stored in the database.

Flow between them is:

1)The system takes WeChat public platform's key data from the database and the construct a clickable link. Then forwarded to the WeChat app by Robot.

2)Monkeyrunner clicks the link which forwarded from Robot on WeChat app to triggering the data exchange between the WeChat app and server.

3)The Trojan is responsible for hooking the data that WeChat app with WeChat server interaction out then sent to Data Handler.

4)Finally, Data Handler process the data and store in the database.

WeChat reptiles is not discussed in detail since is not the point in this article.

3.3 Experimental results and performance analysis

In general, an article's reading number will have a rapid growth phase along with the time going, and then grow very slowly can be ignored at time t0. By monitoring the test set reading number's growth trends available t0 = 7. Thus after 7 days the amount of reading will not have a significant increase in general. Therefore, the amount of seven days reading as a reference data set compare with articles those recommended use TNAPR for TNAPR further evaluation.

In the experiment, the value of i and j need to meet the conditions: i + j = 1 and i <j. For {i = 0.1, j = 0.9; i = 0.2, j = 0.8; i = 0.4, j = 0.6} 3 different groups values, come in 3 different accuracy curve. Figures 3-5 are three different comparison between the actual amount of the final reading and recommended score(Due to space limitations, the figure only shows the amount of reading and recommended score for each public platform's headline articles when March 1, 2016). The abscissa is public platform ID every article belongs to. Left ordinate is the actual amount of reading of the article, and the right ordinate is predictive scores use of TNAPR.

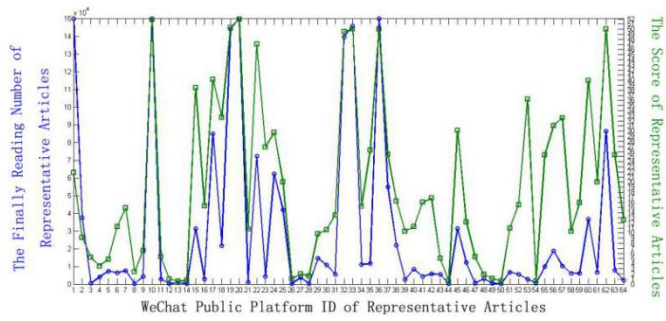

Figure 3 The comparison between the actual amount of the final reading and recommended score (i=0.1,j=0.9)

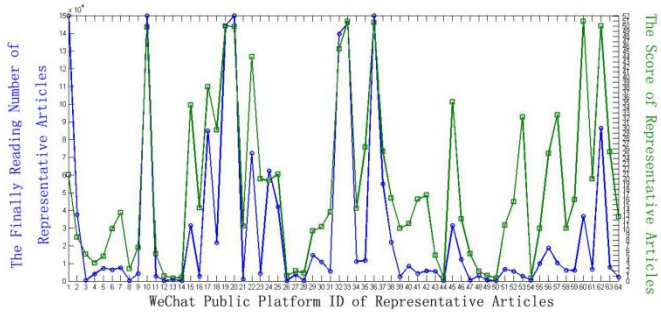

Figure 4 The comparison between the actual amount of the final reading and recommended score (i=0.2,j=0.8)

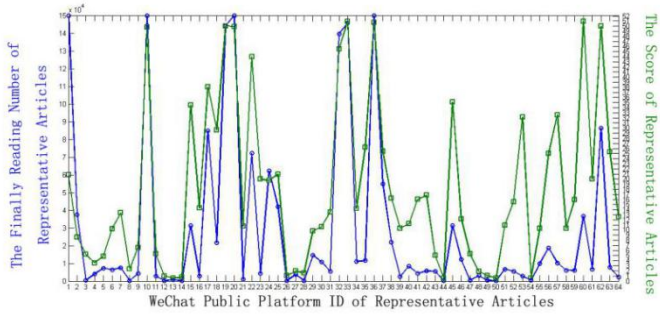

Figure 5The comparison between the actual amount of the final reading and recommended score (i=0.4,j=0.6)

Figure 3 is comparison of predicted scores and the actual reading number when{i = 0.1, j = 0.9}. It is seen from figure 3 that there are 6 in 64 articles' reading number is bigger then 10w+ which ID are 1、10、19、20、33and36. The predicted reading number could exceed 10w (predicted scores greater than 50) are 10,19,20,33,36 and 62. So the predicted errors are 1and 62. The accuracy rate behalf of the 64 articles is 96.88%. It is seen from figure 4 and 5 that the accuracy rate is 93.75% and 90.63% when {i = 0.2, j = 0.8} and {i = 0.4, j = 0.6}. These accuracy rates only represent the status of 64 representative articles. Predictions on the overall data may differ.

Continue to adjust the value of i and j with actual test data and least squares in order to get the best results. Figure 6 is the accuracy rate curve of different value of i and j. Through simulation analysis, the highest accuracy rate can reach 73.54% when{i = 0.1, j = 0.9}.

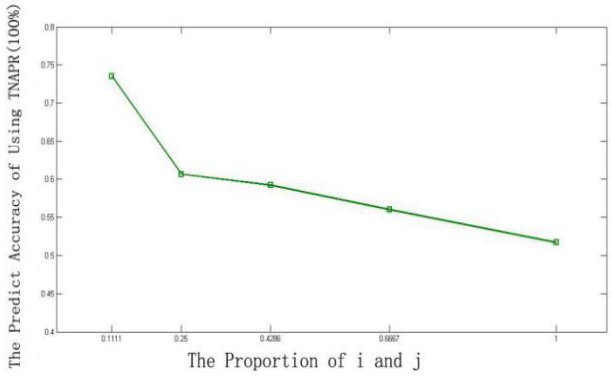

Figure 6 The accuracy curve of using TNAPR

4 Summary

This paper presents a recommendation algorithm which recommend articles that reading will reach 10w+ and this recommendation algorithm can be widely used article recommendation system. The recommendation algorithm has been proposed based on WeChat public platform's new articles, which the accuracy of the simulation and evaluation in training data set can reach 70% and is able to accurately predicting articles which reading number will reach 10w+ then recommend it.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants Nos. 61302105 and 61302163), and the Fundamental Research Funds for the Central Universities (Grants Nos. 2014MS99).

References

[1] ZHAO Z D, SHANG M S. User-based Collaborative-filtering Recommendation Algorithms on Hadoop[A]. Knowledge Discovery and Data Mining, WKDD'10 Third International Conference on IEEE[C]. 2010：478-481.

[2] WU Y, SHEN J, GU T Z, etal. Algorithm for Sparse Problem in Collaborative Filtering. Application Research of Computers, 2007;24(6):94-97.

[3] LI B Q. SEO and UEO-Based OPAC Optimization. Journal of Library Science in China, 2013; 39(206): 120-128

[4] CHEN H. Research on Recommendation Algorithm of Personalized Search Engine. Hunan: Institute of Computer and Communication of Hunan University, 2009

[5] ZHANG Q. The Public Influence and Development Stategy of WeChat Official Accounts of Party Newspapers. Jilin: Cultural Institute of Jilin University, 2015

[6] XIE W L. Service Innovation of Academic Journals' micro-message Public Numbers in the Mobile Internet era. Chinese Journal of Scientific and Technical, 2015;26(1):65-72