Research on Hybrid Recommendation Model Based on PersonRank Algorithm and TensorFlow Platform

Guangqi Wen\textsuperscript{1,a}, Chunmei Li\textsuperscript{1,b,*}

\textsuperscript{1} Department of Computer Technology and Application, Qinghai University Xining 810016, China
\textsuperscript{a}595800468@qq.com
\textsuperscript{b}li_chm0422@sina.com
\textsuperscript{*} Corresponding Author

Abstract. At present, the borrowing data of university libraries in western China generally has the problem of large sparseness and inaccurate recommendation results. However, the traditional recommendation algorithm does not solve the problem. Therefore, solving the data sparsity problem and improving the recommendation accuracy has become a very important issue. Through research, the author proposes a hybrid recommendation model that combines the PersonRank algorithm with the neural network trained by Google's artificial intelligence framework TensorFlow in the context of excessive data sparseness in western college libraries. First, using the existing borrowing records to model through the bipartite graph probability random walk method, the author obtains a list of books that the reader may be interested in and calculates the probability value of interest. After that, the author converts the probability value into a pseudo-scoring table, and uses TensorFlow to calculate by pseudo-scoring, and supplements the data training set and book classification information to obtain the recommendation result. The model effectively solves the problem of data sparsity in the recommendation system. Finally, it compares with the traditional recommendation algorithm and an improved recommendation algorithm. The accuracy of the model is evaluated by AUC index, and the accuracy of the proposed model is higher.

1. Introduction

With the advent of the era of big data, it has become increasingly difficult for people to obtain the resources they need or are interested in from massive amounts of data. The emergence of this problem has brought enormous challenges to the related work of the library, especially for readers, who cannot quickly and accurately find the collection resources they need.

The concept of "recommended system" was proposed by Robert Armstrong, a professor at CMU University, in the American Association of Artificial Intelligence (AAAI) in 1995. The more famous recommendation system applications are: Amazon and Taobao's e-commerce recommendation system, Netflix and MovieLens' movie recommendation system, Youtube's video recommendation system, Douban and Last.fm's music recommendation system, Google's news recommendation system, and Facebook and Twitter's friend recommendation system. Among them, the collaborative filtering recommendation algorithm as a representative of the traditional recommendation algorithm has been the focus of many scholars. These two problems are the data sparsity problem and the cold start problem. The collaborative filtering algorithm recommends very good results when the amount of data...
is sufficient and a certain user base is recommended. But when the data is too sparse, its accuracy will drop dramatically.

There are also many well-known traditional recommendation algorithms, such as topic-based model-based recommendations. These traditional recommendation algorithms usually use the similarity between computing readers or books to cluster. Similarly, these traditional algorithms have good recommendations when the amount of data is sufficient and the data size is small. However, with the explosive growth of book resources, these traditional recommendation algorithms gradually reveal their drawbacks. With the explosive growth of data volume, they cannot evolve in time, and the sparseness of data becomes more and more prominent, which directly leads to the failure of traditional recommendation algorithms to make accurate recommendations.

Zhou[1] et al. proposed a network-based resource allocation model in 2007, which achieved better results than the traditional collaborative filtering algorithm. Wang Qian[5] et al. proposed an improved recommendation algorithm based on the bipartite graph, introducing the ratio $\Theta$ of the sum of the project degree and the weight of the project, which improves the recommendation accuracy and diversity.

At the same time, the emergence and popularity of machine learning artificial intelligence has also brought huge development space for the recommendation work. Google's YouTube recommendation uses the TensorFlow framework to recommend video to users and has achieved great success. Some scholars have introduced the FFM model into the recommended field, and the recommendation results are generated by predicting the scores of the items. Although this solves the sparseness problem of the traditional book recommendation algorithm, the efficiency is low and the recommendation effect is not obvious.

2. The source of the data

2.1. Data collection
In the research process of this paper, the author collected the data of Qinghai University Library. The collected data table has more than 90,000 pieces of reader information table, more than 300,000 pieces of data in the borrowing record table, and a book information table and a book category table. Simple processing of each data table to remove useless data. Keep valid information such as reader name, job number (student number), borrowed books, and book categories.

2.2. Organize data
All the data tables are associated, and all the information is organized into one table. The data in the final data table is the reader barcode (work number)-name-reader level-unit-grade-professional-title.

| Table 1. data |
|---------------|
| Reader’s barcode | Reader’s name | Reader’s level | Reader’s unit | Borrowing books |
| 201100502 | Shen Hui | student | Medical school | Introduction to Organic Chemistry |
| 201113080 | Zhang Dejing | student | School of Chemical Engineering | Principle of inorganic chemistry |
| 209021404 | Bian Pengguo | student | material science | Gas turbine (1) |
| 209021404 | Bian Pengguo | student | material science | Gas turbine (2) |
The above is some of the data listed. After the statistics are completed, a total of 11137 rows of data are obtained.

2.3. Data cleaning
There are many problems in the data set after sorting, and the data needs to be cleaned.

- In the process of building a library management system, testers often test the system and generate a large amount of test data. These are often not real borrowing data and need to be manually deleted.

- The library has undergone many changes in the development process, and each change will take a different form of record. For example, the book type, the book number format corresponding to a group of books is similar to K249.07; but there is also a batch of books corresponding to the classification number is similar to the J993/997 format. The research in this paper has dealt with this kind of data uniformly and unified the format.

- The research data table format in this article uses the csv file format. The csv file format uses a comma to separate the data. In many book names, there are commas, which have a great impact on the reading of the data. Therefore, using python to write a script to traverse the data set, mark out the problematic data bar, and manually adjust the title of the book to adjust the problem.

- In the process of importing datasets into the model, the variety of reader barcode formats always leads to the unrecognizable model. Therefore, each reader is mapped and mapped to the list of 0-Personmax. Personmax is the total number of readers. After the mapping is completed, the mapped data table is trained as a model input part.

3. Mixed model

3.1. PersonRank
In the research process of this paper, the author collected the data of Qinghai University Library. The collected data table has more than 90,000 pieces of reader information table, more than 300,000 pieces of data in the borrowing record table, and a book information table and a book category table. Simple processing of each data table to remove useless data. Keep valid information such as reader name, job number (student number), borrowed books, and book categories.

3.1.1. PersonRank algorithm
The idea of the PersonRank algorithm is to calculate the importance of each node by means of material diffusion. The algorithm transforms the reader's borrowed data into a bipartite graph of the image. For a borrowed record dataset, the reader and the book are divided into two parts. If a reader has read and borrowed a book, the reader and the book are connected by a line. In this way, all the data is connected, which constitutes the reader-book. We define the nodes, the user collection is \{A, B, ..., W, ... Z\}, the book collection is \{a, b, c, ...\}, and \(R(n)\) represents the importance of the n-node.

The PersonRank algorithm is for a certain reader. Here, reader W is used as an example to walk around the established bipartite graph by random walk, and the process is iterated until the data
converges. Finally, the degree of importance to all other nodes of the reader W is obtained, from which the book object nodes and the probability values of interest are selected.

3.1.2. PersonRank algorithm calculation process
Select the reader W to recommend, and initialize the two parts:
$$R(A) = 0, R(B) = 0, \ldots, R(W) = 1, \ldots, R(Z) = 0$$
After initialization, the two parts are swam, and the starting point of the walk must be a point where the R value is not zero. After starting the walk, the probability of going to the next point is \(\alpha\), and the probability of staying at the current node is \(1-\alpha\). Let the current node be \(n\) and the node of the reader W be \(N\). Then the importance value of a node is calculated as follows:
$$R(n) = \begin{cases} \alpha \times \frac{\sum_{n' \in in(n)} R(n')}{|out(n')} | & n \neq N \\ (1-\alpha) + \alpha \times \frac{\sum_{n' \in in(n)} R(n')}{|out(n') |} & n = N \end{cases}$$

calculation process:

![Initialization diagram](image1)

![Take a walk](image2)

After a finite number of iterations, the probability values gradually stabilize and eventually reach stability. Screen out the book nodes that are more important to the reader and scale them up. Since the value of importance is a probability value, which is a value between 0 and 1, the magnitude is increased by two values that reach the range of 1-100. Calculate all readers and obtain a pseudo-recommendation list for each reader and the reader's level of interest in each book.

3.1.3. Pseudo-scoring table
In many cases, there is no reader's rating of books in the library borrowing data. Without the score sheet, we can't grasp the reader's personal preferences, and we can't make accurate predictions. So we will ask for books that are of interest to each reader. The conversion of probability values into pseudo-score tables can effectively solve this problem. For example:
The books and probabilities calculated by a reader are as follows (only five books are listed):

| Book Title                                      | Probability Value |
|------------------------------------------------|-------------------|
| "Mathematical model"                           | 0.1828            |
| "SPSS Data Analysis Tutorial"                  | 0.1342            |
| "Wake up and still love you"                   | 0.0344            |
| "MATLAB Scientific\nComputing and Visual Simulation Collection" | 0.0220            |
| "computer network"                             | 0.0205            |

The probability values are all in the range of 0-1. The current work is to convert it into a value within 0-100, which is a pseudo-score. Multiply all probability values by 100 and the result of the
calculation is a pseudo-score. Calculate the pseudo-scores of all readers and organize them into pseudo-scores.

3.2. Tensorflow

3.2.1. TensorFlow platform introduction

TensorFlow is Google Brain's second-generation open source machine learning system, and the versatility of the formal system makes it widely used in other computing fields.

The core of TensorFlow is the data flow graph, which uses a directed graph of "nodes" and "lines" to represent a large number of mathematical calculations. "Node" generally refers to some mathematical operations, sometimes also to the beginning or end of the system; "line" represents a relationship between points. As shown below:

Figure 3. TensorFlow workflow

In this paper, we use TensorFlow to predict pseudo-scoring tables and reader eigenvalues that have been collated using the PersonRank algorithm. The model used is a content-based recommendation model.

3.2.2. TensorFlow training on data sets

Through the secondary cleaning and finishing of the data, each reader's rating information for each book and the category of each book are obtained. The training data set must contain the user's preference information as well as the rating information, so we divide the data into a single set to facilitate the training of TensorFlow.

Content-based recommendations are divided into two parts. First, the data is compiled to calculate the similarity between readers and books, and simple clustering between similar objects is performed. The second is to recommend the reader through the results of similarity.

Calculate the loss value of the training model:

\[ \text{loss1} = \frac{1}{2} \sum_{p=1}^{n} \sum_{W(r, p) = 1} ((\theta^{(p)})^T \times x^W - y^{(W, p)})^2 \]

\[ n \text{ represents the number of readers; } r(W, p) \text{ represents a pseudo-scoring table; } (\theta^{(p)}) \text{ represents the preference of the p-user; } x^W \text{ represents the feature of the W-book; and } N \text{ represents the number of features.} \]

\[ \text{loss2} = \frac{1}{2} \sum_{p=1}^{n} \sum_{k=1}^{N} (\theta_k^{(p)})^2 \]

\[ \text{loss2} \text{ represents the loss of } \theta \text{ parameters.} \]
Loss2 represents a regularization term, and the role of regularization is to prevent the training data from eventually over-fitting. There are usually two ways to solve the over-fitting phenomenon. One is to minimize the number of selected variables, and the other is to use regularization. The regularization method automatically weakens unimportant feature variables and automatically "extracts" important feature variables from many feature variables, reducing the order of magnitude. The introduction of regularization is very helpful to this model, which balances empirical risk and structural risk.

\[ \text{lose} = \text{lose1} + \text{lose2} \]

The loss value loss of the model is obtained, and further calculations and optimizations are required. The algorithm is optimized, the calculated loss value is calculated again, the loss gradient with respect to the model parameters is calculated, and then the variable is updated by the calculated loss gradient to achieve a better training effect. The loss value can be used to measure the degree of error in the prediction, the loss gradient.

Next, the data set is trained by the CPU, and the training result is stored in the file after one training, so that the next direct acquisition can reduce the calculation amount and time consumption.

3.3. Feature-incremental hybrid model
At present, the accuracy of the traditional, single recommendation model is getting lower and lower, resulting in unsatisfactory final recommendation results. Hybrid models often compensate for certain deficiencies in a single model, and the results are complemented by complementary advantages between models.

In this paper, the feature-incremental hybrid model is used to calculate the former calculation result as part of the input of another algorithm. The pseudo-scoring table calculated by the PersonRank algorithm in this paper is used to calculate the TensorFlow data set, which is a standard feature-incremental hybrid model. The pseudo-scoring table calculated by the PersonRank algorithm effectively solves the sparseness problem of the data and provides the potential interest of the reader. The content-based recommendation algorithm makes up for the inaccuracy caused by the PersonRank algorithm ignoring the edge weight.

4. Experimental result

4.1. AUC indicator evaluation
AUC is a model evaluation index for the evaluation of the two-category model. The recommended result in this paper is a two-category model, which divides the book object into two categories: recommendation and non-recommendation. To calculate the AUC, the ROC curve is first constructed. The ROC curve is drawn based on the real category and prediction probability of the sample. Building the ROC curve requires predicting the true category and predicted category of the object. By calculating the AUC we can know the accuracy of the prediction. When the value of the AUC indicator is higher than 0.5, the prediction result is good. If it is lower than 0.5, the prediction result is not ideal, and the randomness is relatively strong.

4.2. Pseudo-scoring table generation and detection
By collecting the borrowing data of Qinghai University Library and calculating it, the data of Qinghai University Library obviously has the problem of excessive data sparsity, and there is no book score. The calculation of the model used in this paper is very suitable. The following is a pseudo-scoring table calculated by the PersonRank algorithm by borrowing data from a teacher. (only the first five pieces of data are taken):

| Pseudo score   | "JAVA from entry to mastery" | "MATLAB Scientific Computing and Visual Simulation Collection" | "PHP, MYSQL and JAVASCRIPT study manual" | "PYTHON scientific calculation" | "Data Mining: Practical Case Analysis" |
|----------------|--------------------------------|-------------------------------------------------------------|------------------------------------------|---------------------------------|----------------------------------------|
| 7.67           | 5.49                           | 5.33                                                        | 5.33                                     | 4.48                            |                                        |
4.3. TensorFlow training model results analysis

The pseudo-scoring table is made into a scoring matrix and is calculated as part of the model input. The results obtained are as follows (only the first five data are taken):

| Book Title                                                                 | Final rating |
|---------------------------------------------------------------------------|--------------|
| "JAVA from entry to mastery"                                              | 8.23         |
| "MATLAB Scientific Computing and Visual Simulation Collection"             | 5.11         |
| "PHP, MYSQL and JAVASCRIPT study manual"                                  | 5.08         |
| "Data Mining: Practical Case Analysis"                                    | 5.03         |
| "Algorithms in the Age of Big Data: Artificial Intelligence, and Typical Examples" | 4.63         |

The calculated book score results are all converted into probability values of 0-1, which are used to represent the user's degree of interest value. The recommended results (ten recommendations) are manually labeled, that is, the recommended object selects the recommended results and selects the book of interest. Finally, two lists are obtained, namely a 0-1 list (1 is of interest, 0 is not of interest) and a probability table. Draw the ROC curve:

![ROC curve](figure3.png)

The X-axis represents the false positive rate, and the Y-axis represents the true positive rate. By calculating the area under the ROC curve, the value of the AUC index is 0.73, the AUC index is greater than 0.5, and the model prediction result is good.

4.4. Comparative analysis

The results calculated by looking at the model presented in this article are far from explaining the problem. It is necessary to compare the results of some of the recommended models that are currently used extensively, and then to observe the recommended effects of the model in this paper.

Here, the average AUC index value of each model is obtained by performing a large number of calculation comparisons with the collaborative filtering recommendation model and the weighted bipartite graph-based recommendation model under the same data set:

The text of your paper should be formatted as follows:
Table 4. Model comparison

| Average AUC indicator value | Collaborative filtering model | Recommendation model based on weighted bipartite graph | Model studied in this paper |
|----------------------------|-------------------------------|-----------------------------------------------------|----------------------------|
|                            | 0.68                          | 0.72                                                 | 0.73                       |

The comparison chart is as follows:

![Average AUC indicator value chart](chart.png)

The collaborative filtering model is the representative of the older recommendation model. The recommendation model based on the weighted bipartite graph is the latest recommendation model. By comparison, we can see that the proposed model has better recommendation results.

5. Conclusion

Aiming at the data set with too large data sparseness and no scoring information in the recommended field, this paper proposes a hybrid recommendation model based on the combination of PersonRank algorithm and TensorFlow. The pseudo-score table is calculated by the PersonRank algorithm and then the TensorFlow platform is used to perform data set training on the content-based recommendation algorithm to obtain the recommended result. Finally, through comparison with the traditional recommendation algorithm, the hybrid recommendation model proposed in this paper has higher accuracy. However, for data sets with large data volume and comprehensive information, the proposed algorithm is less accurate, because ecause the recommended model in this paper is only applicable to data sets with large data sparsity.

References

[1] Zhou T, Ren J, Medo M, et al. Bipartite network projection and personal recommendation[J]. Physical Review E, 2007, 76(4): 046115.
[2] Zhang Jun, Li Xin. Handwritten Character Recognition under TensorFlow Platform[J]. Computer Knowledge and Technology: Academic Exchange, 2016 (6): 199-201.
[3] Zhang Shuang, Huang Peng. An Empirical Analysis of the Sparse Problem in the Book Recommendation System of University Libraries[J]. Journal of Academic Libraries, 2014 (6): 47-53.
[4] Zhou Lingyuan, Duan Longzhen. Design and Implementation of Personalized Book Recommendation System——Taking Nanchang Aviation University Library as an Example[J]. Library Theory and Practice, 2014 (12): 106-109.
[5] Wang Wei, Duan Shuangyan. An improved recommendation algorithm based on bipartite graph network structure [D]. , 2013.
[6] Li Shuqing, Xu Xia, Xu Minjia. Book Qualification Measurement Method and Personalized Book Recommendation Service Based on Reader Borrowing Two-Division Network[J]. Journal of Library Science In China, 2013 (3): 83-95.

[7] Xia Jianxun, Wu Fei, Xie Changsheng. Application of Data Filling to Alleviate Sparse Problems and Realize Personalized Recommendation[J]. Computer Engineering and Science, 2013, 35(5): 15-19.

[8] ZHANG Xinmeng, JIANG Shengyi. Personalized recommendation algorithm based on weighted bipartite graph[J]. Journal of Computer Applications, 2012, 32(03): 654-657.

[9] Wang Lianxi. A Personalized Book Recommendation System for University Libraries[J]. Modern Information, 2015, 35(12): 41-46.

[10] YANG Wu, TANG Rui, LU Ling. News recommendation method based on content-based recommendation and collaborative filtering[J]. Journal of Computer Applications, 2016, 36(2): 414-418.

[11] An Dezhi, Liu Guangming, Zhang Heng. Book recommendation model based on collaborative filtering[J]. Library and Information Service, 2011, 54(01): 35-38.

[12] Jiang Zhoufeng, Yang Jun, E Haihong. Content-based recommendation algorithm combined with social labeling[J]. Software, 2015 (2015 01): 1-5.

[13] ZHANG Yiwen, WANG Wei, CHENG Jiaxing. Improved recommendation method for bipartite random walk based on user interest[J]. Journal of Computer Applications and Software, 2015, 32(6): 76-79.

[14] Zheng Xiangyun, Chen Zhigang, Huang Rui, et al. Personalized book recommendation algorithm based on topic model[J]. Journal of Computer Applications, 2015, 35(9): 2569-2573.