Advances in Intelligent Systems Research, volume 147
International Conference on Network, Communication, Computer Engineering (NCCE 2018)

Recommender System for Books in University Library with Implicit Data

Guang Liu[1, a)], Xu Zhao[2, b)]

[1] *Key Laboratory of Modern Teaching Technology, Ministry of Education, P. R. China, Network Information Center, Shaanxi Normal University, Xi'an 710119, China.*
[2] *Library, Shaanxi Normal University, Xi'an 710119, China.*

[a)] liuguang@snnu.edu.cn,
[b)] zhaoxu@snnu.edu.cn

Abstract. Recommender system is a very important tool to help customers make choices more easily in a large variety of offered products. However, it is difficult to make directly use of the recommender system to provide suggestion for the traditional books in a library because of the shortage of the explicit feedback, like readers' rating, reviews etc. We propose a model that transfers the implicit data of readers borrow history to explicit data and apply the SVD++ algorithm in the recommender system.

Keywords: Implicit Feedback; Explicit Representation; SVD++; Model-based Collaborative Filtering; Book Recommender Systems.

INTRODUCTION

There are thousands of or millions of books in the library of a university. It is a hard work for readers to find the books that they need if they just know a little information about books. At the same time, it is a difficult thing for the library to make full use of these books in paper form. How to help readers find the books what they need? And how to make full use of the books in a library? These are great challenges for a library.

Recommender systems provide users with personalized suggestions for products or services, and some researchers have adopted the recommender systems for the library [1, 2]. One approach that is commonly used is collaborative filtering (CF) [3]. CF is a technique of predicting a user's preferences based on preferences from many users, it is only depending on the user's historical behavior data, and it has worked very well in various domains, for instance, many big successful websites such as Amazon and so on [2].

Hence CF can be used in building book recommender system. There are two main categories in CF: the neighbor-based approaches and Latent Factor Models (LFM) [4]. LFM is a model-based CF method which use the observed preferences information to create the data model. It is based the matrix factorization technique [5] to tackle the sparsity and to predict the unobserved preferences. LFM has better prediction accuracy and extensibility [2, 6]. SVD++ is a more accurate approach based LFM by adding biases and implicit feedback. It takes advantage of the explicit feedback and implicit feedback. Therefore, SVD++ algorithm is a preferred choice in the personalized recommendation service of university library.

There are two types of preference data: explicit feedback and implicit feedback. In the web-service such as Amazon, system can easily collect several explicit feedback and implicit feedback including user evaluation information, user shopping records and browsing records. However, there is little explicit data in the book management system, so the explicit feedback is not easily obtained although it can reflect user's preference. In contrast to this, implicit feedback is more easily get and has various data forms and large scale, although it doesn't reflect user's preference directly. So

implicit feedback can be used in book recommendation. Since SVD++ uses explicit feedback and implicit feedback at the same time, how to recommend only by using implicit feedback in SVD++ is a big problem.

The available library implicit data is binary (borrowed or not borrowed books), however most majority of existing model-based approaches are designing for non-binary data [2]. Therefore, explicit representation for library implicit feedback datasets is necessary to using the existing model-based CF. Meanwhile, because the model-based methods use objective functions to distinguish positive preferences from negative preferences, so positive preferences and negative preferences are both needed. So, the second problem that will exist with the existing model-based approaches is that it has only positive feedback. Have Borrowed books is considered positive feedback, but not borrowing books is not easily considered negative feedback, because there may be some reasons for not borrowing, for example, the reader didn't like this book or he didn't find it at all. Therefore, it needs the negative feedback while using the existing model-based approaches [2, 11]. Some researches which take advantages of matrix factorization set positive feedback to 1 and negative feedback to 0. Although this method gets negative feedback, filling with 0 in rating matrix bring the noise problem that affects the recommender system [6].

This paper proposed an explicit representation model, which can transform binary implicit data into non-binary explicit data at different levels while filtering the negative samples in implicit data. So, we can use both explicit feedback and another part of the implicit feedback in SVD++ algorithm at the same time while getting negative feedback automatically. The model takes the natural logarithm of the total borrowing days of reader as the reader's rating, to make an automatic grading for reader's preference. Since ln1=0, the model takes the length of borrowing time is 1 day as negative feedback and takes the length of borrowing time that over 1 day as positive feedback.

The following chapters are arranged as follows: The second section explains the main principle of SVD++ algorithm, the equation shows what kind of data is needed, and how to get the recommended list. The third part describes how the explicit representation model converts implicit feedback to explicit feedback to make the rating matrix. The fourth section represents how to tackle library datasets and how to adjust the parameters in using SVD++ algorithm. The last section is a summary.

RECOMMENDATION ALGORITHM

In this part, LFM and SVD++ will be simply introduced, more details can be inferred to Koren 's paper [5].

The Basic Matrix Factorization Model

The LFM equation is as follows:

$$\hat{R}_{ui} = \sum_{k=1}^{K} P_{uk} Q_{ik} \tag{1}$$

For each reader u, a vector P_{uk} is the relationship between the interest the user has and the implicit feature k. For each book i, a vector Q_{ik} is the relationship between the book i and the implicit feature k. The resulting dot product $P_{uk}Q_{ik}$ is approximates reader u's rating of book i, which is denoted by R_{ui}, and all R_{ui} forms the readers-books rating matrix.

To learn P_{uk} and Q_{ik}, the system minimizes the regularized squared error function on the set of historical ratings:

$$\min_{p_*,q_*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{2}$$

The constant λ controls the extent of regularization and is usually determined by cross-validation.

Adding Biases

For each book i, a vector b_i is the biases between the book i's rating and the mean rating, which is denoted by μ. For each reader u, a vector b_u is the biases between the reader u's rating and the mean rating. The biases equation is as follows:

$$b_{ui} = \mu + b_i + b_u \tag{3}$$

Biases extend equation 1 is as follows:

$$\hat{R}_{ui} = \mu + b_i + b_u + q_i^T p_u \tag{4}$$

To learn the factor vectors, the system learns by minimizing the squared error function:

$$\min_{p_*,q_*,b_*} \sum_{(u,i)\in\kappa} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2) \tag{5}$$

SVD++

For each user u, N(u) denotes the set of items for which user u showed an implicit preference. Considering the implicit feedback in users' historical behaviors, SVD++ equation is as follows:

$$\hat{R_{ui}} = \mu + b_i + b_u + q_i^T (p_u + \frac{1}{\sqrt{|N(u)|}} \sum_{j\in N(u)} y_i) \tag{6}$$

To learn the factor vectors, the system learns by minimizing the squared error function:

$$\min_{p_*,q_*,b_*} \sum_{(u,i)\in\kappa} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2 + \sum_{j\in N(u)} \|y_j\|^2) \tag{7}$$

The minimizing process by using stochastic gradient descent [7] is as follows:

$$e_{ui} = R_{ui} - \hat{R_{ui}}$$
$$b_u \leftarrow b_u + \gamma \cdot (e_{ui} - \lambda \cdot b_u)$$
$$b_i \leftarrow b_i + \gamma \cdot (e_{ui} - \lambda \cdot b_i)$$
$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u)$$
$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot (p_u + \frac{1}{\sqrt{|N(u)|}} \sum_{j\in N(u)} y_j) - \lambda \cdot q_i)$$
$$y_j \leftarrow y_j + \gamma \cdot (e_{ui} \cdot \frac{1}{\sqrt{|N(u)|}} \cdot q_i - \lambda \cdot q_i)$$

Once all factor vectors are calculated, for each reader u, the all $\hat{R}_{ui}$ can be sorted in descending order. According the sorted results, system can make a recommend list for each user u.

EXPLICIT REPRESENTATION MODEL

There are many kinds of implicit feedback, e.g. book borrowing time, book return time, the number of borrowed books, the borrowing times. Among these data, the borrowing days can be changed into the rating data of the reader for the borrowing books. So, we calculated the reader's bias preference for some books by the accumulated borrowing days.

According to the rules of the sample school's library [8], the maximum days for one reader borrowing a book is 150 days. If one reader borrows a book more than one time, the accumulated days will be larger. The number of the borrowing days is too large to calculate easily. So, we take natural logarithms for the days as the reader's rating for the books. The readers-books rating matrix is described as follows:

We reserved special indexing letters for distinguishing users from books: for user u and for book b, the n denotes the user u borrowed the book b n times. b_n refers to the date of user u the nth borrows book b. r_n refers to the date of the user u the nth returns book b. $t_n = r_n - b_n$ are the days for the user u the nth borrows the book b. And the N refers to the total times for the user u borrowing the book b. So, the total days for user u borrowing the book b is $\sum_{n=1}^{N} t_n$ .the rating for the user u borrowing the book b is:

$$\mathrm{P(u)} = \ln \sum_{n=1}^{N} t_n$$
$$t_n = \begin{cases} t_n, t_n < 150d \\ 150, t_n \geq 150d \end{cases} \quad (8)$$

The function is the rating model. The data of the 150 days comes from the sample school's library rules. If the number of borrowing days exceeds 150 days, we just keep the 150 for the calculation. If the number of borrowing days is less than 150 days, we will adopt the actual days for the calculation.

RECOMMENDATION

In this section, we introduced the data source for the test, the data cleaning, and the adjustment of parameters in SVD++ algorithm.

Datasets

The datasets come from the sample school's Aleph library system which contain the borrowing records during a four years period. We need to take all the transactions for each user and put these into a format recommender system can use. The data just contains the user id, book id, lending days, timestamp.

Then we do a job of cleaning data, like to delete some test records. And we choose 1000 users' borrowing records as the datasets who have more 100 times borrowing records. The final datasets have 146109 records, which contain 1000 users, 81817 books. We just keep this information just like user's id, book's id, borrowing days, borrowing date and return date for compute easily. To protect the user's privacy, we renumber the user's id and book's id.

We calculate each user's rating based on the proposed model. And we got a matrix with rating 0 – 7.2. The ratings indicate the reader's different preferences where high ratings denote user's strong preferences, while low ratings show user's little interest.

Then we adopt 10-fold cross-validation model to split the datasets into training set and testing set.

Adjustment of Parameters and Recommendation

The approach of adjusting the parameters is to set a range and step size, then choose one from candidate values. Due to the limited number of experiments, the selected parameters may not be the best, but considering the computation cost, this is a compromise and feasible method. Because SVD++ has multiple parameters to be set, the process of the whole parameter adjustment could be a big job. Take the process of parameter adjustment when the latent factor changes as an example, we can see the chang by using RMSE (Root Mean Square Error) as the evaluation indicator. When training the method on the datasets of the sample school library, we use the following values for the meta parameters: γ=0.007. it is beneficial to decrease step sizes by a factor of 0.9 after each iteration.

TABLE 1. Performance for Different Number of Latent Factor

factors	**20**	**50**	**100**	**150**	**200**	**500**
RMSE	0.9802	0.9794	0.9772	0.9787	0.9791	0.9823
Time/iteration	192 s	381 s	694 s	1034 s	1425 s	3328 s

The iterative process runs for around 20 iterations till convergence. The table 1 summarizes the performance over the dataset for different number of latent factors. Here there is no more benefit in increasing factor k, as adding factors covers more global information. Through the finite experiment, the value has been already captured adequately. We report running time on a server with Intel Xeon E5 Series CPU. Increasing the number of the latent factors contributes to accuracy, but also adds the running time.

When we get the proper parameters for the model by the training dataset, we use the full dataset to get a recommend list of books for each reader.

SUMMARY

This work we proposed a new rating model which can transfer the implicit data to an explicit rating matrix. Then we build a Top N recommender system for the library's book based the rating matrix using the SVD++ algorithm. The result shows that the rating model can transfer the reader's implicit feedback to an explicit feedback, at the same time, it can make full use of the advantages of the algorithm SVD++. This system is versatile and can be used for similar book recommendation environments.

ACKNOWLEDGMENTS

Supported by Interdisciplinary Incubation Project of Learning Science of Shaanxi Normal University (SYSX201504).

REFERENCES

1. Resnick P, Varian H R. Recommender systems[J]. Commun. ACM, 1997, 40(3): 56-58.
2. Volkovs M, Yu G W. Effective Latent Models for Binary Feedback in Recommender Systems[C]. Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015: 313-322.
3. Schafer J B, Dan F, Herlocker J, et al. Collaborative Filtering Recommender Systems[M]. Springer Berlin Heidelberg, 2007: 291-324.
4. Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model[M]. 2008: 426-434.
5. Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems[J]. Computer, 2009, 42(8): 30-37.
6. YIN Jian, WANG Zhi-Sheng, LI Qi, SU Wei-Jie. Personalized Recommendation Based on Large-Scale Implicit Feedback[J]. Journal of Software, 2014, 25(9): 1953-1966.
7. Funk S. "Netflix Update: Try This at Home", Dec. 2006, http://sifter.org/~simon/journal/20061211.html.
8. Books and Periodicals Lending Service. http://www.lib.snnu.edu.cn/action.do?webid=w-s-jyfw.