# Latent Factor Model for Book Recommendation System --- Taking Douban as an Example

Hanqiao Yu
266 Xinglong Section of Xifeng Road, Xi'an, Shaanxi, China
angela@cas-harbour.org

**Keywords:** recommendation system, Collaborative Filtering, latent factor model, gradient descent, classification

**ABSTRACT:** Recommendation system is a type of web intelligence technology that can perform daily information filtering for users. It has a more and more important position in the Internet era, so the filtering technology has become a focus of it. This paper introduces a technique called latent factor model which belongs to Collaborative Filtering, and it can identify hidden themes or categories, and establish the relationship between features through implicit themes or categories. The article takes book recommendation system in Douban as an example to explain the kind of technology can contribute to improve the recommendation system.

## 1. INTRODUCTION

In the recommendation system, it is difficult for information producers to make their information stand out and catch attention. The personalized recommendation system is based on customers and attributes. The problem it needs to solve is to explore the user's behavior and find the user's personalized needs, so that the long tail product can be accurately recommended to the users who need it, and help the user to find those that are interesting to them but difficult to find[1].

Douban, as the most popular book review website in contemporary China, urgently needs to improve the accuracy of recommendations and better meet the needs of users. The recommendation system of Douban is mainly item-based collaborative filtering. Since latent factor model is a hotspot in this filed, it has its own strengths and weaknesses compared to the former. Compare the recommended effect of this model with the traditional technology to determine whether it can improve the accuracy of the existing model.

## 2. Overview of Douban book recommendation system

With the popularity of the Internet and the publication of a large number of books, there are more and more ways to obtain books and information about books, obtaining accurate recommendations has become a great problem for readers. As the current prevalent book evaluation and comment sharing platform, Douban contains many books reviews and scores that are freely written by users. Due to the high quality and non-profitability of these evaluations and comments, its credibility is far greater than that of ordinary book sales websites to a large number of ordinary people. It mainly uses two ways to recommend books.

The first one is item-based collaborative filtering, which is shown as "Customer who likes the item also likes". This algorithm idea is recommending an item that is similar to the item he liked before to the target user. It mainly consists of two steps: (1) Calculating the similarity between items; (2) Generating a recommendation list for the user based on the similarity of the items and the historical behavior of the user[2].

However, it has a problem that those popular items which do not belong to the same category are easier to have connections than those actually similar items do, because a large group of people will buy them. For instance, on Amazon, many books are related to Harry Potter[3]. It affects the recommendation accuracy aiming to customers' interests and it will prevent customers from finding unpopular items.

The second method of Douban is to provide users with some labels that allow users to make choices when scoring or commenting on books, and these labels serve as a basis for classification and recommendation. However, this method is difficult to control the appropriate granularity, and more seriously, there is a big difference in people's preferences and levels of classification. Therefore, this method is difficult to provide accurate recommendations.

To improve the prediction and recommendation performance of Douban book recommendation system, it is best to introduce the latent factor model into the Douban book recommendation system.

## 3. Method and Analysis

### 3.1 Data Source

As a recommendation system, it must be a large enough data source. According to the traditional thinking, grabbing data one by one is the most direct method, but unfortunately, Douban's "anti-data digging crawler" mechanism is very mature, so it is arduous to get enough data from Douban directly. This is very unfavorable. Another problem is that there are quite a lot inactive users on Douban, and many users have a rating of 0, which further reduces the benefits of direct crawling. Therefore, obtaining data directly from Douban can only be used as a supplementary solution, and it is not suitable as a main data source.

Use the dataset from the website as the basis, but only scores from customers, not including comments, tag dates, etc.

As a result, getting this information from the search engine crawler becomes the solution to this problem, and should also have the page data. After trying, Google has a large number of pages of the Douban book page, and the cached URL format is as follows.

Http://webcache.googleusercontent.com/search?q= cache:book.douban.com/subject/2152385/

However, the user information page is relatively small and can be used as a supplement. And get the snapshot address by searching for the URL in Baidu.

So the total data source is mainly from the data package. In terms of customers who has relatively heavy weight, the book information is scanned by the search engine, and other information is directly captured from the Douban website.

### 3.2 Latent Factor Model

LFM (Latent factor model) can identify hidden themes or categories, and establish the relationship between features through implicit themes or classifications. The general solution of LFM is to us the gradient descent method to minimize the cost function, but considering the application in the field of recommendation system to deal with massive data. In this paper, it proposes a parallel method of stochastic gradient descent to improve the efficiency of recommendation. The core idea of this algorithm is to set the user-item rating matrix and solve the two low-dimensional matrix, so that the multiplication of two low-dimensional matrix can approximately represent the rating matrix.

LFM is based on matrix factorization, which characterizes both users and items with factor vectors inferred from user-item rating matrix. It is used to find the latent themes of those items.

For example, each item is associated with a vector $q_t \quad \mathbb{R}^f$, and each user is represented by a vector $p_s \quad \mathbb{R}^f$. Here f is the number of latent factors. Then matrix factorization models map both items and users to a joint latent factor space of dimensionality *f*, a recommendation occurs in high correspondence between user and item factors, and the user item interactions are modeled as inner products in that space. For example, every book has some tags like technology, education, math, or other uninterpretable factors. From the mathematical perspective, the idea of matrix factorization is to break a $m \; n$ rating matrix *R* down into a $n \quad f$ user factor matrix *P* and an $m \quad f$ item factor matrix *Q*, as shown in formula[4],

$$R_{UI} = P_U Q_I = \sum_{k=1}^{K} P_{U,k} Q_{k,I}$$

R[i][j] represents the rating of user u for item i,

P[i][j] represents the degree of interest the user has in item factor k,

Q[i][k] represents the share the element k owns in the item i, and T(Q) represents the transpose of matrix Q.

| | item 1 | item 2 | item 3 | item 4 |
|---|---|---|---|---|
| user 1 | R11 | R12 | R13 | R14 |
| user 2 | R21 | R22 | R23 | R24 |
| user 3 | R31 | R32 | R33 | R34 |

R

=

| | class 1 | class 2 | class 3 |
|---|---|---|---|
| user 1 | P11 | P12 | P13 |
| user 2 | P21 | P22 | P23 |
| user 3 | P31 | P32 | P33 |

P

×

| | item 1 | item 2 | item 3 | item 4 |
|---|---|---|---|---|
| class 1 | Q11 | Q12 | Q13 | Q14 |
| class 2 | Q21 | Q22 | Q23 | Q24 |
| class 3 | Q31 | Q32 | Q33 | Q34 |

Q

The next step to calculate the parameter values in matrix P and matrix Q. The general approach is to optimize the loss function to find the parameters. Before defining the loss function, you need to prepare the data set and explain the value of the interest.

The data set should contain all the users and items they have labeled “like”. All of these items form a complete set of items. For each user, refer to the item whose behavior is called a positive sample, and specify the interest level ( Rui) as 1. In addition, select samples randomly from the whole set of items, and select the same number of positive samples with negative samples. The specified interest level is Rui=0.

Therefore, the value range of interest is [0, 1]. After sampling, the original data set is expanded to obtain a new user-item set K= {(U, I)}, where if (U, I) is a positive sample, then Rui=1, otherwise Rui=0. To determine p and q, the loss function is as follows:

$$C = \sum_{(U,I)\in K} (R_{UI} - \hat{R}_{UI})^2 = \sum_{(U,I)\in K} (R_{UI} - \sum_{k=1}^{K} P_{U,k} Q_{k,I})^2 + \lambda \|P_U\|^2 + \lambda \|Q_I\|^2$$

$\lambda \|P_U\|^2 + \lambda \|Q_I\|^2$ It is used to avoid overfitting.

λ needs to be experimentally repeated according to the specific application scenario. The optimization of the loss function uses a stochastic gradient descent algorithm.

The fastest descent direction is determined by finding the partial derivatives of the parameters PUK and QKI; the iterative calculation continuously optimizes the parameters (the number of iterations is manually set in advance) until the parameters converge[5].

$$\frac{\partial C}{\partial P_{Uk}} = -2(R_{UI} - \sum_{k=1}^{K} P_{U,k} Q_{k,I}) Q_{kI} + 2\lambda P_{Uk}$$

$$\frac{\partial C}{\partial Q_{kI}} = -2(R_{UI} - \sum_{k=1}^{K} P_{U,k} Q_{k,I}) P_{Uk} + 2\lambda Q_{kI}$$

$$P_{Uk} = P_{Uk} + \alpha((R_{UI} - \sum_{k=1}^{K} P_{U,k} Q_{k,I}) Q_{kI} - \lambda P_{Uk})$$

$$Q_{kI} = Q_{kI} + \alpha((R_{UI} - \sum_{k=1}^{K} P_{U,k} Q_{k,I}) P_{Uk} - \lambda Q_{kI})$$

α is the learning rate, and the larger the α is, the faster the iteration drops. Like λ, it also needs to be experimentally obtained according to the actual application scenario.

### 3.3 Evaluation Standard

$$Recall = \frac{\sum_{u\in U} |R(u) \cap T(u)|}{\sum_{u\in U} |T(u)|}$$

$$Precision = \frac{\sum_{u\in U} |R(u) \cap T(u)|}{\sum_{u\in U} |R(u)|}$$

$$Coverage = \frac{\left| \bigcup_{u \in U} R(u) \right|}{|I|}$$

R(u) is a list recommended for user u, and T(u) is a reading list of user u..

## 4. Analysis and Discussion

### 4.1 Result

Experiments are conducted to verify the performance of the proposed method in modeling user interest and the metrics of our approach on user-recommended results are evaluated.

First, create a user-book matrix, then use the interaction matrix of all users with 90% of the book as the training set, and the remaining 10% as the test data set. In the experiment, we set the number of hidden factors to 10, the learning rate α is 0.02, λ is 0.01. After that, it carried out LFM and item-based collaborative filtering recommendation. Comparison of accuracy, recall and coverage.

The evaluation of TOP 10 recommendations

| Method \ Evaluation | Precision | Recall | Coverage |
|---|---|---|---|
| Item-based | 8.1% | 25% | 18% |
| LFM | 13% | 47% | 100% |

The evaluation of TOP 5 recommendation

| Method \ Evaluation | Precision | Recall | Coverage |
|---|---|---|---|
| Item-based | 10% | 15% | 11% |
| LFM | 14.30% | 27% | 92.50% |

Summary: the precision and recall of LFM means that the predictive ability of this model is better than the method used by Douban now. Higher coverage indicates a wider range of recommendation in the whole list.

### 4.2 Comparison with Item CF

*4.2.1 Advantages*

In order to make recommendation precisely, from the perspective of data, the latent factor model adopts automatic clustering based on user behavior statistics. In this system, LFM technology has the following four advantages:

(1) The classification of items is obtained by statistics on user behavior which represents the user' s view of the classification of items. Every customer has its own category of items based on the data sets. It relatively solves the problem that different customers choose book from different perspectives. For instance, a person chooses Discrete Mathematics since it is a math book, but another one thinks it is computer related. In this case, the algorithm can give better recommendation.

(2) The granularity of the classification can be set. The larger the number of classifications, the finer the granularity of the classification, and the coarser the granularity of the classification. The precision and efficiency can be controlled easily by this feature.

(3)Each classification is a different dimension and is calculated entirely from the user' s historical data.

(4) The weight of the item in each category can be determined by statistical user behavior, so each item is not hardly assigned to a certain category, which just indicates the probability that the items belong to this category.

(5) From the results, it shows that the coverage of LFM is higher than item CF. This indicates that the recommendation system has better ability to excavate the long tail of the item, and is more likely to be a relatively unpopular book for the user as a whole but in line with the reader's interest.

Therefore, the model is efficient and useful to the Douban book recommendation system.

*4.1.2 Disadvantages*

When generating a recommendation list, LFM needs to calculate the user' s interest in all items and then rank it. The time complexity reaches F(M*N*F). Therefore, with the increasing number of users and items, the complexity of collaborative filtering algorithm is increasing rapidly, lacking scalability. This process is very time consuming. The list is difficult to update in time based on user behavior. If online systems need to recommend users in real-time after considering users' comments and evaluation history, the real-time requirements cannot be met.

### 4.3 Suggestions

Since the method proposed can improve the precision of this system and it still needs more refinements to hold large quantity of data, it is better to mix with the item-based collaborative filtering which is mainly used by Douban now. First, a smaller alternative list can be converted with the neighborhood-based algorithm, and then use lfm to derive an interest matrix. Or a weighted hybridization can be used, which means both are used, but a relatively long cycle is set for LFM. Meanwhile, Item CF can be updated in real time according to user's behavior. Then set the appropriate weights for the two recommended results, and combine them for recommendation.

## 5. Conclusion

This paper takes the book recommendation system of Douban as an example, and proposes a recommendation method based on the latent factor model. The optimization method is studied. The traditional method of random gradient descent is employed. All samples are processed in each iteration. Since the number of samples in the data set to be processed is huge, the stochastic gradient descent method is used for optimization. The results of the experiments illustrate that this model improves the precision, recall and coverage of the system, which means better performance in providing potential products for readers. There is an obvious defect of the algorithm above. The system cannot provide personalized recommendations to new customers or customers who have few records on Douban. This problem is called cold boost which greatly reduces the efficiency. That is because the user-items matrix will be too sparse. That is one of the topics that will be studied in the future to develop the book recommendation system[6].

## References

[1] Resnick P, Varian H R. Recommender systems[J]. *Communications of the ACM*, vol 40(3) , pp.56-58, 1997.

[2] Teng Xiao, Hong Shen. Neural variational matrix factorization for collaborative filtering in recommendation systems[J]. *Applied Intelligence*, vol 49(10), 2019.

[3]Greg Linden, July, 2009. [Online]. Available: http://glinden.blogspot.com/2006/03/early-amazon-similarities.html [Accessed Sept. 11, 2019].

[4] Y. F. Zhao. Latent Factor Model for Traffic Signal Control. In Proc. IEEE International Conference on Service Operations and Logistics, and Informatics, 2014, pp. 6.

[5] Liang Xiang, *Recommended system practice Beijing* CA: Beijing University Telecommunications Press. 2012.

[6] Nozomi Nori, Danushka Bollegala and Mitsuru Ishizuka. In Proc.of International AAAI Conference on Weblogs and Social Media (ICWSM 2011) 5th. 2011, pp.241.