



Improvements on Recommender System Based on Mathematical Principles

Fu Chen, Junkang Zou, Lingfeng Zhou, Zekai Xu, Zhenyu Wu

School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan, China

Email: 2302018631@qq.com

How to cite this paper: Chen, F., Zou, J.K., Zhou, L.F., Xu, Z.K. and Wu, Z.Y. (2023) Improvements on Recommender System Based on Mathematical Principles. *Open Access Library Journal*, **10**: e10281. <https://doi.org/10.4236/oalib.1110281>

Received: May 21, 2023

Accepted: July 8, 2023

Published: July 11, 2023

Copyright © 2023 by author(s) and Open Access Library Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The main goal of this article is to give optimization methods of the algorithms of the Recommender System in calculation acceleration and accuracy based on mathematical theory. We first introduce the Collaborative Filtering Algorithm and the similarity function used in this algorithm. Both the weakness and the strength of two different mathematical distance used to describe the similarity will be illustrated detailedly in this article. And both nonparametric and parametric methods will be applied to improve it. After that, we introduce BM25 Algorithm in search engines and accommodate it to Recommender System. In this article, we will give the result of performing quantile estimation in Collaborative Filtering Algorithm on the MovieLens Datasets. It is shown that it not only skips the classification process and thus accelerates calculation but also gives accurate recommendation results.

Subject Areas

Machine Learning

Keywords

Recommender System, Collaborative Filtering Algorithm, Quantile Estimation, BM25 Algorithm

1. Introduction

Nowadays, with the popularity of platforms that provide e-commerce service and short video, the need of accurate and fast Recommender System is surging. Given this background, the research of the Recommender System has great significance. Original research of this field hasn't applied optimizations based on mathematical theory like quantile estimation in Recommender System, which can skip the classification process or at least reduce the time spend on the classi-

fication process.

Generally, Recommender System is used to sift out desirable messages from a large amount of data based on some algorithms. The algorithms like algorithm applied for ranking, the algorithm applied for filtering are the core of a recommender system. If we ignore the algorithms' mathematical principles, the speed and the accuracy can be unpromising.

In order to group users that have similar preference to improve efficiency, we need to quantify the degree of similarity between them according to mathematical distance and similarity function. After several determinant aspects are chosen, user's preference can be simplified into a vector. Based on the vector, the comparison of the variety of algorithms can be conducted. The idea above leads to this short article: algorithms based on Mathematical statistics not only help to accurately filter out large amounts of useless data but also decrease the time used to calculate in the ranking process.

2. Implementation of Recommender System

In this chapter, we are going to introduce the most basic Recommender System Model.

The Recommender System generally consists of two modules: Offline Algorithm Module and Real-Time Algorithm Module.

The Offline Algorithm Module trains several models including ranking model based on historical data and generates offline preference result. The Offline Module typically updates the data everyday. Since it has 24 hours to compute the preference, the concrete algorithms in this module can be complicated.

The Real-Time Algorithm Module collects latest data and then modifies it with the feedback from the Offline Algorithm Module.

Figure 1 below is an example of the implementation of Recommender System in video application.

In the rest of this article, we will focus on the Real-time Algorithm Module, introducing its basic algorithm and giving potentially feasible optimization methods.

3. Real-Time Algorithm and Improvements Based on Mathematical Principles

3.1. Collaborative Filtering Algorithm

Collaborative Filtering Algorithm is the most widely used algorithm in Real-Time Algorithm Module. It first divides users into certain groups by their historical preference similarity and then gives latest recommendation based on other group members' feedback.

3.1.1. Implementation Process

Firstly, we determine several aspects that are enough to describe users' preference by assumption. Then we collect users' data of predetermined aspects to form a vector for each user.

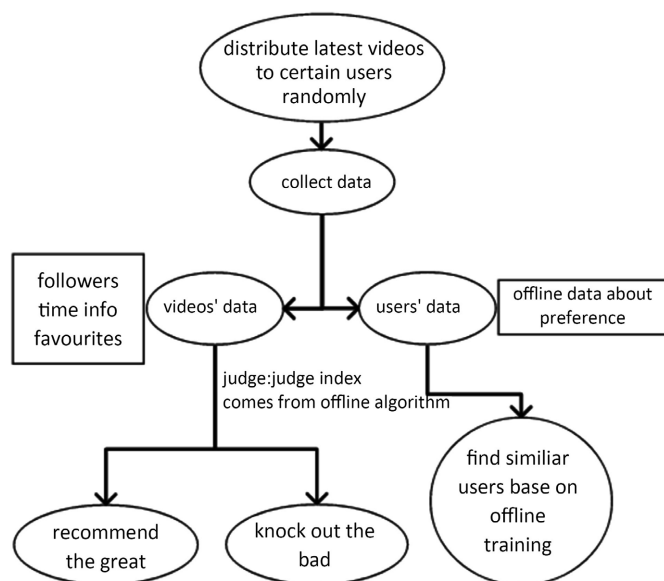


Figure 1. Example of the implementation of Recommender System in video application.

Secondly, we choose an appropriate mathematical distance to measure the similarity of users' preference. Below are two commonly used mathematical distances from [1]:

1) **Euclidean Distance**

$$d_E(A, B) = \sqrt{(a_1 - b_1)^2 + \dots + (a_n - b_n)^2}$$

where $A = (a_1, \dots, a_n)$, $B = (b_1, \dots, b_n)$

2) **Mahalanobis Distance**

$$d_M(A, B) = \sqrt{(A - B)^T S^{-1} (A - B)}$$

where S^{-1} is the sample covariance matrix.

Then we use the similarity function to measure the similarity of different users:

$$sim(A, B) = \frac{1}{1 + d(A, B)}$$

Lastly, we give preference similarity based on the similarity distance between users. The larger value the sim() function returns, the more similar the users' preference.

3.1.2. Weakness and Strength of Two Mathematical Distance

The Euclidean Distance is easier to compute, since it only involves the operation of subtraction, addition, square and square root.

However, since Euclidean Distance equates the importance of all components, the less significant component might deviate the measured similarity from the true one. To fix this incommensurate value problem, we need to look no further than the standardization process.

Moreover, the result might be influenced by the component that has dominant absolute value. For example, 10 mm height gap and 10 cm height gap are totally different in percentage change of height but they are viewed as the same in Euclidean distance.

The Mahalanobis Distance uses the sample covariance matrix to minimize the correlation between different components and assigns weight to different components to indicate its importance.

However, since the inverse of sample covariance matrix is hard to calculate and may not exist, we shall be careful when using Mahalanobis Distance.

3.2. Optimization of Filtering Algorithm

Though Collaborative Filtering Algorithm can reflect users' preference comprehensively, its processing speed is always unsatisfactory, which inspires us to optimize it with mathematical principles.

The first problem is that we need to classify the outcome of relevance according to the collaborative filtering algorithm. To reduce the time of classification and to accelerate the calculation, we need to find a possible and rational standard. The two main problems are listed below.

The second problem is that individual with different magnitude level will result in bias towards high similarity value. To obviate this bias, the Collaborative Filtering Algorithm should choose the Mahalanobis distance to guarantee the accuracy, and choose Euclidean distance to accelerate calculation otherwise.

In this section, we are going to introduce two mathematical statistics' applications in optimizing the algorithm.

3.2.1. Filtering Algorithm with Multivariate Normal Prior Distribution

With Multivariate Normal Prior Distribution, the sample (long-term historical data) also obeys multivariate normal distribution. And since multivariate normal distribution can be determined by its first two order moments, all the thing that the algorithm has to do is to estimate parameters.

The PDF of estimated sample distribution is given below:

$$f_X(x_1, \dots, x_m) = \frac{1}{(2\pi)^{\frac{m}{2}} |S|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T S^{-1}(x-\mu)}$$

Given the predetermined determinant index between 0 and 1 and the estimated parameters, we compare the determinant index with the value of PMF of the multivariate normal distribution and assign corresponding weight to components to indicate its importance.

3.2.2. Filtering Algorithm with Quantile Estimation

The Multivariate Normal Prior Distribution is too strict for most cases. A more widely used method is not to have a prior assumption and use the Quantile Estimation. To make formulas simple, we only consider unidimensional situation.

We first give some theoretical foundations about Quantile Estimation:

1) **p-Quantile Point** ξ_p

$$\text{Prob}(X < \xi_p) \leq p$$

$$\text{Prob}(X \geq \xi_p) \leq 1 - p$$

2) **Order Statistics of Sample**

If we have sample X_1, \dots, X_n that is independent identically distributed (i.i.d.) and we sort it into increasing order and get Order Statistics $X_{(1)} \leq \dots \leq X_{(n)}$.

3) **Test Statistics**

Naturally, $X_{[np]}$ is an estimator of ξ_p .

To meet with certain significance level needs, we could construct a confidence interval using the following result:

$$P(X_{(r)} \leq \xi_p < X_{(s)}) = \sum_{i=r}^{s-1} \binom{n}{i} p^i (1-p)^{n-i}$$

Readers can find more detailed introduction in [2].

Obviously, this method doesn't require any distribution assumptions. However, the exactness could be undesirable compared with method with prior assumption.

Besides, the Quantile Estimation could also be used in the Offline Algorithm Module to modify and verify results of other models.

3.2.3. Performance of Quantile Estimation in Filtering Collaborative Algorithm

In this section, we will perform Quantile Estimation in Filtering Collaborative Algorithm on the MovieLens Datasets. The dataset is detailedly illustrated in [3].

We choose "toy story" as our "origin" and randomly choose 1000 movies in the dataset as our object, and we sample 100 movies randomly and calculate its quantile estimation of similarity using the Euclidean Distance, since its relevance is given in the dataset, as the population quantile. Set the quantile 0.95, the recommendations out of 1000 movies are shown in **Figure 2** below.

```
[1] "Jumanji (1995)"
[1] "Babe (1995)"
[1] "Drop Zone (1994)"
[1] "Client, The (1994)"
[1] "Paris, France (1993)"
[1] "Chasers (1994)"
[1] "Cronos (1993)"
[1] "Naked in New York (1994)"
[1] "Brady Bunch Movie, The (1995)"
[1] "Eddie (1996)"
[1] "Country Life (1994)"
[1] "Horseman on the Roof, The (Hussard sur le toit, Le) (1995)"
[1] "Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)"
[1] "Golden Earrings (1947)"
[1] "Foreign Correspondent (1940)"
[1] "Affair to Remember, An (1957)"
[1] "To Catch a Thief (1955)"
[1] "Band Wagon, The (1953)"
[1] "Gigi (1958)"
[1] "Reluctant Debutante, The (1958)"
[1] "Adventures of Robin Hood, The (1938)"
[1] "Laura (1944)"
[1] "Ghost and Mrs. Muir, The (1947)"
[1] "Glimmer Man, The (1996)"
> number
[1] 24
```

Figure 2. Recommendations by setting quantile 0.95.

Now we choose 6000 movies as our object, and we sample 1000 movies to estimate the quantile. Set the quantile 0.99, the recommendations out of 1000 movies are shown in **Figure 3** below.

Given the results, clearly this method has two advantages. One is that the number of recommendations is controlled by the choice of quantile, and it's especially important when the dataset is large. Another is that the results are mostly animations and comedies, which are reasonable recommendations. But with the Quantile Estimation method, the movie "toy story2" has similarity value of 0.1229625, which is too low considering its relation with topic "toy story". In order to fix this problem, we will introduce the concept of corpus and an algorithm used in the search engine and try to accommodate it to our problem in the next chapter.

In order to give an intuitive result of the use of corpus, we use tags given by the dataset to perform a similar quantile estimation. For each movie, the dataset provide 1128 tags to describe it. We use the Euclidean Distance to describe the similarity and set quantile 0.99, with confidence level of 80%, we have the confidence interval of 0.99 quantile: [0.8925, 0.964]. We choose 0.964 the estimation of 0.99 quantile, and the corresponding recommendations are shown in **Figure 4** below.

The recommended tags describe "toy story" well.

```
[1] "Babe (1995)"
[1] "Client, The (1994)"
[1] "Paris, France (1993)"
[1] "Naked in New York (1994)"
[1] "To Catch a Thief (1955)"
[1] "Return of Jafar, The (1994)"
[1] "Clan of the Cave Bear, The (1986)"
[1] "Boy Who Could Fly, The (1986)"
[1] "Rollercoaster (1977)"
[1] "Golden Child, The (1986)"
[1] "Don't Look in the Basement! (1973)"
[1] "Mr. Mom (1983)"
[1] "King in New York, A (1957)"
[1] "Bird on a Wire (1990)"
[1] "Duets (2000)"
[1] "Howling III: The Marsupials (1987)"
[1] "Zorro, the Gay Blade (1981)"
[1] "Man Who Wasn't There, The (2001)"
[1] "Hot Shots! (1991)"
> number
[1] 19
```

Figure 3. Recommendations by setting quantile 0.99.

```
[1] "animation"
[1] "computer animation"
[1] "kids"
[1] "kids and family"
[1] "pixar"
[1] "pixar animation"
[1] "toys"
```

Figure 4. Recommendations by setting quantile 0.99 with confidence level of 80%.

3.3. BM25 Algorithm

In this section, we are going to introduce BM25 Algorithm used in search engines and try to apply it in the Recommender Algorithm System.

3.3.1. Corpus and Sentence Division

We first introduce the Corpus and Sentence Division, foundation of BM25 Algorithm.

Corpus means the collection of text and Sentence Division means to divide sentence according to certain rules. Corpus and Sentence Division is important for search engines. If the sentence is divided into too small parts, the number of individuals with the same meaning will be larger, and personalized degree may not be guaranteed. A very commonly used standard is to categorize by synonym, antonym and idiom.

TF-IDF (Term Frequency-Inverse Document Frequency) is often used to extract keywords from articles. It can be defined as the calculation of how relevant a word in a series or corpus is to a text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the corpus. TF represents the frequency of a Term in the Document, and the higher the frequency, the more important it is. DF represents the total number of Documents that contain this word. The larger the DF, the more common the word, and the less important it is. The smaller the DF, the more important it is. And IDF is a function of DF, so that the larger the IDF, the more important the word. To find more detailed information, readers could refer to [4]

If the frequency relationship between words in the article is that “Blink” > “practice” > “conclusion” then we can say that “Blink” is the most important keyword for this article. However, if we find that the three words “Blink”, “practice” and “conclusion” have the same frequency, it does not mean that they are equally important as keywords. It is because the total number of blogs containing these keywords is “Blink” < “practice” < “conclusion”, indicating that “Blink” is not that common. But when it appears, it is more important than “practice” and “conclusion” for this article.

3.3.2. BM25 Algorithm

Based on the information given in Corpus and Sentence Division section, we are going to introduce the BM25 Algorithm. We first give its definition:

$$score(D, Q) = \sum_{i=1}^n IDF(q_i) \frac{(k_1 + 1) f(q_i, D)}{f(q_i, D) + k_1 \left(1 - b + b \frac{|D|}{avgdl} \right)}$$

$$IDF(q_i) = \ln \left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1 \right)$$

q_i denotes a certain word or phrase;

$f(q_i, D)$ denotes the time that label q_i appears in file D ;

$|D|$ denotes the length of document D in the text collection from which documents are drawn;

k_1 and b are free parameters based on the results of Offline Algorithm Module;

$Avgdl$ is the average length;

Q and D stand for query and document respectively.

More details about the definition are in [5].

$Score(Q, D)$ describes the matching degree of D and Q based on words and phrases. Though the BM Algorithm is used in search engines, it has reference significance in label recommendation in Recommender System. For example, IDF can be used to calculate the weight of label, the number of synonyms in the video when giving videos recommendation.

3.3.3. Optimization Based on BM25 Algorithm

In this section, we are going to give a potentially feasible optimization method based on BM Algorithm in Real-time Algorithm Module.

Firstly, we shall divide all labels into appropriately small labels and all the synonymous labels should be classified into one group.

Secondly, if we treat all the labels from one object a query in the BM25 Algorithm and all the labels in all the objects a document, and naturally we could apply the BM25 Algorithm in Recommender System.

But as mentioned before, we must pay attention to the similarity and importance of certain labels, whose information generally lies in the IDF. Typically, video or other objects' tags don't repeat as much as searching for documents. So the BM25's function is enough to satisfy our needs, if given some adjustment. So it spires us to devise the score formula based on specific needs. We couldn't give the detailed formula, but only illustrate the potential of this algorithm in Real-Time Algorithm Module.

As for the judgement of the score value, we shall look back to the Quantile Estimation in the previous chapter. If the value is larger than value of quantile estimation whose standard is predetermined, then the object should be recommended.

4. Conclusions

In this article, we analyze the Recommender System in detail, and its related mathematical principles in this article.

In the first part of this article, we first introduce the two modules the common Recommender System consists of and how these two modules interact with each other.

Then we dig deep in the Real-Time Algorithm Module. We begin with Filtering Algorithm and its two different mathematical distances, illustrating both its strength and weakness. And we give two possible methods to improve the algorithm based on mathematical principles.

In the last part, we try to take a page from search engines' book and apply

BM25 Algorithm to Recommender System. We have listed basic concepts involved in search engines' algorithm and given the score function used in BM25 Algorithm. Based on this, we come up with the several possible modification of this function to meet needs of Recommender System.

Acknowledgements

Keith Briggs and Fabian Yang introduce the detailed technique and algorithm of quantile estimation. Its practical significance is revealed in our improvements on recommender system.

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] Gao, H.X. (2005) Applied Multivariate Statistical Analysis. 3rd Edition, Peking University Publisher, Peking.
- [2] Briggs, K. and Ying, F.B. (2017) How to Estimate Quantile Easily and Reliably. <https://www.maths.ox.ac.uk/system/files/attachments/Yingestimatequantiles.pdf>
- [3] Harpe, F.M. and Konstan, J.A. (2015) The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, **5**, 1-19. <https://doi.org/10.1145/2827872>
- [4] Connelly, S. (2023) Practical BM25. Elastic search B.V. <https://www.elastic.co/cn/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables>
- [5] Wikipedia (2023) Okapi BM25. In: The Ranking Function. https://en.wikipedia.org/wiki/Okapi_BM25