Hybrid-Based Academic Recommender System

Author: 顾健喆 5140309153
Teammates: 韩雨桐，王楠

Abstract

Most recommender systems use Collaborative Filtering or Content-based methods to predict new items of interest for a user. However, both methods have their own disadvantages in many situations. Thus, incorporating the two methods can overcome these shortcomings. Besides, Content-based methods usually adopt LSI to extract the features of papers. But CNN is increasingly popular in feature extracting and is proved to be effective. So, we adopted CNN instead of LSI to handle the task. In addition, generally recommender tasks have enough users to rate, while Acemap has scarce users. Therefore, we mapped the citation web to Collaborative Filtering Ratings Matrix. In short, our approach used both LSI and CNN as content-based predictor to enhance existing user data, and then provides personalized suggestions through collaborative filtering which has been mapped from citation web to ratings matrix. We present experimental results that show how this approach performs well on publications from Acemap.

Related Work

Recommender systems help overcome information overload by providing personalized suggestions based on a history of a user’ s likes and dislikes. Many on-line stores provide recommending services e.g. Amazon, CDNOW, BarnesAndNoble, IMDb, etc. There are two prevalent approaches to building recommender systems — Collaborative Filtering (CF) and Content-based (CB) recommending. CF systems work by collecting user feedback in the form of ratings for items in a given domain and exploit similarities and differences among profiles of several users in determining how to recommend an item. On the other hand, content-based methods provide recommendations by comparing representations of content contained in an item to representations of content that interests the user.

Content-based methods can uniquely characterize each user, but CF still has some key advantages over them (Her- locker *et al.* 1999). Firstly, CF can perform in domains where there is not much content associated with items, or where the content is difficult for a computer to analyze — ideas, opinions etc. Secondly a CF system has the ability to provide serendipitous recommendations, i.e. it can recommend items that are relevant to the user, but do not contain content from the user’ s profile. Because of these reasons, CF systems have been used fairly successfully to build recommender systems in various domains (Goldberg *et al.* 1992;

Dataset Description

We demonstrate the working of our hybrid approach on the dataset from Acemap. We take the use of data in two forms, one is the citations and the other kind of data is abstracts of publications. We get the paper-paper citations from mag-new dataset from Acemap server, crawled by the students from crawling group. Most of the citations are imported from Microsoft Scholar dataset, which was a free-to-download dataset, and the rest are crawled from IEEE dataset, ACM dataset, google scholar and so on. Since the dataset is too large which contains billions of papers, we decide to use only the papers from computer science field to do the experiments, which we are more familiar with, and the number of papers in computer science is also enough for evaluation. The citation dataset contains 35 subfields, and remains 33 subfields after being filtered 2 subfields that has low confidence in belonging to computer science field, which are 0.4 and 0.6 respectively.

The content information, the abstracts, of publications are also got from Acemap dataset. The source of these abstracts are similar to the source of the citations, but it is harder to crawl the abstract from datasets such as IEEE dataset and ACM dataset than crawling citation information. So, there are only about 20% publications has an abstract in our experiments, some of which are even incorrect, such as including messy codes or misplacement In view of the huge number of the publications, however, even 20% of all dataset is enough for our experiments, but it’ s far less enough to build a favorable website.

System Description

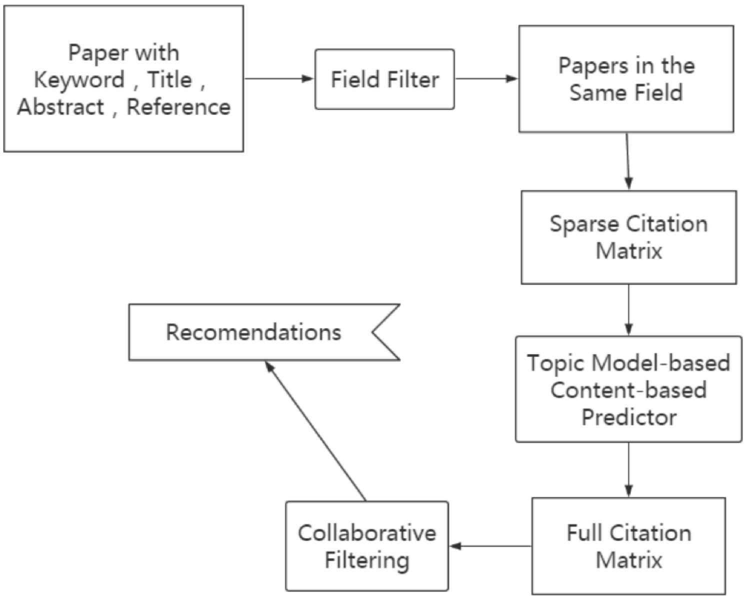


Figure 1. System Chart of the Hybrid-Based Academic Recommender System

The general overview of our system is shown in Figure 1. The web crawler uses the URLs provided in the academic datasets to download abstracts from datasets, such as google scholar, IEEE or ACM. After appropriate preprocessing, the downloaded content is stored in Acemap server. The Acemap dataset also provides the paper-paper citation matrix, which is a matrix of a paper in computer science citing another paper in or maybe not in computer science.

Since Acemap has few users currently, it is impossible for us to adopt real user-based recommending method. Thus, we will refer to each row of this matrix as a user-ratings vector. The paper citing other papers is called user, and the paper being cited is called the item. If two papers have the citing relationship, the rating between them is 5, else if they are similar in context, they will get a score from 0 to 5. The user-ratings matrix is very sparse, since most items have not been rated by most users. The content-based predictor is trained on each user-ratings vector and a pseudo user-ratings vector is created. A pseudo user-ratings vector contains the user’ s actual ratings and content-based predictions for the unrated items. All pseudo user-ratings vectors put together form the pseudo ratings matrix, which is a full matrix. Now given an active user’ s ratings, predictions are made for a new item using Collaborative Filtering on the full pseudo ratings matrix.

The Content-Based Recommender has many forms. Here we adopt a kind of topic model(Latent Semantic Indexing) and CNN(Convolutional Neural Network) to handle the task.

The following sections describe our implementation of the content-based predictor and the pure CF component; followed by the details of our hybrid approach.

Collaborative Filtering Part

We implemented a pure collaborative filtering component that uses a neighborhood-based algorithm. In neighborhood-based algorithms, a subset of users are chosen based on their similarity to the active user, and a weighted combination of their ratings is used to produce predictions for the active user. The algorithm we use can be summarized in the following steps:

1. Weight all users with respect to similarity with the active user.
   Similarity between users is measured as the Pearson correlation between their ratings vectors.
2. Select *n* users that have the highest similarity with the active user.
   These users form the neighborhood.
3. Compute a prediction from a weighted combination of the selected neighbors’ ratings.

In step 1, similarity between two users is computed using the Pearson correlation coefficient, defined below:

$$P_{a,u} = \frac{\sum_{i=1}^{m} (r_{a,i} - \overline{r}_a) \times (r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i=1}^{m} (r_{a,i} - \overline{r}_a)^2 \times \sum_{i=1}^{m} (r_{u,i} - \overline{r}_u)^2}}$$

where $r_{a,i}$ is the rating given to item *i* by user *a*; is the mean rating given by user *a*; and *m* is the total number of items.

In step 3, predictions are computed as the weighted average of deviations from the neighbor’ s mean:

$$p_{a,i} = \overline{r}_a + \frac{\sum_{u=1}^{n} (r_{u,i} - \overline{r}_u) \times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}}$$

where $p_{a,i}$ is the prediction for the active user *a* for item *i*; $P_{a,u}$ is the similarity between users *a* and *u*; and *n* is the number of users in the neighborhood. For our experiments we used a neighborhood size of 30, based on the recommendation of (Herlocker *et al.* 1999).

It is common for the active user to have highly correlated neighbors that are based on very few co-rated (overlapping) items. These neighbors based on a small number of overlapping items tend to be bad predictors. To devalue the correlations based on few co-rated items, we multiply the correlation by a *Significance Weighting* factor (Herlocker *et al.* 1999). If two users have less than 50 co-rated items we multiply their correlation by a factor $sg_{a,u} = g/50$, where n is the number of co-rated items. If the number of overlapping items is greater than 50, then we leave the correlation unchanged.

Content-Based Predictor Part

Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a method for discovering hidden concepts in document data. Each document and term (word) is then expressed as a vector with elements corresponding to these concepts. Each element in a vector gives the degree of participation of the document or term in the corresponding concept. The goal is not to describe the concepts verbally, but to be able to represent the documents and terms in a unified way for exposing document-document, document-term, and term-term similarities or semantic relationship which are otherwise hidden.

Convolutional Neural Network for Text

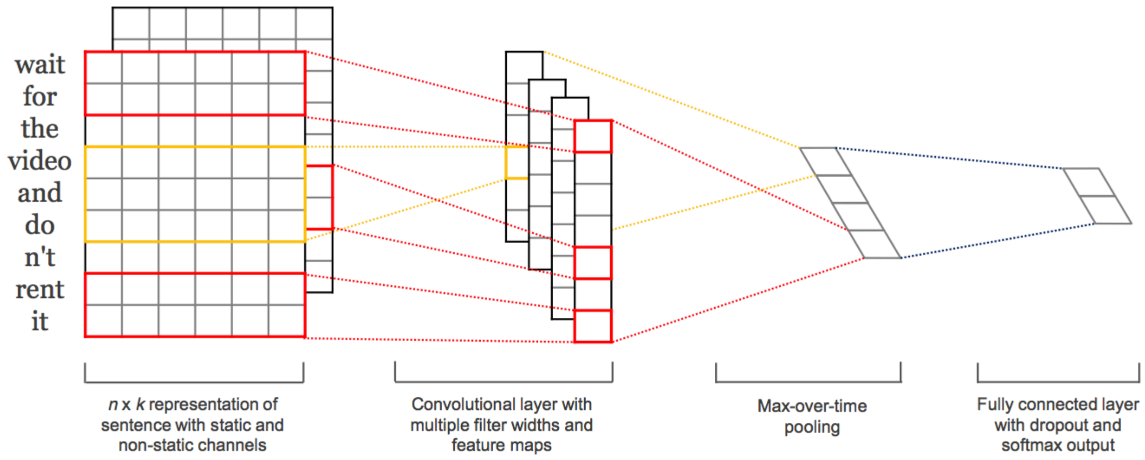

wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 2. Traditional Convolutional Neural Network for Text Classification

The model architecture, shown in figure 2, is the CNN architecture of Collobert et al. (2011).

Let $x_i \in R^k$ be the k-dimensional word vector corresponding to the i-th word in the sentence.

A sentence of length n (padded where necessary) is represented as:

$$x_{1:n}=x_1 \oplus x_2 \oplus \ldots \oplus x_n, \tag{1}$$

where $\oplus$ is the concatenation operator. In general, let $x_{i:i+j}$ refer to the concatenation of words x_i , x_{i+1} , . . . , x_{i+j} . A convolution operation involves a *filter* $w \in R^{hk}$, which is applied to a window of h words to produce a new feature. For example, a feature c_i is generated from a window of words $x_{i:i+h-1}$ by

$$c_i = f(w \cdot x_{i:i+h-1} + b) \tag{2}$$

Here $b \in R$ is a bias term and f is a non-linear function such as the hyperbolic tangent. This filter is applied to each possible window of words in the sentence $\{x_{1:h}, x_{2:h+1}, \ldots, x_{n-h+1:n}\}$ to produce a *feature map*

$$c = [c_1,c_2,\ldots,c_{n-h+1}], \tag{3}$$

with $c \in R^{n-h+1}$. They then apply a max-overtime pooling operation (Collobert et al., 2011) over the feature map and take the maximum value $c\hat{} = max\{c\}$ as the feature corresponding to this particular filter. The idea is to capture the most important feature—one with the highest value—for each feature map. This pooling scheme naturally deals with variable sentence lengths.

They have described the process by which one feature is extracted from one filter. The model uses multiple filters (with varying window sizes) to obtain multiple features. These features form the penultimate layer and are passed to a fully connected softmax layer whose output is the probability distribution over labels.

Since the task is to predict whether the altitude a comment of a movie is positive or negative. Thus, there are two labels in their work, “positive” and “negative”.

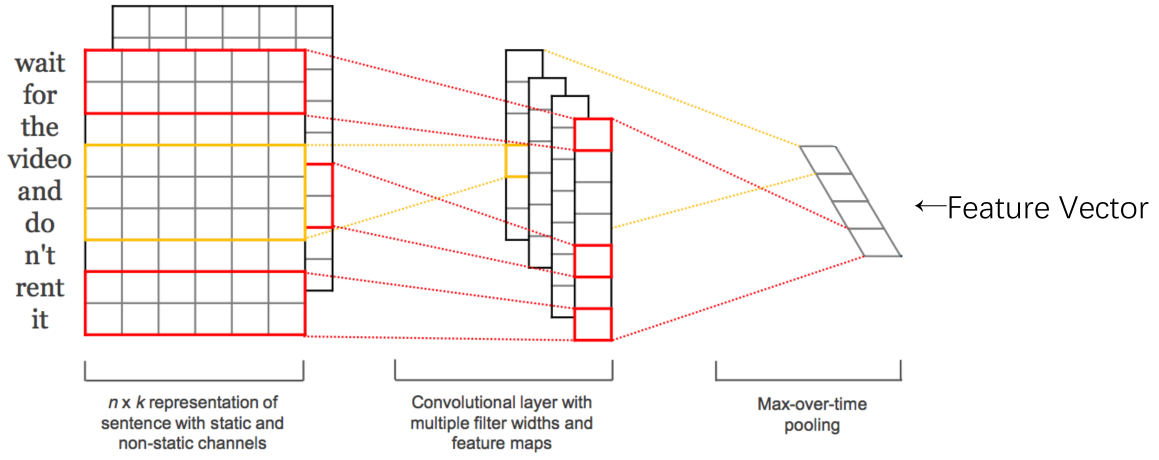

wait
for
the
video
and
do
n't
rent
it
←Feature Vector
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling

Figure 3. Convolutional Neural Network for Text Feature Extracting

Our task is slightly different from Colloberts, we should determine the similarity between papers and recommend most similar papers instead of classify papers, but it is still a meaningful CNN model which can be referenced.

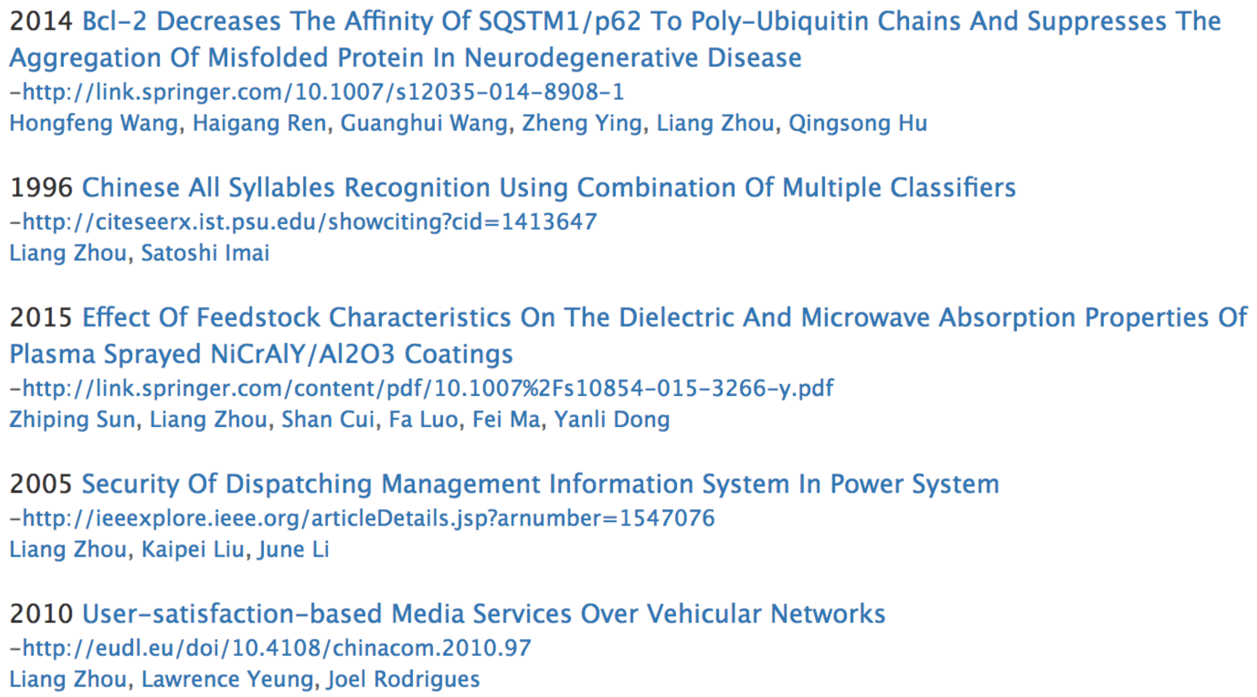
We assume that if a neural network has been trained well and has the ability to predict which field a paper belongs to, the network has the ability to recognize most features of a single paper. The feature contains the information of the topic, the research method and attitudes of a paper, so if the features between two papers are similar, or the distance between the two feature vectors are short, it also means that the two papers are similar. Intuitive, the similar paper is what users want. Thus, we use the same method to train, except that we replace the labels descripting attitude with the labels descripting academic fields and do the training.

After training the network, we change the structure of the testing network. We abandoned the last layer, label layer, and use the output of the previous layers as the feature vector of a paper, since we think this vector contains most topic and knowledge information of a paper, which can be used to calculate the similarity.

Result

We apply the Hybrid-Based recommender algorithm onto the constructing of Acemap. We calculate the similarities between all papers in computer science field, which include 33 subfields, each includes 3 million papers on average, and then select the top-5 similar papers of each recommended paper and save the results on the database. Finally, we displace the original recommending result in the "Similar" section on "Paper Page" with the new results calculated by our algorithm.

Here we select the original results and current results of one publication "Distributed Scheduling Scheme For Video Streaming Over Multi-channel Multi-radio Multi-hop Wireless Networks" (2010 Liang Zhou, Xinbing Wang) for comparing and discussion.



2014 Bcl-2 Decreases The Affinity Of SQSTM1/p62 To Poly-Ubiquitin Chains And Suppresses The Aggregation Of Misfolded Protein In Neurodegenerative Disease
-http://link.springer.com/10.1007/s12035-014-8908-1
Hongfeng Wang, Haigang Ren, Guanghui Wang, Zheng Ying, Liang Zhou, Qingsong Hu

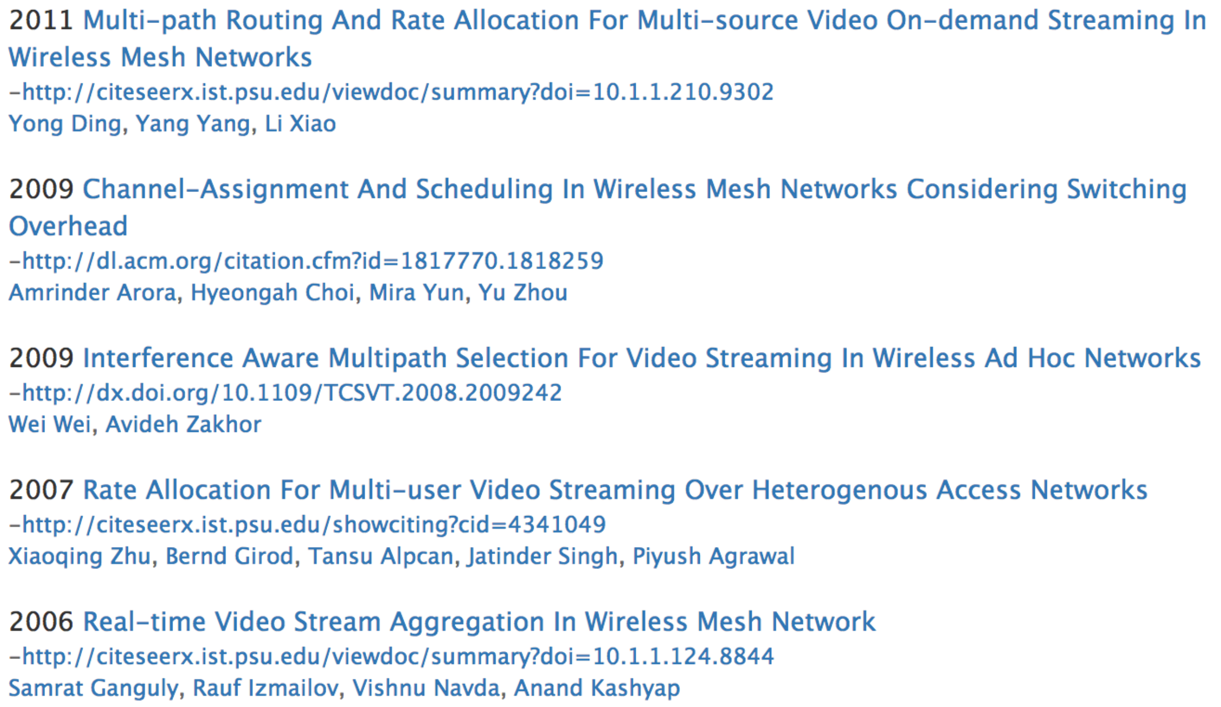
1996 Chinese All Syllables Recognition Using Combination Of Multiple Classifiers
-http://citeseerx.ist.psu.edu/showciting?cid=1413647
Liang Zhou, Satoshi Imai

2015 Effect Of Feedstock Characteristics On The Dielectric And Microwave Absorption Properties Of Plasma Sprayed NiCrAlY/Al2O3 Coatings
-http://link.springer.com/content/pdf/10.1007%2Fs10854-015-3266-y.pdf
Zhiping Sun, Liang Zhou, Shan Cui, Fa Luo, Fei Ma, Yanli Dong

2005 Security Of Dispatching Management Information System In Power System
-http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1547076
Liang Zhou, Kaipei Liu, June Li

2010 User-satisfaction-based Media Services Over Vehicular Networks
-http://eudl.eu/doi/10.4108/chinacom.2010.97
Liang Zhou, Lawrence Yeung, Joel Rodrigues

Figure 4. Recommends with Original Algorithm



2011 Multi-path Routing And Rate Allocation For Multi-source Video On-demand Streaming In Wireless Mesh Networks
-http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.210.9302
Yong Ding, Yang Yang, Li Xiao

2009 Channel-Assignment And Scheduling In Wireless Mesh Networks Considering Switching Overhead
-http://dl.acm.org/citation.cfm?id=1817770.1818259
Amrinder Arora, Hyeongah Choi, Mira Yun, Yu Zhou

2009 Interference Aware Multipath Selection For Video Streaming In Wireless Ad Hoc Networks
-http://dx.doi.org/10.1109/TCSVT.2008.2009242
Wei Wei, Avideh Zakhor

2007 Rate Allocation For Multi-user Video Streaming Over Heterogenous Access Networks
-http://citeseerx.ist.psu.edu/showciting?cid=4341049
Xiaoqing Zhu, Bernd Girod, Tansu Alpcan, Jatinder Singh, Piyush Agrawal

2006 Real-time Video Stream Aggregation In Wireless Mesh Network
-http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.124.8844
Samrat Ganguly, Rauf Izmailov, Vishnu Navda, Anand Kashyap

Figure 5. Recommends with Hybrid-Based Recommender System

Future Work

The performance of CNN is not good enough. The first reason may be the structure of the CNN model. Since the meaning of a text is determined not only by whether the words appeared but also by the positions and orders of words. This CNN model has only one layer, and the pooling method is max-pooling, which means it will never ingest the position information of the text. Thus, we will find a better model to handle the task, and GRU can be a good choice.

The second reason is the training way. What we hope to get is the ability to recognize different papers, but the current output label is which field the paper belongs to. So, this training method is not necessarily means the ability to recognize different papers. What we will do is to find a better training way.

The third reason is the noise in data. Our data is the abstracts of publication, and we assume that the abstract has very similar feature as the whole context. Many abstracts in our database, however, are incorrect, which includes messy codes or even is totally wrong information. So, we will contact scrambler group to refine the abstract data.

The time cost of the whole algorithm is too high. The complexity of the algorithm is O(n^2). This cannot be reduced at least for now, and there is no more power servers. Thus, the only choice for us is to deploy a distributed calculating system based on Hadoop to accelerate the calculation.

Reference

- Item-Based Collaborative Filtering Recommendation Algorithms 2010 Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl

- Latent Semantic Analysis 2006 Thomas K Landauer

- Convolutional Neural Networks for Sentence Classification 2014 Yoon Kim