

# LSA Based Topic Analysis on Chinese Book Titles

Linyang He

May 9, 2018

## 1 Introduction

Latent Semantic Analysis, or LSA, is a basic topic model in Natural Language Processing and information retrieval. The traditional VSM(vector space model) algorithm can not solve the problem of neither polysemy nor synonymy, which is not good for searching or information retrieval. Therefore, we try to introduce new algorithm to figure this out. In this project, we will use Chinese books' titles from Douban Book as the dataset to build a topic model based on the LSA algorithm.

## 2 Background

### 2.1 LSA

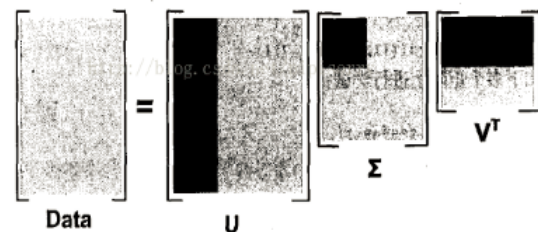
LSA introduces a 'latent' semantic information between the word level to the document level, which we could call it as "Topic" or "Concept". The traditional VSM algorithms represents different words with the similar or the same meanings into different vectors, which leads to wrongly computed cosine similarity. So the LSA is trying to find the real semantic information of some important terms in the doc. We will use the SVD matrix decom-

position algorithm to implement the LSA model. And below is how we do:

1. Build a matrix with the line denoting the terms and the column denoting the document. Notice that we will only choose the words showing in different docs. The entry of the matrix is the **TF-IDF value** of each term. We call it as the A matrix.
2. Implement the SVD matrix decomposition. That is

$$A = U \times S \times V^T$$

3. We will choose the top n important dimension of all  $U, S, V$  matrix to get off the noise as the figure shows. We denote them as  $U', S', V'$  respectively.



4. We will rebuild the  $A$  matrix using the dark part.

$$U' \times S' \times V' = A'$$

Then the  $A'$  matrix representing all the docs could include the latent semantic information. We will compare the  $A$  and the  $A'$  to find how the LSA algorithm introducing the latent semantic information.

## 2.2 Chinese Word Segmentation

Considering that we use the Chinese dataset to build the LSA model, we need to turn the Chinese sentence into segmented words. We use the Jieba segmentation in our project. Considering that the topic model is often used in information searching and retrieval, we will use the 'cut\_for\_search' model of the jieba segmentation lib. This is how it works:

```
>>> import jieba
>>> sent = '我爱复旦大学'
>>> words = [i for i in jieba.cut_for_search(sent, HMM=True)]
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\Leon\AppData\Local\Temp\jieba.cache
Loading model cost 0.991 seconds.
Prefix dict has been built successfully.
>>> words
['我', '爱', '复旦', '大学', '复旦大学']
```

# 3 Project

## 3.1 Data

We get 10 books' titles from the "youth romance"(青春) category from Douban including:

No	Title
0	致我们终将逝去的青春
1	那些年，我们一起追的女孩
2	那些回不去的年少时光
3	我们无处安放的青春
4	与青春有关的日子
5	那些忧伤的年轻人
6	凉生，我们可不可以不忧伤
7	是你来检阅我的忧伤了吗
8	我只是难过不能陪你一起老
9	时光若有张不老的脸

## 3.2 Experiment

For each title, we will segment it first. Then we would get rid off the some ignored chars and stop words including “是、的、不”. Next, we choose those key words which show in different titles including: “那些”, “青春”, “时光”, etc. And we can get the term-doc matrix as follows. We denote this matrix as  $A$

一起	0	0.54	0	0	0	0	0	0	0.4	0
你	0	0	0	0	0	0	0	0.54	0.4	0
忧伤	0	0	0	0	0	0.6	0.6	0.4	0	0
我	0	0	0	0	0	0	0	0.54	0.4	0
我们	0.46	0.31	0	0.46	0	0	0.46	0	0	0
时光	0	0	0.8	0	0	0	0	0	0	0.8
老	0	0	0	0	0	0	0	0	0.4	0.8
那些	0	0.4	0.6	0	0	0.6	0	0	0	0
青春	0.6	0	0	0.6	1.2	0	0	0	0	0

Then we implement the LSA algorithm. Here, we choose just top 2 dimension to rebuild the term-

doc matrix. We can get the result as showing in the next section.

## 4 Result

After we finished the LSA algorithm, we could get a new matrix as following. We call it as the  $A'$  matrix.

一起	0.02	0.04	0.12	0.02	0.03	0.05	0.03	0.03	0.05	0.13
你	0.01	0.04	0.11	0.01	0.00	0.05	0.02	0.03	0.04	0.12
忧伤	0.05	0.06	0.17	0.05	0.06	0.08	0.05	0.05	0.07	0.19
我	0.01	0.04	0.11	0.01	0.00	0.05	0.02	0.03	0.04	0.12
我们	0.24	0.06	0.06	0.24	0.37	0.05	0.08	0.03	0.03	0.06
时光	-0.01	0.19	0.60	-0.01	-0.06	0.27	0.11	0.16	0.25	0.68
老	-0.01	0.12	0.38	-0.01	-0.04	0.17	0.07	0.10	0.15	0.43
那些	0.03	0.12	0.35	0.03	0.01	0.16	0.07	0.09	0.14	0.39
青春	0.67	0.11	-0.03	0.67	1.07	0.04	0.17	0.02	-0.00	-0.06

And here are some results we can find through  $A$  and  $A'$ .

1. Let us take the **second** title(the second column in both matrices) “那些年，我们一起追的女孩” as the example. In  $A$ , we find that only the entry of “一起”, “我们”, “那些” is not zero. But in the second matrix,  $A'$ , we find that the value of the “青春” entry is 0.11. This shows that the title “那些年，我们一起追的女孩” has some relationship with “青春”, which is quite reasonable.
2. Let us check both the “时光” and “老” lines in  $A$  and  $A'$ . We can find that although there seems no relationship between the “时光” and “老” in  $A$ , but the value of each entry of the

“时光” and “老” lines in  $A'$  is quite similar. This meets the semantic information similarity of “时光” and “老” in Chinese!

## 5 Conclusion

Our results above demonstrates that the LSA model indeed introduces the latent semantic information. We can use this LSA model in more NLP and information retrieval tasks.

## Reference

- [1] Scott Deerwester, Susan T. Dumais, Richard Harshman. Indexing by Latent Semantic Analysis.