University of Virginia

Final paper report for University of Virginia SYS 6014
Decision Analysis Spring 2020

Product-Recommender-System

Songminghuang Huang

sh4wd

Abstract

Online shopping is all over the internet. All our needs are just a click away, the biggest online shopping website is Amazon. In this project, I built an "Amazon-like" recommender system. The system will generate two similar product recommendations and summarize the 5 review scores word cloud figures, top 20 positive/negative words ranking. The accuracy and similar products' ratings generates new payoff value that can lead to better decision. Based on the results, I think my work can help the decision-maker(shopping website) to attract and keep more loyal users.

1 Introduction

1.1 Background

Online shopping is all over the internet. All our needs are just a click away. Nowadays, an increasing number of online companies are taking advantages of recommendation engines to attract user's attention and enrich shopping potential. Over the last 4-5 years, use cases of recommender system have been expanding rapidly across many fields, and this trend is expected to continue. Everyday people receive messages and emails about product or service they might be interested in, therefore they will feel easier to purchase the right product. Companies also make personalized recommendation to spread out good product and increase the profit.

The biggest online shopping website is Amazon. Amazon is known not only for its variety of products but also for its strong recommendation system. For two decades, Amazon has been working to build a store with thousands of faces. Everyone who comes to Amazon sees it differently because the site is personalized to their personal interests. Just like when you walk into a store, the merchandise on the store shelves begins to re-arrange, placing you may need to be in front, you are unlikely to like to be in the back.Based on your current scenario and your past behavior, Amazon's recommendation system picks out a small number of items that may interest you from a library of hundreds of millions of products. The algorithm behind it is not magic, it just shares information that others have discovered with you. Everything is done automatically by algorithms. With the help of computers, people help each other implicitly and anonymously.

1.2 Objective

The popularity and usefulness of recommendation motivate me to build a recommender system based on my knowledge. I want to build an "Amazon-like" recommender system. In this project, the decision-maker is a shopping website who try to emulate Amazon's advertisement recommendation model. They want to attract more users and keep them long term using the website

2 Data

I select Amazon dataset because Amazon pays much attention collecting user data by asking them to rating their purchase and provide feedback to their experience. Therefore it's possible to gather plenty of data about each user as well as their purchase history. The more user data collected, the wider range of algorithm can be applied and the output would be more personalized.

The product recommender system bases on Amazon Review Dataset available at Amazon product data. This dataset contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014. This dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs).

3 Model of the decision problem

In this project, the decision-maker is a shopping website who try to emulate Amazon's advertisement recommendation model. So this model uses Amazon dataset to verify. Here are some mathematical explanations of the model

3.1 The decision-maker's action sets $\mathbb{A}$

A complete itemization of all the options available to the decision-maker, from which the decision-maker selects a unique choice
In this project, from the dataset, the decision-maker is foucsing on the amazon review dataset considering the reviews given by the user to different products as well as his/her reviews about his/her experience with the product(s). To find the 2 most similar items, thus $a^* \in \mathbb{A}$, the similar product can be recommended by the test data.

3.2 Sample space $\mathbb{X}$

The space from which observation data are drawn.
In this project, the sample space $\mathbb{X}$ contains product reviews and metadata from Amazon, we can observe the user reviews including the below attributes:

- reviewerID - ID of the reviewer
- asin - ID of the product
- reviewerName - name of the reviewer
- helpful - helpfulness rating of the review
- reviewText - text of the review
- overall - rating of the product
- summary - summary of the review
- unixReviewTime - time of the review (unix time)
- reviewTime - time of the review (raw)

3.3 Data generating process

This might take the form of a statistical model describing the statistical properties of random variables $X_1, X_2 \ldots X_n \in \mathbb{X}$
In this project, The dataset includes 142.8 million reviews spanning May 1996 - July 2014.

3.4 The parameter space Θ

The set of possible values for the unobserved parameters.
In this project, the unobserved parameters will be considered in the payoffs function, the accuracy ranges from 0% to 100%.

3.5 The decision-maker's prior beliefs

These beliefs describe the information or beliefs the decision-maker has about the values of the unobserved parameters before collecting data. These beliefs can generally be represented as a probability distribution over the parameter space.
In this project:

(1) People's desire to buy is unlimited. You recommend product to them, they might buy.

(2) The performance of this model is based on the accuracy and RMSE value.

(3) Decision-maker will consider the satisfaction value(payoff value) to get the decision.

3.6 Payoffs

Payoffs will be a function of the selected action, and the realized value of an uncertain random variable. In this model, the select action based on the function:

$$Payoff = Accuracy_{rec} \times Avg_{rec_ratings}$$

It means the user's satisfaction with the recommendation made by the decision-maker, that is, the average score of the product ratings multiplied the accuracy of the recommendation. For the better decision, the value of the payoff is an important indicator.

3.7 Utility function or Loss function

The most typical loss function of a recommendation systems is the symmetric loss functions such as RMSE. Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors).

$$RMSE = \sqrt{\frac{1}{N} \sum (predicted - actual)^2}$$

3.8 The rule the decision-maker will use to choose a preferred action

Based on the payoff value of the model, the decision-maker Will select the corresponding action based on the result.

4 The predictive model

The recommender system uses the ItemCollaborationFilter algorithm. Since the payoff function of this model is about the accuracy,the way to increase the accuracy is just find the most similar product. The algorithm bases on items (here in after referred to as ItemCF algorithm) pushes the recommendation system to an unprecedented scale of serving millions of users and processing millions of products. The success of this algorithm comes from the following aspects: simple and scalability, can often give surprising and useful recommendations, recommendations can be updated immediately base on new user information.
The tool can provide the recommended suggestion to the decision-maker, and the decision-maker will use the information generated by this tool to recommend the similar product to increase benefits and profits.
K - Nearest Neighbors is used to perform ItemCollaborationFilter filtering. The traditional KNN calculated the similarity of users and items. After calculation of similarity, KNN predicted user-item ratings by the existing ratings that a user given to the k nearest neighbors of specific items. The formula was as follow:

$$r_{\mu j} = \mu_j + \frac{\sum_{i \in P_j(\mu)} Sim(i,j) \times (r_{\mu i} - \mu_i)}{\sum_{i \in P_j(\mu)} |(i,j)|}$$

k -NN does not have a loss function that can be minimized during training. In fact, this algorithm is not trained at all. The only "training" that happens for k-NN, is memorising the data (creating a local copy), so that during prediction you can do a search and majority vote. Technically, no function is fitted to the data, and so, no optimization is done (it cannot be trained using gradient descent).
To calibrate this model, Amazon Instant Video dataset was used to train, the tool will calculate and get two similar product about the reference product and generate 5 word figure about the 5 scores. Also it will sum up the 20 most positive and negative words respectively.

5 The expected value of information

(1) For shopping websites, this will help them make more effective recommendations, which can stimulate user consumption and leave more users.

(2) For buyers, it can let you buy more or better product you like.

6 Results

The predictive tool has 3 parts, First it will convert the json file to the csv file, then the recommendation part will give the decision-maker the recommendation results and 5 figures about 5 scores, also the decision-maker will get the positive and negative words counting.

6.1 Recommendation

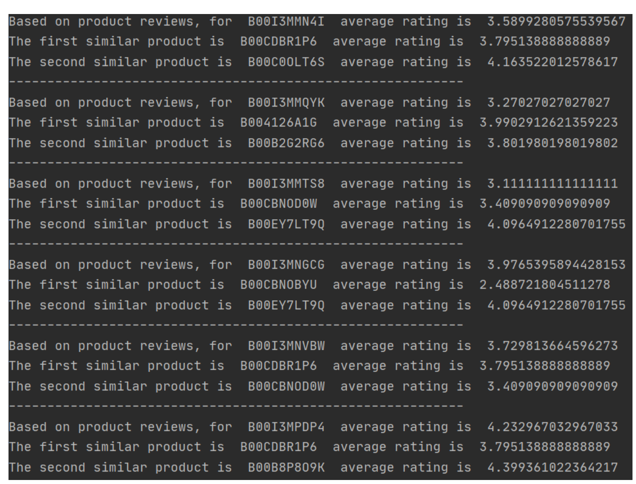
```
Based on product reviews, for  B00I3MMN4I  average rating is  3.5899280575539567
The first similar product is  B00CDBR1P6  average rating is  3.795138888888889
The second similar product is  B00C0OLT6S  average rating is  4.163522012578617
-----------------------------------------------------------
Based on product reviews, for  B00I3MMQYK  average rating is  3.27027027027027
The first similar product is  B004126A1G  average rating is  3.9902912621359223
The second similar product is  B00B2G2RG6  average rating is  3.801980198019802
-----------------------------------------------------------
Based on product reviews, for  B00I3MMTS8  average rating is  3.111111111111111
The first similar product is  B00CBNOD0W  average rating is  3.409090909090909
The second similar product is  B00EY7LT9Q  average rating is  4.0964912280701755
-----------------------------------------------------------
Based on product reviews, for  B00I3MNGCG  average rating is  3.9765395894428153
The first similar product is  B00CBNOBYU  average rating is  2.488721804511278
The second similar product is  B00EY7LT9Q  average rating is  4.0964912280701755
-----------------------------------------------------------
Based on product reviews, for  B00I3MNVBW  average rating is  3.729813664596273
The first similar product is  B00CDBR1P6  average rating is  3.795138888888889
The second similar product is  B00CBNOD0W  average rating is  3.409090909090909
-----------------------------------------------------------
Based on product reviews, for  B00I3MPDP4  average rating is  4.232967032967033
The first similar product is  B00CDBR1P6  average rating is  3.795138888888889
The second similar product is  B00B8P8O9K  average rating is  4.399361022364217
```

Figure 1: Sample recommendations

In Figure1, we can see some example outputs about the recommendation model. For a selected product, it will get two most similar product recommendation candidates, and their average ratings. For example, based on product B00I3MMN4I, we can get first product B00CDBR1P6 and second product B00C0OLT6S, and the related ratings are 3.5899, 3.7951 and 4.1635.

6.1.1 Performance index

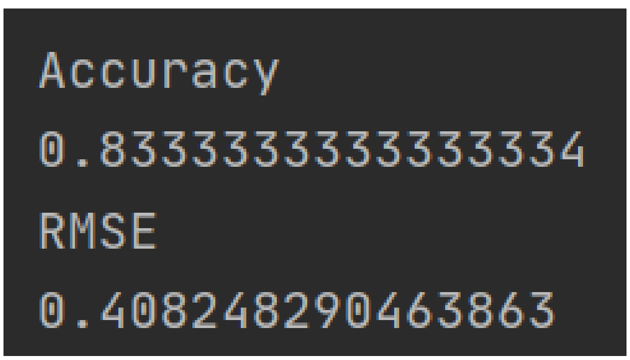

Figure 2: Performance index

In Figure2, we see the accuracy of this model is 80% and the RMSE value is 0.4082. We can calculate the payoff value of the example recommendation mentioned in 6.1.1 is 3.1384 ($0.8 \times \frac{3.7951+4.1635}{2}$). In other words, this product is recommended to the user, after the user buys it, they will get a 3.1384 rating.

6.1.2 5 scores figures

Figure 3: 5 review scores

In Figure3, from top to bottom, as the score increases, the positive word and its frequency also become higher and higher.

6.2 Words counting

Table 1: Top 20 positive

Word	Coefficient
love	6.751873
great	6.702032
enjoyed	4.327477
good	3.831966
fun	3.727715
hooked	3.537747
excellent	3.512835
season	3.361284
best	3.265526
loved	3.112800
entertaining	3.098135
glad	2.989821
enjoy	2.939723
favorite	2.904300
hope	2.897397
amazing	2.812968
keeps	2.742008
liked	2.562549
series	2.544215
wait	2.445722

The Table1 is the top 20 positive words counting.

Table 2: Top 20 negative

Word	Coefficient
money	-2.790238
dumb	-2.859187
horrible	-2.886813
care	-2.894564
flat	-2.897822
didn	-2.911112
poor	-3.145933
don	-3.269409
bad	-3.402045
worse	-3.585556
couldn	-3.716816
stupid	-3.884865
awful	-3.885744
worst	-3.903444
just	-3.962601
poorly	-3.980735
minutes	-4.175993
terrible	-4.646741
boring	-6.086880
waste	-6.345231

The Table2 is the top 20 positive words counting. You can see with 6.1.2, the positive reviews have more positive words and vice versa.

7 Conclusions

In this project, based on my interest in exploring a product recommendation system, I made its a decision model and prediction algorithm. At the same time, as a statistical verification requirement, I also collected 5 word frequency graphs scored, as well as a ranking of 20 words on both positive and negative sides to observe the words distribution and ranking. I also can calculate the payoff value according to above mentioned parameters.The accuracy and similar products' ratings generates new payoff value that can help the decision-maker lead to better decision. I think my work can help the decision-maker(shopping website) to attract and keep more loyal users. Future work can be considered more algorithm, increasing the performance is also a possible future research activity.

8 References

(1) Amazon product data:
http://jmcauley.ucsd.edu/data/amazon/

(2) Collaborative Filtering Recommender Systems J. Ben Schafer, Dan Frankowski, Jon Herlocker, Shilad Sen
http://link.springer.com/chapter/10.1007/978-3-540- 72079-99

(3) Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering R. He, J. McAuley WWW, 2016

(4) Image-based recommendations on styles and substitutes J. McAuley, C. Targett, J. Shi, A. van den Hengel SIGIR, 2015