Recommendation System Challenge Project Report

COSI 129A: Introduction to Big Data Analysis

Professor:
Hongfu Liu

Brandeis Student Team:
Daniel Zhang
Matthew Millendorf
Wenxiao xiao

May 2nd, 2019

Table of Content

1. Introduction

1.1 Background

The internet and the digital age has given birth to entirely new paradigms of commerce. Chief among them is the “Long Tail Hypothesis” or the Long Tail Effect. First proposed by prominent British-American entrepreneur Chris Anderson, the theory explains the opportunity and challenges of e-commerce in the twenty-first century. For low marginal-cost products, such as music, there are millions and millions of songs to choose from. In similar vein, the internet allows for business’ consumer markets to not be constrained by geospatial location. As opposed to one-hundred years ago, a consumer does not have to buy from a business located close to them. Similarly, a business does not have to sell only to their local market. The e-commerce market place creates new opportunity for businesses to sell and consumers to buy many more products.

With this great opportunity, there comes an obvious opportunity cost. An entire body of research has arisen from the problem of determining which products to recommend to a consumer. With thousands, if not millions of goods/services from a single firm, the most profitable ventures will understand which of their products to advertise and showcase to specific users. As Chris Anderson explains it, the future of business is less about selling a few items many times, and about sellings many items a few times. The theory is that the area under this curve is far larger than the area under the curve for a few products but many sales because of the ‘long tail’.

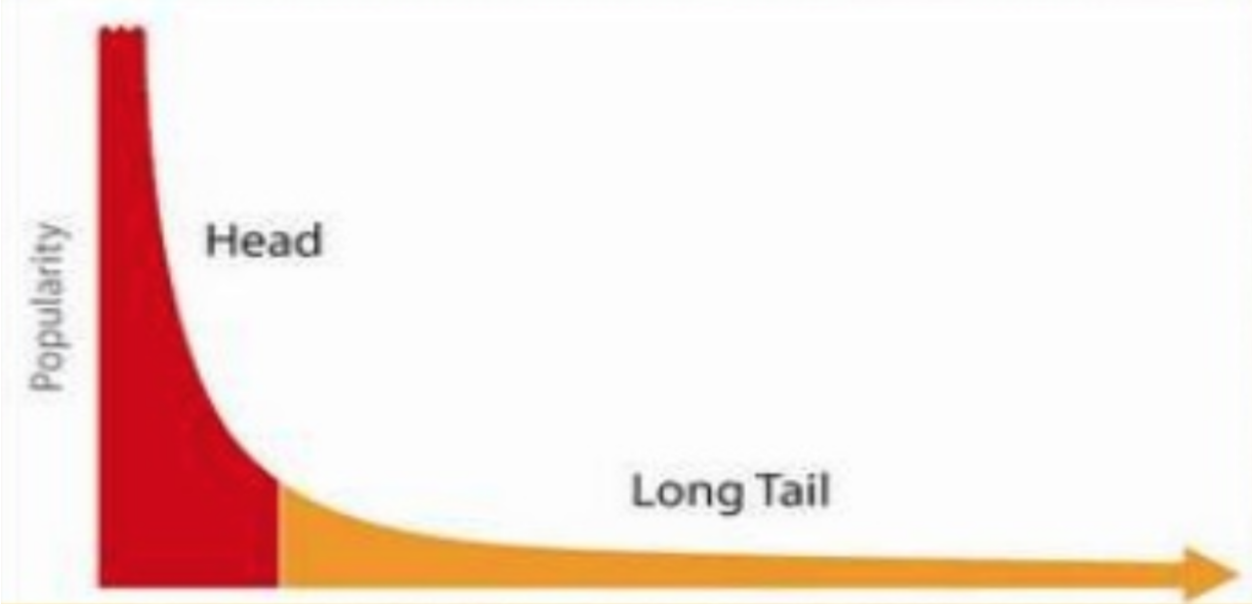
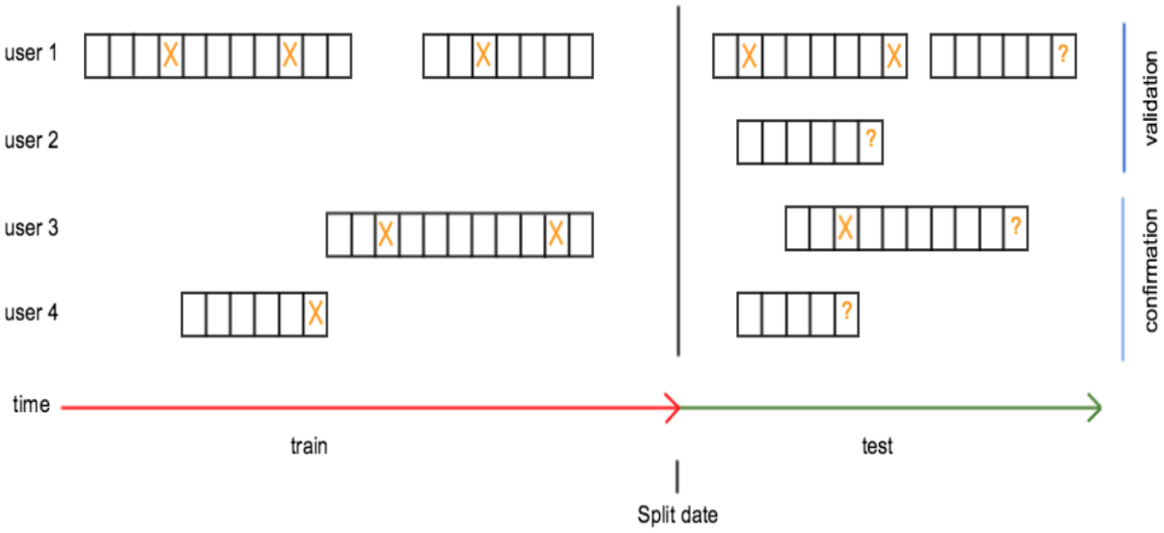
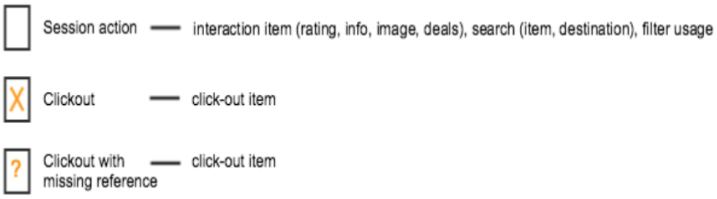


Figure 1: The Long Tail Hypothesis.

1.2 Problem Definition

The significance of such problems has risen over the years and as a result, the Association for Computing Machinery hosts a yearly recommendation system competition. This year's competition is to build a recommendation engine for the popular hotel search website, Trivago. The data provided for this challenge consists of three datasets, a training dataset, a test dataset, and metadata about the items, which are hotels. The training dataset is comprised of 910,683 unique sessions on 730,083 users. Features in the dataset were for the user profiles, such as the device the user was using. However, the core value came from the user interactions within a session. The sessions consisted of user interactions with items, such as clicking out on a hotel or viewing a photo. The problem lies within the test data; given a user's interactions during a session, can we predict which items, or hotels, this user will want to purchase. The "split" of this data can be seen in figure 2.


user 1
user 2
user 3
user 4
validation
confirmation
time
train
test
Split date


Session action
interaction item (rating, info, image, deals), search (item, destination), filter usage
Clickout
click-out item
Clickout with missing reference
click-out item

- **clickout item**: user makes a click-out on the item and gets forwarded to a partner website. The reference value for this action is the item_id. Other items that were displayed to the user and their associated prices are listed under the ‘impressions’ and ‘prices’ column for this action.
- **interaction item rating**: user interacts with a rating or review of an item.
  The reference value for this action is the item id.
- **interaction item info**: user interacts with item information.
  The reference value for this action is the item id.
- **interaction item image**: user interacts with an image of an item.
  The reference value for this action is the item id.
- **interaction item deals**: user clicks on the view more deals button.
  The reference value for this action is the item id.
- **change of sort order**: user changes the sort order.
  The reference value for this action is the sort order description.
- **filter selection**: user selects a filter.
  The reference value for this action is the filter description.
- **search for item**: user searches for an accommodation.
  The reference value for this action is the item id.
- **search for destination**: user searches for a destination.
  The reference value for this action is the name of the destination.
- **search for poi**: user searches for a point of interest (POI).
  The reference value for this action is the name of the POI.

Figure 2: Training and test data

1.3 Project Objective

So the objective of the project is to predict which item the user will click out and rank all the candidate hotels based on the users’ previous action in the session and maybe also based other users with similar interests with the user. Specifically, by diving into the data and find out related features, then use different recommendation techniques to build utility matrix and predict the click out probability of each hotel. For example, we may calculated the distribution of each user’s score on different attributes and then calculate the similarity score between each user and each hotel base the user profile and hotel profile. Also, there are some other techniques in the following chart which tried and get a improvement on our prediction

RS	Input	Core assumption	Work mechanism	Pros	Cons
Content-based filtering (CBF) [125]	User, item content information	A user likes what he/she liked	Matching up user profile against item content	Simple and straightforward, can handle cold-start issues	The assumption may not fit real-world cases well
Collaborative filtering (CF) [115]	User-item interaction data	A user likes what he/she liked	Modeling user-item interactions	Effective and relatively simple	Easily suffering from sparsity and cold-start issues
Context-aware RS [5]	Users, items, context and user-item interaction data	A user may have different preferences under different contexts	Modeling user-item-context interactions	Incorporating more information and fitting the real-world cases better	Data availability and sparsity issues
SBRS [125]	Session data	User preference changes along with the corresponding session context	Recommending items that have occurred in a similar context	Considering the user preference evolution, which fits the real-world cases better	Ignoring user's general and long-term preference

accuracy.

Figure 3: Different types of user interactions within a given session.

2. Data

2.1 Data Description

There are mainly three datasets which is meta data, train data and test data.

The first one is metadata which include 2 columns and 927143 rows. It means that we have 927143 unique hotels and each hotel has some properties.

Specifically, the first column is the item id which is a unique key to target each hotel. Then the second column is a "|" separated string which is actually is the properties of each column. If we split the second column we could see that there are many properties for each hotel and these properties may be used to build utility matrix for user profile and hotel profile for recommendation uses.

```
5101 ['Satellite TV', 'Golf Course', 'Airport Shuttle', 'Cosmetic Mirror', 'Safe (Hotel)', 'Telephone', 'Hotel', 'Sitting Area (Ro
oms)', 'Reception (24/7)', 'Air Conditioning', 'Hypoallergenic Rooms', 'Cable TV', 'Hotel Bar', 'Pool Table', 'Bathtub', 'Sati
sfactory Rating', 'Room Service', 'Luxury Hotel', 'Terrace (Hotel)', 'Television', 'Minigolf', 'Business Hotel', 'Shower', 'Cot',
'Gym', 'Hairdryer', 'Hypoallergenic Bedding', 'Accessible Parking', 'From 3 Stars', 'Good Rating', 'Radio', '4 Star', 'From 4
 Stars', 'Family Friendly', 'Desk', 'Tennis Court (Indoor)', 'Balcony', 'WiFi (Public Areas)', 'Openable Windows', 'Express C
heck-In / Check-Out', 'Restaurant', 'Laundry Service', 'Ironing Board', 'Tennis Court', 'From 2 Stars', 'Business Centre', '
Bowling', 'Conference Rooms', 'Electric Kettle', 'Accessible Hotel', 'Porter', 'Bike Rental', 'Non-Smoking Rooms', 'Car Pa
rk', 'Safe (Rooms)', 'Fitness', 'Fan', 'Flatscreen TV', 'Computer with Internet', 'WiFi (Rooms)', 'Lift', 'Central Heating']
```

Figure 4. Sample Metadata

Then another useful dataset is the training dataset which is a session based user action. From the chart we could see that this is all the actions taken by a user in a session. Specifically, the user firstly search hotel for a destination, then the filter was used. The user then interact with all the items under the filter he set.

We could notice that the user click the item after he interact with "deals" and other item related information. Then we have a feeling here which is that the user's action within the search session is very important. For example, if a user interact with an item more than one time, then the probability that this user will click this item is very high.And this assumption is also worked in our later modeling part.

user_id	session_id	timestamp	step	action_type	reference	platform	city	device	current_filters	impressions	prices
93F7WGHBPO3A	569f5ea70df51	1541543231	1	search for destination	Barcelona, Spain	US	Barcelona, Spain	desktop			
93F7WGHBPO3A	569f5ea70df51	1541543269	2	filter selection	Focus on Distance	US	Barcelona, Spain	desktop	Focus on Distance		
93F7WGHBPO3A	569f5ea70df51	1541543269	3	search for poi	Port de Barcelona	US	Barcelona, Spain	desktop	Focus on Distance		
93F7WGHBPO3A	569f5ea70df51	1541543371	4	interaction item deals	40255	US	Barcelona, Spain	desktop			
93F7WGHBPO3A	569f5ea70df51	1541543425	5	clickout item	40255	US	Barcelona, Spain	desktop		6744|40181|40630|84610|2282416|1258693|974937|147509|128238|7998246|40255|3058538|1637385|40285|147502|921707|40849|6757|12770|893733|685091|147522|40708|860451|6819	162|91|218|190|176|365|272|159|139|240|136|5099|164|116|90|192|191|213|109|178|131|128|168|101|331
93F7WGHBPO3A	569f5ea70df51	1541543741	6	search for item	81770	US	Barcelona, Spain	desktop			
93F7WGHBPO3A	569f5ea70df51	1541543770	7	interaction item info	81770	US	Barcelona, Spain	desktop			
93F7WGHBPO3A	569f5ea70df51	1541543813	8	clickout item	81770	US	Barcelona, Spain	desktop		6832|40396|6621784|40197|6743|147488|40635|6177052|6742|1319782|40763|945255|83855|39937|1870125|1354432|6812|82400|40181|6834|81770|5056102|40797|923935|40284	347|245|199|65|359|233|227|270|294|625|208|174|121|217|226|616|293|166|91|198|274|272|123|130|131

Figure 5. Sample Train Data

Then for the testing data, it is nearly same with the distribution of training data except that the last click out item is missing so that we need to propose appropriate model to predict the ranking of all the candidate items and try to put the one that user actually clicked to the first several places.

2.2 Data Processing

Different methods was used for processing data based on different assumptions and different models.

Firstly, in order to do the content based recommendation we build utility matrix for the user profile and item profile. Specifically, we could easily get the hotel profile by iterating through each hotel and check whether they have every specific property. The value for that property is 1 if they have and will be 0 if not. So that there is a matrix consists of 0 and 1 for each property of each hotel.

Then since we have the data that which user click which hotel, we could get the users' score for each hotel. Here we assume the score is 1 if the user clicked the hotel and score is 0 if not. Then easily we could use these 2 matrix to calculate the user profile on each property which is actually the user's propensity score for each property.

Another method we used to process data is based on the session behaviour of each user. We tried to build a utility matrix for the user-item interactions since we found that the user-item interaction within each search session is also very important for the prediction and improved a lot of prediction accuracy.

Specifically, we summed up the total click outs for each user-item pair so that we know the click number distribution for each hotel and also for each user. Then we selected 6 user interaction types and assign a weight to the user-item interaction score so that its not a matrix consists of only 1 and 0. Instead, it has kind of a continuous score which is more meaningful on some extent

since uses' propensity for every hotel is not a 1 and 0 score, there must exist several different level

```
flags = np.array(["search for item", "interaction item info",
                  "interaction item deals", "interaction item image",
                  "clickout item", "interaction item rating"])
```

Figure 6. Selected Interactions for assigning score purpose

Here is the result after doing such kind of data processing. It could be seen clearly is that for each user-item pair, the score become a relatively continuous number which helped a lot in the model building and prediction part.

user_id	reference	score3
0001VQMGUI65	2019467	10
0001VQMGUI65	3133074	36
0001VQMGUI65	4521140	2
0001VQMGUI65	477811	10
0001VQMGUI65	950829	10
000324D9BBUC	1000915	4
000324D9BBUC	1241375	3
0003QTCX5MJX	2195060	10
0004IOZI7CKF	110985	10
0004IOZI7CKF	2627602	10
0004IOZI7CKF	3381482	2
0004IOZI7CKF	7822344	10
0004WCFRV3FB	1439375	10
0006W0R5A5V8	6776722	10
00071784XQ6B	22721	10
0008B0X0HC39	2762974	10
0008BO33KUQ0	1949601	20
0008BO33KUQ0	2143854	10

Figure 7. Utility Matrix based on Session-based Assumption

3 Modeling

3.1 Popularity Baseline

The first model being investigated is the the popularity model which is based on a very simple assumption. It is that it is highly likely that user will select the popular hotel.

So the detail is just sum up the counts for each hotel so that the popularity for each hotel can be ranked based how many times they are clicked. Then when you predict the last click out for each session, just simply rank all the the candidates based on their popularity and it shows that the result is not bad which is 0.288 and the screen shot for this model is showing below.

Status	Score
valid	0.288448

3.2 Collaborative Filtering & SVD

In order to improve the performance of prediction, one guess is that users are similar within their small group and different with users from other users. Similarly, hotels may also have the concept of cluster.

So, we decided to user K-means clustering technique to cluster users into 4 clusters and hotels into 2 groups. Specifically, the package we used is called “Surprise” which is a RecSys package and use it to do the co-clustering job. After that, we conduct SVD decompression for the utility matrix and the final prediction accuracy is 0.20 and 0.21 which is even not good as the popularity model.

Status	Score
valid	0.201237

The possible reason is that the data is so sparse so that the co-clustering cannot work very well and the prediction is not accuracy.

In addition to SVD, we also tried a simple ensemble method which is under the assumption that the prediction of SVD method is not under the same perspective if the popularity model. So, we tried to ensemble these 2 models together by using the meaningful prediction from SVD prediction and result from the popularity model and got the result which is 0.29, higher than the single result got from any model of these 2.

Status	Score
valid	0.292355

3.3 Rank by Utility Matrix

Here we changed the assumption after looking into the data since we found that the users' session-based interaction is highly correlated with the final click out.

For example, it the user search the item directly, interact with the image of the hotel or the user interact with the deal of the hotel, then it is highly likely the user will click the hotel. After find this, we decided to add this information into the utility matrix.

Specifically, for each user-item pair, there is a score which is 1 or 0 originally which is based on whether the user clicked the item or not. But right now we give weights to the users' interaction so that the score for each pair become a relatively continuous data and become more meaningful. Then we could easily ensemble this utility matrix with popularity model by using popularity model as a base. Then for each recommendation set, if we could find the user-item pair in utility matrix, then rank them by the score and put is at the first several position of the recommendation.

Finally the accuracy improved to 0.50.

4 Challenges and Further Recommendations

Firstly the data is sparse which lead to a result it's hard to find effective pattern by looking at the data and then feature engineering part can not be done effectively within short time. For example, there are 948041 users and 927142 hotels which is a lot. However, the pair between user and hotel has only 152467 pairs which means that there is few data for many users and it would be hard to learn their behaviour pattern.

Another challenge is it's hard to do a good job on feature engineering. We looked into the data and tried to do some feature engineering part but sometimes the result becomes better and sometimes worse so we don't have enough time to do this kind of experiment to find the way to do perfect feature engineering and then do the prediction.

The further recommendation for the project is understanding the data. We gradually found that it is very important to understand the data and how each feature may influence other features and the target variable. For example, the assumption that users' interaction may influence their future click out behaviour is a useful insight by looking into the data and bring us lots of improvement.

5 Conclusion

Regarding the project, we firstly looked into the data a little bit and then processed the data and get the utility matrix for recommendation model building usage. After trying different models such as popularity-based model, collaborative filtering method, we increased the accuracy step by step.

During the process of trying different model, we also dive into the data and try to make better understand of the data and do better feature engineering. And the fact shows that the work of keep understanding the detail of the data works and the final prediction accuracy is 0.50 which is a huge improvement from the baseline model.