USER BASED COLLABORATIVE FILTERING RECOMMENDER SYSTEMS

CHAITANYA DEVAGUPTAPU
Keshav Memorial Institute of Technology affiliated to JNTUH
E-Mail:chaitanyadeva96@gmail.com

ABSTRACT
This document mainly discusses the building of a Recommender system to make predictions based on reviews of books. Given a user item pair our main goal is to predict whether the user purchased the item, to show user based collaborative filtering is a good approach we addressed this problem user three methods which are discussed in detail in the following sections.

Keywords- *personalized recommendation; jacquard similarity, user based recommendation, collaborative filtering.*

INTRODUCTION

The world today has vast amounts of data, being data rich companies are looking for more accurate ways of using this data. Companies are using their huge amounts of data to give recommendations for users. Recommender Systems are to help people discover new content, find the content we were already looking for, discover which things go together, personalize user experiences in response to user feedback, recommend incredible products that are relevant to our interests and identify things that we like, which are basically to model people's preferences, opinions and behavior. Recommender system model in this document can be used for evaluating users' purchase habits, predicting whether or not users may buy some items. Then we can recommend book items to users. Recommender systems have become extremely common now a days and are utilized in a variety of areas, some popular applications include movies, music, news, books, research articles, and several other products in general . If you have used services like Amazon, Job Board or Netflix, it is often to find some recommendations suggesting items for you to buy, jobs you may be interested or movies to watch. We will use this idea to build a similar application like them to make predictions based on reviews of books.

RELATED WORK

There are many algorithms that could be applied on user demographics to predict auser preference. User-based, Item-based, and Model based methods are more popular ways of predicting a user preference. The number of users, items, or clusters in each one respectively will determine performance of the function. However, the most popular and common one is User-based Collaborative Filtering. This document discusses three approaches that we implemented for making the predictions.

1. First, the baseline prediction was to find the most popular products that accounts for 50% of purchases in the training data, the model would return '1' whenever such a product is seen at test time, '0' otherwise.
2. Then we applied logistic regression, here we found two ways to generate features and response in training and validation set, we will talk in detail later.
3. Finally, we built an user-based collaborative filtering which was implemented by Jaccard similarity between users [1]. Besides, we refined the percentage threshold that works better than a simple 50% threshold.

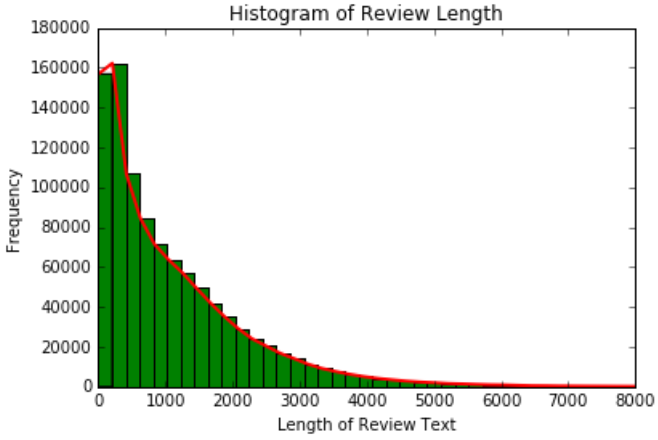
DATA

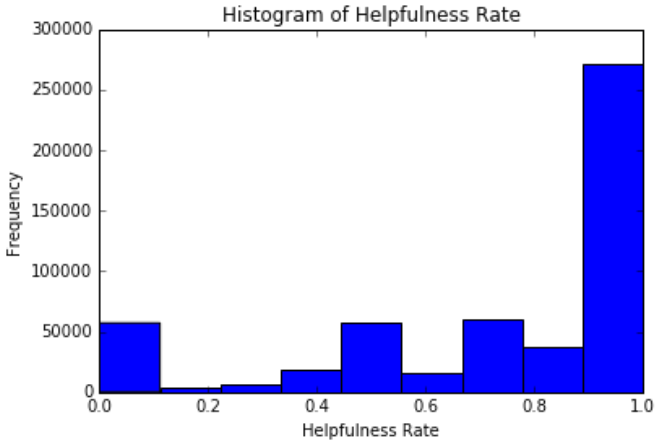
Dataset we have used for building this RS is the Amazon book review dataset. The dataset consists of 1,000,000(1milllion) reviews, dataset has been split to suffice the training and testing to make the RS more dynamic on unseen data. The data contains several attributed related to the purchase of the book (itemID, reviewerID, helpfulness rate of the review, summary of the review, rating of the review, review time, category) .In order to make the model more robust, the test set has been constructed such that exactly half of the pairs correspond to purchased items and the other half do not.

Considering the basic statistics of the data , though the dataset on a total has 1000,000 user item pairs , but there are only 35736 unique users and 37801 unique items. The average user frequency (number of reviews a user has contributed) is ~28 and the average item frequency (number of reviews for a particular book) is 26.4 . From this information we can infer that each user has many purchase records and each item has been reviewed by different users , so the item frequency and user frequency can be considered as important features when making the prediction .The 'rating', 'reviewText' and 'summary' features seem to be useful features, because we can conclude how much each user's likes the item they purchased, but in the original dataset, we do not have 'rating' information of 'non-purchased' user-item pairs , so we decided to drop this feature as we are making predictions to estimate the accuracy of the model . Since we only care about user-item pairs in this project, we do not need to be worried about abnormalities problem in any other features, in the reveiwerID and itemID features, there aren't really any outliers and missing values.

We need to be careful when generating a validation set for this data . We have generated user-item pairs for based on all users and items which also include the non-purchased items. Instead of just taking the last 100k reviews for validation, we instead took 50,000 , but also randomly select 50,000 'non-purchases' by randomly selecting a user and an item, and checking that they do not already show up together in the training set. In the Collaborative Filtering model, since it takes really a long time for prediction when testing set is large, so we used a smaller size validation set (500 Reviews + 500 non-purchases) .This

may cause the accuracy of Collaborative Filtering model to be lower than using the same size of original validation set. We should consider this situation when comparing three models.
Getting into the exploration and visualization part examining the length of the review ,we found most review content are not very long. Usually if the review are really long, the reviewers may complain about the items, so we can use this as another implicit evidence which shows that most users are satisfied with their purchase, the histogram of the length of review text makes the behaviour discussed to be evident ,visualizing the helpfulness rate attribute of the data , it is very straightforward to show us that usually users reviews are considered to be very helpful and useful since the frequency of helpfulness rates between 0.9-1.0 are much higher than others. So users' feedback is a very useful resource to affect other
people's feelings about items .Both the reviewlength and the helpfulness rate histograms are shown below .





METHODS IMPLEMENTATION

A. *Baseline popularity model :*
Baseline for this task simply ranks products by popularity, and returns '1' for popular products, by finding the most popular product that account for 50% of purchases . Based on the data exploration done earlier, it is evident that popularity is a very important feature. In reality, usually the popular items are purchased in high probability, so we chose the baseline popularity model to implement this idea.
First we generated a popularity list to record how many times each item was purchased and sorted items by decreasing popularity . Then we picked out the most popularity items that accounts for 50% of purchases.(The sum of purchases of these items is 50% of the total purchases). To predict the pairs 'user-item' in test set, if the item showed up in the most popularity list, we predict this pair as '1'(purchase), otherwise, considered the pair as '0' (non-purchase) .The accuracy of the baseline item popularity model on the validation set is 0.6366 (threshold = 0.5), the accuracy of the baseline user popularity model on the validation set is 0.65174 (threshold = 0.5) .
B. *Logistic Regression:*
As discussed earlier our second implementation would be the LR. LR is good and common choice for classification problem, here we used the popularity which got from baseline model as a feature in a logistic regressor. There are lots of other classification machine learning algorithms, like SVMs. But we do not use SVMs here because they do not work well in very large data sets and the training time happens to be cubic in the size of the dataset [2].
In order to get the model feature matrix X for logistic regression, we will convert features represented as lists of dict object to matrix, each row of the matrix represents for each
user-item pair .We build a matrix to represent all the pairs as a feature matrix [3].
In order to implement logistic regression, our first step is to construct the response in training set,because in the training set are all "purchase" i.e we only have label 1 without any label 0 .The idea to construct response is to use the popularity of items, if the pair in training is considered to be popular, then we label the pair as '1', otherwise as '0'. In other words, we generate the response by using baseline popularity model to predict on the training set . In the data preprocessing step ,we have already got the model matrix X by using DictVectorizer[4] to convert features list, now we got a one-zero matrix to represent the purchase information, each row in the matrix represents for each user-item pair. The accuracy of logistic regression with threshold = 0.5 is 0.65174.
C.*User-basedCollaborative Filtering :*
This is a method that performs recommendation in terms of user similarity. Evaluated if items were purchased by users based on similarity measures between users. The items we may infer to be purchased by a user are drawn from those purchased by similar users. The collaborative filtering is a good idea for recommender system model, but we still have a few problems, since this method is actually kind of slow given a huge dataset . This method is implemented by Jaccard Similarity between users [5]. Collaborative filtering using Jaccard Similarity between users is a standard implementation which defined as the intersection of two users divided by the union of them ,this method works best when the user space is large and more or less insensitive to user size. In this project,

we have a large user space and collaborative filtering is a good idea for recommender systems. Collaborative filtering is to build a matrix which shows the relationship between users or items, then we can infer the preference of an user by checking the matrix and matching the user's information.

Similarity Measure : We measured the similarity by using the Jaccard Similarity , which is given by the following formula :

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

To simplify the problem, we considered if two users purchased at least one common item, then recorded the similarity between two users as 1, otherwise is 0 .

Utility Measure : First we built a matrix, the row label represents each user and column label represents each item. The value in this matrix is either 1 or NaN, 1 represents the user purchased the item and the value is given for each user-trim pair. The matrix is sparse, meaning that most of the entries are unknown .Here is an example of a matrix, describing three users: user1, user2, user3. The available items: item1, item2, item3 :

item	item1	item2	item3
user			
user1	1	NaN	1
user2	1	NaN	NaN
user3	NaN	1	1

Prediction Process: The goal of the recommender engine is to predict the 'NaN' in a matrix. For example, we want to predict user1-item2, we checked other users who purchased item2, here we found user3 purchased item2. The second step is to measure the Jaccard similarity between user1 and user3, since user1 and user3 have common purchase item-item3, so according to the definition of Jaccard similarity above, the similarity between user1 and user3 is 1. So we predict the value of user1-item2 as 1. As another example of prediction, suppose we want to predict user2-item2, we checked other users who purchased item2 and we found user3. Then we measure the similarity between user2 and user3, they have no items in common, so the similarity of user2 and user3 is 0. As a result, we predict the value of user2-item as 0. Here, because this model is time-consuming when test set is large, so we reduced the validation set to 1000. The accuracy of user-based collaborative filtering is 0.771 .

RESULTS AND JUSTIFICATION

1. Baseline Popularity Model: The most efficiency model, very fast, but the performance is not good as my expectation. The user popularity model performs better than the item popularity model, the best performance of the user popularity model is 0.63357.

2. Logistic Regression: Slower than the baseline popularity model, the performance is similar to the baseline popularity model.

3. User-Based Collaborative Filtering: Performs best with really high accuracy, but when test set is large, this method is slower than the above two. The accuracy is 0.771 on the validation set.

Although the collaborative filtering is the slowest one, we still considered this model as our final model, since it's performance is really good compare to the baseline and logistic regression. To demonstrate the result of our final model is significantly better than the Baseline popularity model, we considered running the code 20 times with different pools of randomly selected data and applied t-test to justify the hypothesis, we got accuracies :

0.774, 0.788, 0.775, 0.761, 0.795, 0.8, 0.793, 0.769, 0.766, 0.779, 0.794, 0.788, 0.795, 0.784, 0.8, 0.8, 0.769, 0.776, 0.782, 0.798 .

We got t statistic = 47.8 and p value = 2.856e-21, p value is smaller than 0.05, so we can reject the hypothesis : mean of accuracy is equal to 0.65, which demonstrate that the result of collaborative filtering model is significantly better than the baseline approach

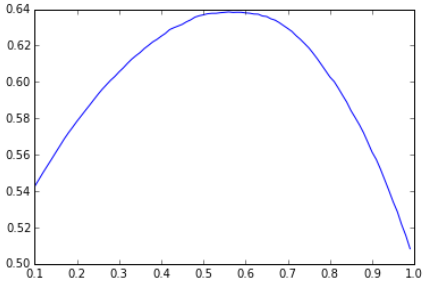
Finally we decided to use user-based collaborative filtering to do prediction, the result of the validation set is 0.771 which is higher than the accuracy we expected. But the accuracy was got by just testing on 5000 validation set, which is not large enough, so the reliability of the accuracy should be doubted. But we still have reason to justify that the collaborative filtering works better than the baseline and logistic regression, this is because when generated the validation set in the first two methods, we got non-purchase data set by generation instead of from real world, so the validation set is not as reliable as what we used in the last model.

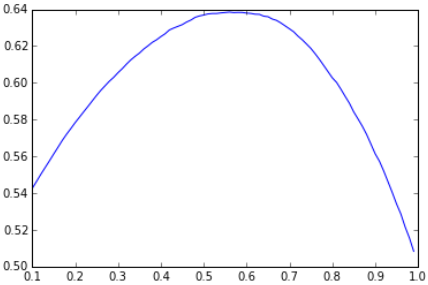
CONCLUSION

This project is a recommender system problem, the main idea is to evaluate people's preference and purchase habits and make some recommendations. In this specific problem, we are given 1 million Amazon book reviews, which provided us purchase information, our target is to predict when user purchased the item given some user-item pairs. To solve the problem, first we generate a user-item list from the original list of review information. Then we reconstruct the training and validation data by randomly selecting user-item pairs which do not show up in the training set as non-purchase pairs. As for models, we considered three models: popularity model, logistic regression and collaborative filtering. Here is a table displaying average accuracy of different methods over 20 times:

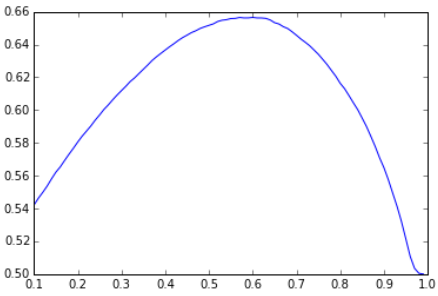
	Average Accuracy
item popularity	0.63851
user popularity	0.65647
logistic regression	0.65647
collaborative filtering	0.78430

For popularity model, we considered the most popular items that accounts for 50% of purchases as label 1, and to refine the result, we compare the accuracy of different thresholds and found the best threshold is 0.56
Similarly, we considered the most 'popular' users that accounts for 50% of purchases as label 0,and compare the accuracy of different thresholds to refine the result the best threshold 0.6





For logistic regression, we combined the popularity in popularity model as a feature in logistic regression model, in other words, we use the prediction of popularity model on training set as the model response, and then transform the feature list to one-zeroes matrix to fit the model, also we select the best threshold by comparing the results of different thresholds, the best threshold is 0.6.



For collaborative filtering, we implemented this model by using Jaccard similarity to measure the similarity between users and evaluated users' preference. We met some difficulties when we worked with the entire dataset for collaborative filtering model, working on the whole dataset is time-consuming, so we changed the validation set size to be smaller. This may resulted in a lower accuracy and undermine the comparison of three models. As for interesting aspect of this project, recommender systems are increasingly popular application of Machine Learning in many industries. By evaluating the users and items popularities and users similarities, we can infer the users preference and purchase habits, which is a very practical and useful application in a variety of areas, we can use this idea in many other places such as music , movies and news recommendation, social tags or online dating.

FUTURE WORK

1.Collaborative filtering in practice is kind of slow given a huge enough dataset. In this project, we reduced the validation set size short the prediction time, but this solution may undermine the comparison of models because of the differences of validation set size. Considering some other models which also have good performance but faster than collaborative filtering may be a good idea.
2.In reality, the non-purchase pairs will be comparatively larger than purchased pairs, so the metrics of using accuracy may not be a good idea in practice, we should consider assign additional weight to negative instances, for example, F_1 score [6].

REFERENCES

[1] http://infolab.stanford.edu/~ullman/mmds/ch9.pdf
[2]http://www.cs.toronto.edu/~kswersky/wp-content/uploads/svm_vs_lr.pdf
[3]https://github.com/scikit-learn/scikit-learn/blob/51a765a/sklearn/linear_model/logistic.py
[4]http://scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.DictVectorizer.html
[5]https://en.wikipedia.org/wiki/Jaccard_index
[6]https://en.wikipedia.org/wiki/F1_score
[7]GithubRepo:https://github.com/think-data/ml-nanodegree/tree/master/capstone