Music Recommendation System on Digital Music Dataset

Bowei Feng
Cong Deng
Department of Computer Science, Rutgers University
{bowei.feng,cong.deng}@rutgers.edu

1 INTRODUCTION

Recommendation system has been widely used in many fields. It can be seen as a platform or engine which predicts user preferences and can recommend items to a user based on them. recommendation system can be used as playlist generators for music or videos in YouTube or Netflix. It can achieve product recommendation in Amazon or eBay. And it can also provide news recommendation in Bloomberg.

The problem this project tries to solve is a music rating prediction problem. The dataset we have has some ratings of some users to some music, However, there are also many vacancies. Our goal is to predict these vacancies and recommend items to users based on these predictions.

Many models can be used in recommendation system. We decide to use user-based collaborative filtering.

The experimental results will be described below in details.

2 RELATED WORK

Here are three types of recommendation system that are frequently used.

2.1 Content-Based Filtering

The Content-Based Recommendation relies on the similarity of the items being recommended. The basic idea is that if you like an item, then you will also like a similar item.The concepts of Term Frequency (TF) and Inverse Document Frequency (IDF) are used to pick important features such as author/actor/director to create an item profile for each music. Then, the user profile vectors are also created based on his actions on previous attributes of items and the similarity between an item and a user is also determined in a similar way.

2.2 Collaborative Filtering

The Collaborative Filtering Recommendation[1] is entirely based on the past behavior and not on the context. More specifically, it is based on the similarity in preferences, tastes and choices of two users. It analyzes how the tastes of one user are similar to another and makes recommendations based on that. In User-User Collaborative Filtering, we find look-alike users based on similarity and recommend musics which the first user has chosen in the past. Item-Item Collaborative Filtering is quite similar to previous algorithm, but instead of finding similar users, we try to find similar music.

2.3 Matrix Factorization

In the previous attempt, rating matrices may be overfitted to noisy representations of user tastes and preferences. In addition to this, it doesn't fit scale well to extremely large datasets because it needs a lot of computations to make recommendations. So, we can apply Dimensionality Reduction techniques to derive the tastes and preferences from the raw data, which is also known as low-rank matrix factorization.

3 PRELIMINARIES

The dataset we use is from Amazon product data. The datasets have dataset for different type of product. Based on our hardware infrastructure and computing resources, we choose the 5-core dataset. This dataset contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014. We choose a digital music data with over 64000 ratings, which is applied to 3568 music by 5541 users. There are several parameters for each record for the dataset, e.g., reviewerID, asin, reviewerName, reviewText, overall, summary and reviewTime, and we only focus on reviewerID, asin and overall, they are userID, productID and rating respectively.

The dataset needs to be separated into two parts — training set and test set. The training set contains 80% of the data. The test set contains 20% of the data. Two ways are used in our project to split the data. We write splitData() to directly split the data into 2 parts. We also choose some of users and some of their ratings to get train/test dataset. Training set is used to train the model and test set is used to assess whether the model is good or not.

4 PROBLEM FORMALIZATION

We can formalize the problem as the following way.

$$U : set\ of\ users$$
$$M : set\ of\ music$$
$$Utility\ function\ u : X * I \rightarrow R$$
$$R : set\ of\ ratings$$
$$A\ sample\ Utility\ Matrix\ can\ be\ presented\ as :$$

	user1	user2	user3
music1	1		5
music2		3	5
music3	4		

There are several steps to solve the problem:

- Create training and validation dataset by splitting the original dataset (the rating in the validation dataset are treated as unknown).
- Predict unknown ratings from the known ones - develop user- based CF model to conduct rating prediction.
- Evaluate predictions by calculating the MAE and RMSE
  MAE: Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Actual_i - Prediction_i|$$

RMSE: Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Actual_i - Prediction_i)^2}$$

- Predict the ratings on all the items that user did not buy before, then rank the items in descending order of the predicted rating, and finally take the top 10 items as the recommendation list.
- Evaluate the quality of the recommendation list by calculating precision, recall, F-measure and NDCG. The formulas of these criteria are shown below:

$$Precision = \frac{prediction\_truth}{all\_prediction}$$
$$Recall = \frac{prediction\_truth}{all\_truth}$$
$$F\_Measure = \frac{2*Precision*Recall}{Precision + Recall}$$
$$NDCG[2] = \frac{DCG}{IDCG}$$

5 THE PROPOSED MODEL

Our model is based on user-based collaborative filtering.

The main idea is to find set N of other users whose ratings are similar to x's ratings and create the recommendation list based on them.

Firstly, we weight the similarity between each two users. Here we combine Jaccard similarity measure and Euclidean distance to calculate the similarity.

$$d(x,y) = \sqrt{\sum(x_i - yI)^2}$$
$$W_{x,y} = \frac{|N(x)\bigcap N(y)|}{|N(x)\bigcup N(y)|}$$
$$sim(x,y) = d(x,y)*W_{x,y}$$

Next, we move forward to implement prediction for music i of user x. Let N be the set of K users most similar to x who have rated music i.

$$r_{xi} = \frac{\sum_{y\in N}sim(x,y)*r_{yi}}{\sum_{y\in N}sim(x,y)}$$

At this time, we could recommend music for each user.

Algorithm 1: User-based Collaborative Filtering

```
1  res = {}
2  for each u ∈ users do
3      Poss_music = set();
4      rank = {};
5      N = the set of top K similar users of u;
6      for each v ∈ N do
7          v_bought = music that u has bought;
8          for each music ∈ v_bought do
9              if u has not bought music then
10                 Poss_music.add(music);
11             end
12         end
13     end
14     for each music ∈ Poss_music do
15         rate = predict(u.music);
16         rank.setdefault(music,0);
17         rank[music] = rate
18     end
19     res[u] = sorted(rank.items())[0:10]
20 end
```

6 EXPERIMENTS

We predict the ratings in the testing set as if we don't know them. Then we compare the true ratings with the prediction. The result is shown below:

$$MAE = 0.7763$$
$$RMSE = 1.1331$$
$$Precision = 0.0512$$
$$Recall = 0.0315$$
$$F\_Measure = 0.0386$$
$$NDCG = 0.2143$$

7 CONCLUSIONS AND FUTURE WORK

In this project, we use user-based Collaborative Filtering to do the prediction. We calculate many criteria to assess whether the model is good or not. We think the most appropriate criterion is MAE and RMSE. Purchasing a certain digital music is a random event. One user can have a certain flavour, but he can't but all music under this genre. It is quite common that one user never buy music A even if he bought music B which is similar to A. There are often dozens of music similar to music B. The user may buy only one or two of them if he is not a music fanatics. That is why precision and recall are very small in the experiments. Making a recommendation list for each user and assessing each model based on it is not a good idea. There is too much randomness in it. MAE and RMSE is more appropriate and stable to use as a criterion.

Here are some pros and cons for our model. Actually, it is easy to implement and content independent. However, the percentage of people who rate items is really low, new users will have no to little information about them to be compared with other users, and

new items will lack of ratings to create a solid ranking. Furthermore, Better recommendation requires more users as well as more computing power and time.

We use several hours to train the model on our laptop. Due to the limitation of our computing resources, we can only choose small dataset. I think this is why our precision and recall are very small.

If we have access to high performance computing clusters in the future, we can use larger dataset(tens of million) and more complex model above to build our music recommendation system.

REFERENCES

[1] Zhi-Dan Zhao and Ming-Sheng Shang. User-based collaborative-filtering recommendation algorithms on hadoop. In *2010 third international conference on knowledge discovery and data mining*, pages 478–481. IEEE, 2010.
[2] Consistent Distinguishability. A theoretical analysis of normalized discounted cumulative gain (ndcg) ranking measures. 2013.