How to Build a Recommender System

Junhua Chen
CS229: Machine Learning

Abstract

- The underlying goal of the recommendation system is to personalize content and identify relevant data for the audiences.
- Given a user's past records, a ranked list of recommended items for that user is generated.
- We model the problem as a binary classification problem, where we learn a function to predict whether a user also likes other items.
- Four different models were studied, and their performances were measured and compared. The model called **Deep Learning to Collaborative Filtering Model** performed best among them.

Motivation

- Recommendation systems help users discover items such as songs, movies or books from the huge number of available resources on the web or other electronic sources.
- The need for personalization or interest has led to the development of systems that adapt themselves by changing their behavior based on the inferred characteristics of the user interacting with them.
- In this project, we will be building a recommendation system by using the past record of users which can helps the users discover their interest for the future.

Evaluation Sequence

We evaluated the following four models for our recommendation system from baseline to more advanced models:

- Popularity Based Recommender Model
- Item Similarity Based Collaborative Filtering model
- Matrix Factorization Based Model
- Deep Learning to Collaborative Filtering Model

Model Introduction

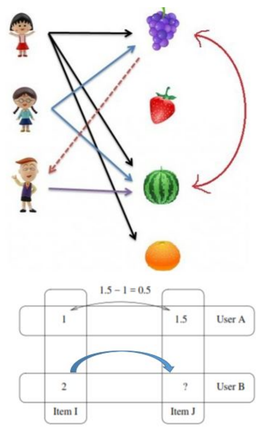

Figure 1:Nearest neighbor collaborative filtering

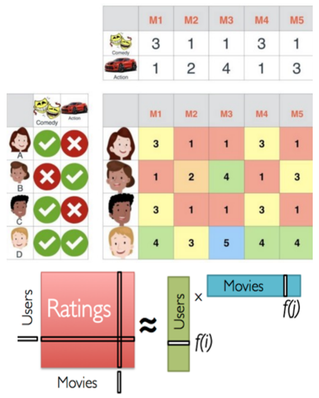

Figure 2:Factorize a matrix to discover latent features

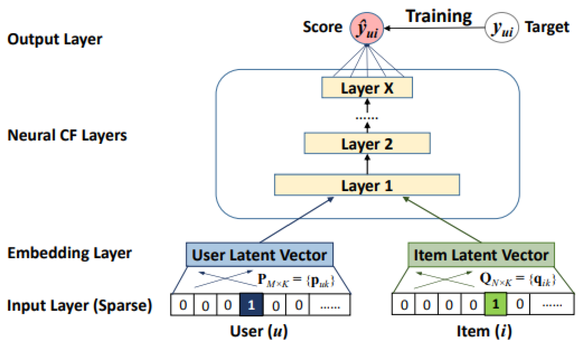

Figure 3:The diagram for Neural Collaborative

Experiment Result

user_id	song	score	rank
46b7dec	She Just Likes To Fight - Four Tet	0.281579	1
46b7dec	Warning Sign - Coldplay	0.281579	2
46b7dec	We Never Change - Coldplay	0.281579	3
46b7dec	Puppetmad - Puppetmastaz	0.281579	4
46b7dec	God Put A Smile Upon Your Face - Coldplay	0.281579	5

Figure 4:Recommendation result for a user "46b7dec" by using item similarity based model

Input

ID	$Song0$	$Song1$	$Song2$	$Song3$
$User0$	5	3	0	1
$User1$	4	0	0	1
$User2$	1	1	0	5
$User3$	1	0	0	4
$User4$	0	1	5	4

Output

ID	$Song0$	$Song1$	$Song2$	$Song3$
$User0$	4.97	2.98	2.18	0.98
$User1$	3.97	2.40	1.97	0.99
$User2$	1.02	0.93	5.32	4.93
$User3$	1.00	0.85	4.59	3.93
$User4$	1.36	1.07	4.89	4.12

Figure 5:Demonstration of the matrix factorization recommendation model

In order to check if the model is working or not, we use Hit Ratio (HR) to evaluate the performance for the recommender systems.

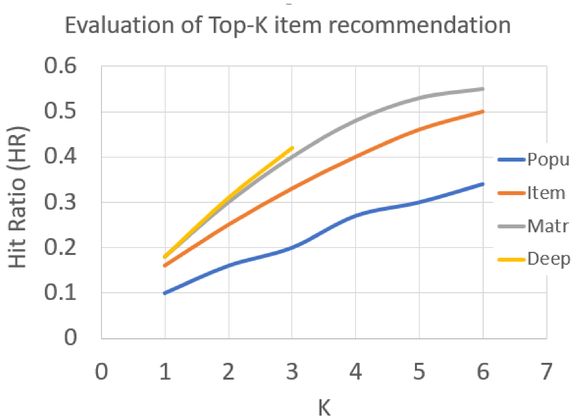

Figure 6:Evaluation of Top-K item recommendation

Conclusion

- Popularity Based Model: Naive model with lowest HR
- Item Similarity Based Model: Linear Algebra with lower HR
- Matrix Factorization Model: Good HR but need large number of latent factor K
- Deep Learning Model: Best HR without limitation from MF

Future Work

- Optimize parameter of deep learning to further improve HR
- Try Normalized Discounted Cumulative Gain to check the trend is consistent
- Apply the method on the industry level yield and error analysis

Reference

- Coverage directed testbench automation
- Music Genre Classification using Machine Learning Techniques
- Finding the genre of a song with Deep Learning
- Deep predictive coverage collection
- Recommender System Using Python

Video Link

https://1drv.ms/u/s!ArgNcBVbSyOuzHobPItEjD1Ogsrx