Hybrid Modelling for Recommender Systems

Anup Jaipraskash Bolli
2018/2019
Dublin City University
Dublin, Ireland
Anup.bolli2@mail.dcu.ie

***Abstract*—with tremendous data being generated day-today basis, it is impossible for the users to go through all the content, and make suitable choices based from this huge data. Recommender Systems(RS) makes life easier by proposing most suitable item which might interest the users using various Machine Learning algorithms based on the user's historical behavior and contextual analysis. Traditional approaches for recommending systems include Content based filtering and Collaborative filtering. But these personalized systems limit the options by displaying only the content liked by user, overtime they homogenize the users interest, making users to lose interest into other topics, which indirectly reduces options for recommendation system. To make this worse User might have to rate low to the recommendation to get rid of them. In the following paper I would like to propose a hybrid approach to recommendation systems to diversify recommendation.**

Keywords—Recommender Systems, Neural network, Collaborative Filtering, Content Filtering, Bandit, Recurrent Neural Networks.

I. INTRODUCTION

The broadening of Internet of things and the boom in web development generates almost 2.5 quintillion bytes of information each day with 90% of data being generated in last two years. This massive data being available on the internet generates problems of information overload, which led to incorrect decisions. Hence Recommender systems came into development. Recommendation Systems plays a key driver of demand for the major E-Commerce sites like Facebook, Amazon, and Netflix etc. Improving customer experience and trying to make experience more personalized is the paramount for all the business.

RS tends to solve this information overload problem by filtering items based on user's interest and prosing only relevant information to the users. These RS leverages the shopping/listening/watching/ activities pattern and predict an option that user would like. The most basic recommender systems are modeled on collaborative filtering which assumes that people like things similar to the other things they like, and things liked by other people with similar interests. But the issue with these models is Filter Bubble problem where a user is trapped inside the bubble wherein the only information that personalized algorithm thinks that user would like is passed. Over the time as the system learns more about the user, this bubble becomes smaller and smaller and trapping user into positive feedback loop. For instance, if the user exclusively reads a fantasy book, then RS should be more biased to fantasy titles that do not mean the user is not interested in the other genres of books, here recommendation system should not be dominated by these genres. Suppose if any other genre is dominant in rest of fantasy readers, then poses an issue with Collaborative filtering by recommending user with that genre even if the user is not interested in consuming it.

In Recommender systems, Deep Learning has emerged and shown promising results. In this research project I would be implementing Clustering based algorithm (K-Nearest Neighbors), Deep learning (Recurrent Neural Network) and recommend movies to user based on the training set provided to systems.

The following paper is detailed as follows Section II reviews the related literature regarding general recommendation systems using Generalized Matrix factorization, (GMF) Multilayer Perceptron(MLP) and Fusion of GMF and MLP, Content based models and their efficiency in recommending. Section III outlines the research plan for the practicum. Section IV gives the review and consolidated proposal drawn from the papers.

.

II. RELATED RESEARCH

"Mixture-of-tastes Models for Representing Users with Diverse Interests" by Maciej Kula published on 29 Jan 2018 in arXiv.org, considers that each user has diverse taste and their recommendations by RS should also be diverse which makes this paper more interesting. It pointed out the drawbacks from recent recommendation approaches which consider a user as a unimodal, observed user activities could be a manifestation of several distinct users or tastes, resulting in mean-of-tastes representation [1]. The Author tries to model using the multiple distinct interests of user tastes through a mixture-of-tastes model, leading to marginal increase in recommendation quality. The model proposes and evaluates several different distinct tastes of users as individual taste vector, which is associated with attention vector, representing competency in evaluation at any given time. This approach was applied both on the matrix factorization models and recent recurrent neural networks. According to the stats provided in the paper clearly states mixture-of-interests models outperform single-representation models on standard ranking quality metrics. In case of recurrent neural network, the improvement achieved at very modest cost and making this model a straightforward progression with other existing models.

Stephen Bonner and Flavian Vasile associated with Citreo AI labs from Paris presented paper on "Causal embeddings for Recommendation Systems", published on 3rd Aug 2018 at 12th ACM Conference meeting on Recommended Systems. In this paper, Authors highlight that natural behavior of the user is being modified due to recommendation systems which aim to increase number of sales or time spent on website [2]. This paper tries to bridge a gap between final recommendation objective and classical setup by building new adaption algorithm that learns from the logged data containing results from a biased recommendation algorithm and predicts recommendation on random exposure. The paper further tries to compare state -of-art results from matrix factorization method to new casual recommendations and conclude that new causal recommendation is more effective and provided more better recommendation. Authors were able to leverage sample of biased feedback data and proved that new recommendation policy outperformed existing classical matrix factorization methods, in addition to recent causal methods BanditNet(Neural network) and Inverse propensity Score based factorization methods. This paper provides motivation to improve the sequential product recommendation by extending existing approach to user sequence and leveraging both organic activity sequence and influenced user activity sequence.

Improved version of book recommender systems "A Book Recommendation Algorithm Based on Improved Similarity Calculation" was presented by Yue Li at 2018 3rd International conference on Mechanical, Control and Computer Engineering held on 14-16 Sept 2018 . [3]This paper present personalized version of recommending algorithm based on the combination of collaborative filtering, content filtering and user's current characteristics/user behavior. The Author focused the drawbacks from the recommended approaches, in which recommendations were made based on the user-item similarities, item-item similarities and mixed approaches by calculating user similarity based on the user -commodity score matrix. These approaches neglect the inherent details of the readers like age, gender, educational level, time of the day, day of week, geographical locations will influence in evaluating recommendations for the readers. The Author has divided User characteristic into fixed data (Age, gender, education, attention type, rank) and dynamic data (click to view, search, buy, recommend, evaluate, questioning).To evaluate the recommendation algorithm, author used a balance of F1 measurement indicators of accuracy and recall rate and proved that new improved user similarity algorithm showed significant improvement in recommendation policy and uplifted readers experience.

Fuzzy logic with multi-criteria recommendation technique has been proposed by Mohamed Hamada, Nkiruku Bridget Odu, Mohammed Hassan in the paper [4]"A Fuzzy-Based Approach for Modelling Preferences of Users in Multi-Criteria Recommender Systems" in 2018 IEEE 12th International Symposium on Embedded Multicore/Many-core Systems-on-chip. This paper mainly focuses on the application and Fuzzy logic to the existing successful multi-criteria recommendation system and comparing the performance to those of single rated RS. Fuzzy Logic is prevalent in different domains of study with its significant benefit being that it can be trained with small amount of training data and its ability to combine human heuristics to computer aided decision systems. To evaluate the precision and efficiency of the proposed system, various experiments were carried using Yahoo Movie dataset. Authors modeled two systems, one RS integrated with Asymmetric Singular Value Decomposition (AsySVD) and other Fuzzy based multi criteria RS integrated with AsySVD. The tests were carried out in 10-fold cross validation rule, where 90 percent of the dataset was used to train the system and 1 percent as test data. The outcomes of the experiments proved that Fuzzy based multi criteria RS performed more accurately that their corresponding traditional single rated based RS.

A paper on "Three-way recommendation integrating global and local information" was presented by Yaun-Yaun Ma, Heng-Ru Zhang, Yuan-Yuan Xu, Fan Mei, Lei Gao at 2nd Asian conference of Artifical Intelligence Technology, which tries to solve the issues with the existing matrix factorization approaches that suffer from under-fitting due to use of global information for all users and items [5]. To minimize the errors in recommendation, The Authors put a proposal to build improved Matrix factorization (three-way recommendation) model that integrates local and global information to predict ratings and make better recommendations. The items were divided into three sets positive, negative and boundary items using k-means clustering method, this forms user's local information and item's local information. User's preference and collaborative filtering of items forms a global information unit. By using global and local information, two MF models are built local-user-global-item (LUGI) and Global-user-local-item (GULI). The results of these algorithms are used to predict p rating. After this threshold pair is calculated α and β based on the three-way decision, which is used to make recommendation. After carrying out various experiments on well-known datasets, it was proved that combination of global and local information has effectively improved the quality of recommendation.

III. Approach

A. DataSet:

The proposal of this project is to implement hybrid approach using Popularity, Collaborative filtering and content filtering to make better movie recommendation. The Movie Lens 100K(ML-100K) dataset is being used to test the effectiveness of the recommendation systems, which contains 100,000 ratings (0.5-5) from 943 users on 1682 movies, each user has rated at least 20 movies and it also contains genre of each movie and demographic information such as age, gender and occupation. This dataset also contains time stamp when the user has given rating and external links to IMDB and TMDB id's which was beneficially to extract meta data about the movies. For the content-based recommender meta data of the movies was required for better predictions, So the addition of movie's meta data information from TMDB (The Movie Database) was added to suffice the request.

The TMDB consists of meta data of 45000 movies listed in the Full Movie Lens Dataset. The Dataset consists of movies released on or before July 2017. Data includes cast, crew, genres, keywords, release dates, revenue, budget, overview, production company, languages, TMDB vote count and vote average.

The length of movie overview provided by the TMDB was not sufficient for the content based recommender system, to get the detailed overview of the movie plot, web scrapping has been done from OMDB(Online Movie DataBase). As a part of Web scrapping, overview of a movie is obtained by using external links provided in the ML-100K dataset i.e TMDB id.

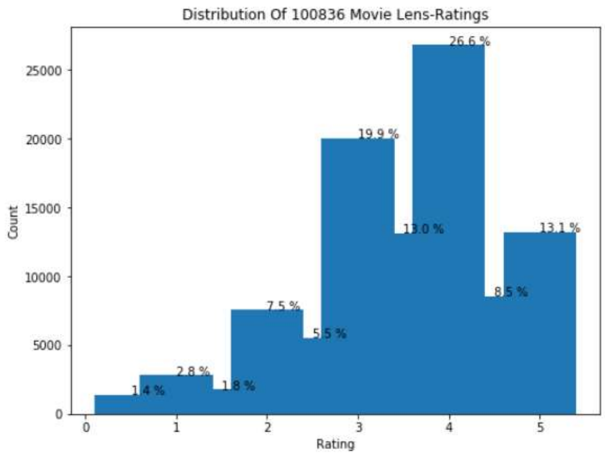

Figure 1:Distribution of ML-100K ratings

The above plot shows the distribution of 100836 Movie-Lens ratings, which shows that most of the movies were given rating 4.

B. Data Preprocessing

Textual data plays an important role in the functioning of content-based recommender engine. Since, the similarity between two movies is determined by content of movies. Keywords, Genres and description were given special treatment and it was extensively used.

The keywords were cleaned using NLTK package in two steps. Initially the number of occurrences of each keyword was calculated, suppressed the keywords which occurred less than 5 times with the synonym of higher frequency keyword. Later suppressed the keywords that appeared less than 3 times.

The description and tagline of the movie data were cleaned by converting them to lower case, removing the unwanted characters and punctuation, removing stop words (NLTK stop words were used as reference),,tokenized the sentence into series of words, later lemmatization was applied to get root words and Term Frequency-Inverse Document frequency for each words was calculated.

The data obtained from OMDB site was originally in Json format, this was converted into CSV file to arrive at an input which can be directly loaded into Pandas DataFrame, Otherwise the data from ML-100K and TMDB was already cleaned.

C. Machine Learning Models

a.) Populairty based Recommendation

This is a simple recommendation system based on TMDB average rating and how many votes a movie received to generate top movies chart for given genre. To makes sure that while generating a top chart, a movie with 5 votes and each 5 star rating should not overcome a movie with 10000 votes and an average of 4.1 star rating, weighted rating was calculated.

The next step is to determine the "m" minimum votes required to be listed in chart. Here 95% percentile was used as cutoff, i.e movie to be feature in top chart requires votes more than at least 95% of the movies in the list.

True Bayesian Estimate formula for Weighted rating is:

WR = (v/(v+m)). R +(m/(m+v)).C

where:

v: number of votes for movie
m: minimum votes required to be listed in chart
R: average rating for the movie
C: mean rating of all movies.

Later on, User profile history was assessed to capture the genres, he liked the most and based on that top charts for each genre was recommended to the user.

b.) Content based Recommendation

Content based recommender is based out of contents of items and description given to that item. The system computes the similarity index between the items and recommends items which are similar to the items liked by user.

1) Content Recommender using Tags and Description:

Movie overview and taglines were used to make recommendation list. After cleaning the data, TF-IDF vectorizer is being used to generate word embeddings for text data and then cosine similarity has been applied to get the similarity scores between the movies. The cosine similarity measures the angle between two vectors projected into multidimensional space, and it is represented as:

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 2:Cosine Similarity between A and B vector

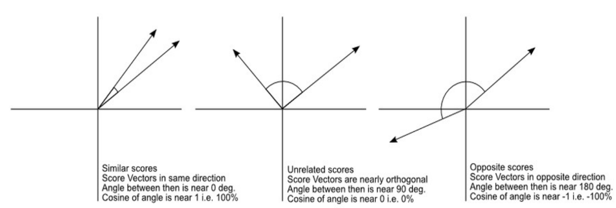

Figure 3: Cosine Similarity values for two different documents,1(same direction),0(90 deg), -1(opposite directions)

Users profile was generated by extracting the recently liked 30 movies and based on this list, 5 similar movies to each movie was used to generate the recommendation list.

2) Content based model using Movie metadata:

The dataset from the above model was combined with Crew, Cast, Genres and Credits. From the Crew, only Director was considered as others don't contribute that much to movie success. From Cast, only top 10 widely known actors were selected, as lesser known actors and their characters don't impact viewers opinion.

While preparing the metadata for Cast, the features were converted to lower case and spaces were stripped, so that two actors with same first name should not be treated equally. Word Embeddings for this text corpus was generated using Count Vectorizer and Cosine Similarity was used to compute similarity scores. Recommendations were made similarly as first model

To improve recommendations, concept of popularity and content-based models was used to generate third model. In this model, top 25 similar movies were calculated as in model two and then movies were ranked based on weighted rating with 60% cut-off.

3) Content based model using user profile and item profile:

In this model, User profile was generated using movie ratings and item profile was built using meta data of the movie such as Tags, Description, Genre, Cast, Crew and credits. As discussed in the above model, data has been cleaned before generating TFIDF matrix.

Once the user profile and item profile are ready, I have used cosine similarity between these two profile to generate recommendation list.

```
improved_recommendations('The Dark Knight')
```

	title	vote_count	vote_average	year	wr
7648	Inception	14075	8	2010	7.917588
8613	Interstellar	11187	8	2014	7.897107
6623	The Prestige	4510	8	2006	7.758148
3381	Memento	4168	8	2000	7.740175
8031	The Dark Knight Rises	9263	7	2012	6.921448
6218	Batman Begins	7511	7	2005	6.904127
1134	Batman Returns	1706	6	1992	5.846862
132	Batman Forever	1529	5	1995	5.054144
9024	Batman v Superman: Dawn of Justice	7189	5	2016	5.013943
1260	Batman & Robin	1447	4	1997	4.287233

Figure 4Sample recommendations for movie 'The Dark Knight'

c.) Collaborative based Filtering

Collaborative filtering is based on the assumption that people like the things that are liked by other people with similar taste, and things similar to other things they like.

1.) Neural Collaborative Filtering:

As Matrix factorization is well known model for rating prediction, its behavior can be easily mimic with the help of Neural Collaborative Filtering.

1.1. Generalised Matrix Factorization (GMF).

In this model, embedding vector resulting from one hot encoding of user vector of is used to represent latent vector of user. Similarly, the embedding vector obtained from one hot encoding of item vector can be used to represent latent vector of item [1]. Let the p_u represent user latent vector and q_i represent item latent vector, hence the mapping function for first Neural CF can eb defined as:

$$\phi(p_u;\ q_i) = p_u \odot q_i,$$

Where $\odot$ denotes element wise product of vectors and this product is projected to output layer:

$$y_{ui} = a_{out}\left(h^T(p_u \odot q_i)\right)$$

Here, a_{out} is an activation function and h^T is edge weights used for output layer, if we use identity activation function and uniform weight vector , then exactly matrix factorization [2] can be formulated.

1.2. Multilayer Perceptron (MLP)

NCF models user and item latent feature vectors into two different path way, but according to multimodal deep learning work [3], it is intuitive to concatenate these two pathways into single model. However, a simple concatenation of user and item vector does not account interaction of user and item latent features, which does not suffice collaborative filtering. To overcome this issue, using MLP a hidden layer has been added over the concatenated vectors, to learn the interaction between user and item latent features.

$$z_1 = \phi_1(p_u; q_i) = \begin{bmatrix} p_u \\ q_i \end{bmatrix};$$

$$\phi_2(z_1) = a_2(W^T_2 z_1 + b_2);$$

$$\ldots\ldots$$

$$\phi_L(z_{L-1}) = a_L(W^T_L z_{L-1} + b_L);$$

$$y_{ui} = \sigma(h^T\ \phi_L(z_{L-1}));$$

Where H_x : Weight Matrix

b_x : Bias Matrix

a_x : Activation function for xth layer perceptron

For activation function of MLP, we have chosen ReLU(Rectified Linear Unit), as it is well suited for sparse matrix and reduces the chances of model getting overfitted. The architecture of the network is designed in the tower pattern, where the bottom layer is the widest layer and each successive layer becomes narrower with smaller number of neurons. The gist of this architecture is, as the tower gets narrower, smaller number of neurons gets to learn more abstractive features of the data.

1.3. Fusion of GMF and MLP

This NCF model, is an ensemble of GMF and MLP Model, Where GMF applies a liner kernel to User-Item latent features and MLP that uses a non-linear kernel to learn the interactions from data.

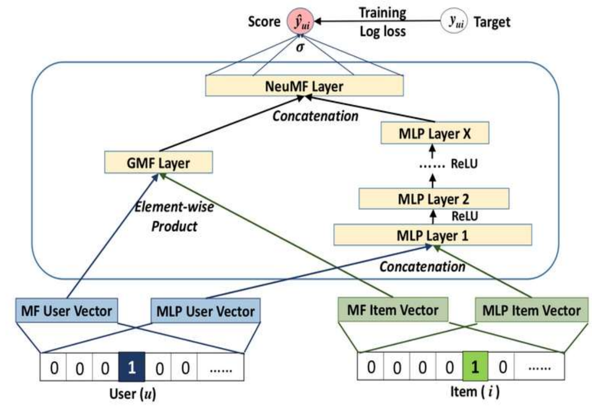

Figure 5: Neural Matrix Factorization Model

The proposed architecture in the paper [1] is:

a) User(u) and item(i) are low dimensional embeddings for User and Item.

b) GMF layer combines the embeddings of user and item latent feature vector using a dot product.

c) MLP layer creates embeddings for user and item latent feature vectors, concatenates them to create embedded feature vector.

d) Neural MF then combines two models by concatenating the final hidden layers of GMF and MLP.

The formulation of the above model is:

$$\phi^{GMF} = p^G_u \odot q^G_i ;$$

$$\phi^{MLP} = a_L(W^T_L(a_{L-1}(\ldots a_2(W^T_2\begin{bmatrix} p^M_u \\ q^M_i \end{bmatrix} + b_2)\ldots)) + b_L);$$

$$y_{ui} = \sigma(h_t\begin{bmatrix} \phi^{GMF} \\ \phi^{MLP} \end{bmatrix});$$

Where: p^G_u and p^M_u represent user embedding for GMF and MLP model and q^G_i and q^M_i denote item embedding for GMF and MLP model, This model combines linearity of GMF and non-linearity of GMF model for modelling user-item latent features. ReLU activation was used function for MLP model.

d.) Hybrid Filtering:

Hybrid filtering is an ensemble method of content-based filtering and collaborative filtering. Initially the best content-based filtering was used to find similar movies based on the user historical movie interactions, these recommendations were further filtered using popularity-based model to get top similar movies. This creates a basic recommendation list.
Further the Collaborative filtering was applied on the movie ratings to get the second list of recommendations. Based on the weighted scores, the movies from basic list from content-based recommender and the movies from second list of collaborative filtering were combined to generate a hybrid list of recommendation.

IV. Experiments and Results

Various experiments were tried to split the data into train and test. Initially, the data was split with random 80 % as train data and 20% test data, results with this dataset splits were good.
For collaborative filtering, ratings are one of the most important explicit feedback from users towards the item. Hence, keeping balance of ratings in train and test was important. As a part of experiment, Data was split into test and train based on held out Users, only those items were filtered which had minimum number of user interaction count. Further the data was split into test and train based on held out users. In this case, I have considered minimum user-item interaction count of 5, which gave filtered dataset of 100836 interactions between 610 Users and 9724 Movies with a sparsity of 1.7%. After filtering the dataset, out of 610 Users 100 users were held out for train, test and validation. The results for collaborative filtering model from this dataset split are:

Model	GMF	MLP	GMF+MLP
RMSE	2.992	2.7474	2.7146

Figure 6:RMSE scores for Split1

To reduce the RMSE values, further a new data split was tried, because in the above data split User profile /history was ignored and each user is considered as a new user. In the new split, Dataset was divided into train and test based on User-Item interactions, which allows our model to learn User's interests from interactions. Initially, dataset was filtered based on movies which had a minimum rating threshold. Later on, this filtered dataset was split based on timestamp, where latest 20% User-Item interactions was considered as test data and rest as training data. After filtering, there were 87051 watching events from 335 users and 9066 movies with the sparsity of 2.865%.The results for collaborative filtering on this new data split are given below.

Model	GMF	MLP	GMF+MLP
RMSE	2.6815	2.6345	2.6313

Figure 7:RMSE scores for Split 2

The User-Item interactions-based data split was considered for popularity and content-based models. In popularity-based model, Each User's interests were learned from training dataset with latest 30 User-Item interactions and unique list of genres were generated for each User. Top 30 charts for each genre was combined to form a recommendation list.

Similarly, for content-based models, User's interest was captured with last 30 user-item interactions and using cosine similarity 5 movies with highest similarity index were listed as recommendations. To evaluate the accuracy of the models, Mean Average Precision and Mean Average Recall were calculated:
Mean Average Precision @K: gives an insight of how relevant the recommendations model from the top k list are,

whereas Mean Average Recall @K: gives a overview of how well the recommender system was able to recall all the items user has rated positively in the test set.

$$P = \frac{\text{\# of our recommendations that are relevant}}{\text{\# of items we recommended}}$$

$$r = \frac{\text{\# of our recommendations that are relevant}}{\text{\# of all the possible relevant items}}$$

Figure 8:Formula for Recommender System Precision and Recall

MAR @k	PM	CF1	CF2	Hybrid
25	0.00015	0.00031	0.00037	0.0195
50	0.00022	0.00037	0.0005	0.0199
75	0.00023	0.00044	0.00061	0.0202
100	0.00025	0.00049	0.00069	0.0204
125	0.00026	0.00051	0.00074	0.0205
150	0.00027	0.00051	0.00083	0.0206

Figure 9:Table for Mean Average Recall@ K top movies

Models	PM	CF1	CF2	Hybrid
Map@K	6.7	9.5	14.432	32.228

Figure 10:Table for Mean Average Recall @k top movies

V. EVALUATIONS

For Collaborative Filtering, Figure 6 and 7 clearly depicts that Fusion of Generalized Matrix Factorization (GMF) and Multilayer Perceptron (MLP) outperforms the individual application of MLP and MLP. The splitting of data into train/test plays a major role for the performance of the model, which is clearly visible from fig 6 and 7. In Split1, Data was split by holding out users, which hides the User-Item historical interactions from model. In Split 2, Models are exposed to historical User-Item interactions which help in making better recommendations. The other important observation is that MLP under performs GMF with minor margin.

In case of Content based Filtering, RMSE could not applied because it makes random recommendation based on similarity scores. To evaluate the effectiveness of content based model , MAP @k and MAR@k were used. Content-based model(CF2) outperforms other CF1 and popularity based model(PM), as CF2 makes excessive usage movie meta data information compared to other models. Hybrid model which is ensemble of best content based model and collaborative model shows better results with better recommendations.

VI. CONCLUSION AND FUTURE WORK

There have been numerous researches in the field of recommendation system from basic matrix factorization methods to deep neural networks, and combination of these algorithms to make better recommendations. In this research, I have tried to explore various architectures of recommender models. Three different instantiations of recommender system that model user-item interaction in sifferent ways : Popularity based model, Content based and collaborative model with neural network flavor. Finally an ensemble of best content based model and collaborative model.

Since movie recommendations are dynamic based on different factors, such as User's characteristics, Demographic information, implicit feedback such as length of the movie watched, sentiments of User. In this research, only User's explicit ratings and movies meta-data was used, which makes recommendations static. Due to limitation of Computing space and power, the entire Movie-Lens 20M was not used for this research purpose. For future work, I would propose to apply these models on entire Movie Lens - 20M dataset.

Further, the movie recommendations can be improved by extending NCF to model auxiliary information, such as user reviews, knowledge bases and temporal signals. This will help in personalization and could be also be extended to group of users, which will help the decision making for social media.

VII. REFERENCES

[1] M. Kula, "Mixture-of-tastes Models for Representing Users with Diverse Interests," *CoRR,* vol. abs/1711.08379, 2017.

[2] S. B. a. F. Vasile, "Causal Embeddings for Recommendation," *CoRR,* vol. abs/1706.07639, 2017.

[3] C. L. H.-b. LI Yue, "A Book Recommendation Algorithm Based on Improved Co-similarity Calculation," *Computer and Modernization,* p. 85, 2019.

[4] H. M, M. Hamada and N. B. Odu, "A Fuzzy-Based Approach for Modelling Preferences of Users in Multi-Criteria Recommender Systems," in *2018 IEEE 12th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*, 2018, pp. 87-94.

[5] Y. Ma, H. Zhang, Y. Xu, F. Min and L. Gao, "Three-way recommendation integrating global and local information," *The Journal of Engineering,* pp. 1397-1401, 2018.

[6] X. He, L. Liao, H. Zhang, L. Nie, X. Hu and T.-S. Chua, "Neural Collaborative Filtering," *arXiv e-prints,* p. arXiv:1708.05031, 2017.

[7] S. Rendle, "Factorization Machines," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, Washington, DC, USA, IEEE Computer Society, 2010, pp. 995-1000.

[8] N. a. S. R. Srivastava, "Multimodal Learning with Deep Boltzmann Machines," *J. Mach. Learn. Res.,* vol. 15, no. January 2014, pp. 2949-2980, 2014.

[9] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Application of Dimensionality Reduction in Recommender System - A Case Study," Minnesota Univ Minneapolis Dept of Computer Science, 2000.