Knowledge Aware Conversational Movie Recommendation System

Jash Gaglani
Rutgers University
jg1700

Janish Parikh
Rutgers University
jrp328

Kunj Mehta
Rutgers University
kcm161

Abstract

A movie recommendation is important in everyone's social life due to its prowess in providing enhanced entertainment. Our life is increasingly being governed by machine-suggested recommendations. Facebook posts, Instagram stories, Spotify songs, Netflix movies – all of them are recommended to us. The rise in popularity of voice-based personal assistants opens up an avenue of using conversational data to make recommendations. In this project, we adopt the data and approach as described in (Volokhin et al., 2021) to further explore this problem space. We use conversational context to infer a user's sentiment about a movie, and use the ratings of "similar" external critics to predict the user's next movie preference. After working through the paper from scratch, we experiment with more sophisticated deep learning matrix factorization techniques like neural collaborative filtering with an aim to attain better results.

1 Introduction

Recommendation engines in general, and movie recommendations in particular have become a staple of our life with over-the-top streaming services constantly recommending us movies to watch. Now, to add to this we have a proliferation of home devices and home assistants like Amazon's Alexa and Apple's Siri, that gives us access to a ton of conversational data. This conversational data can be used to extract information that can then be used to recommend movies to users through these home devices.

(Volokhin et al., 2021) introduce a dataset for recommending movies from conversational data, and publish baseline results for the same. The problem that they solve can be formally stated as: *Given a conversational database between two users (out of one which simulates a conversational personal assistant) with k turns and m movie references, to predict the user's $m + 1^{st}$ movie preference (Volokhin et al., 2021).* This is done by inferring user sentiment towards the $m = 2$ movies from the conversation and then using collaborative filtering and domain adaptation to find ratings of "similar" external critics reviews and use them to make the $m + 1^{st}$ movie preference prediction for the user.

This approach is a first of its kind in that it solely works on conversational data, which in itself is not available freely enough. Traditional approaches to recommending movies that use either of collaborative or content filtering techniques would need extracted information from the conversation data to serve recommendations. By incorporating preferences of other, external users with established preferences (critics), via shared discussed entities, and the user's sentiment towards them, we also address the "cold start" problem that traditional recommendation systems face with new/unknown users.

This is what (Volokhin et al., 2021) does: it extracts the relevant information entities from the well-formed conversation data, cleans it to be in accordance with the problem statement defined above and serves recommendations using collaborative filtering techniques like KNN, SVD and SVDpp. In this project, we replicate the approach in (Volokhin et al., 2021) using the same dataset. After that, we go a step further and apply neural collaborative filtering on the data with an aim to achieve better results. For all of the approaches we also perform hyperparameter tuning.

In the rest of this report, we outline the prior work done in this domain, present the recommendation techniques and evaluation metrics used in (Volokhin et al., 2021), and in our extension of the same, and conclude with the results of our experiments.

2 Related Work

As touched on previously, the popularity of home assistants is increasing. With this, there has been a lot of research in making user experiences with home or voice assistants better. An example of user experience would be recommendations; movie recommendations in particular.

Movie recommendation, traditionally too, has been given a lot of attention. Collaborative filtering (Katarya and Verma, 2017) (He et al., 2019) and content-based filtering (Elahi et al., 2017) are two of the most general categories of approaches used to tackle movie recommendations. Recently, conversational recommendation systems (Dalton et al., 2018) (SenseTechnologies) (Li et al., 2018) have also been explored.

However, making conversational recommendations for new users and specifically, establishing users' preferences through a conversation is first explored by (Volokhin et al., 2021). The two most popular approaches to to movie recommendation: collaborative filtering (CF) model (Katarya and Verma, 2017) and a model that incorporates user reviews (Zhao et al., 2017) are both taken into consideration in this novel recommendation system, which also takes care of the cold start problem as previously mentioned.

3 Method

3.1 Data

We use the *MovieSent* dataset made available by the authors. The dataset contains the following: conversation data with fine-grained user sentiment labels and reviews of critics scraped from Rotten Tomatoes. We clean the available dataset to extract conversations that have at least two movie mentions, so that we can predict the third one. This is the case where $m = 2$.

3.2 Detailed Approach

Encoding: We use pretrained RoBERTa (Liu et al., 2019) to encode the conversations at the sentence level, to obtain one vector per conversation. We also encode the critic reviews and the movie metadata.

Sentiment Estimation: We use Random Forest Regression Trees trained on the *MovieSent* dataset to perform sentiment estimation for movies mentioned in the conversation, with a 90-10 train-test split due to low data as suggested in (Volokhin et al., 2021).

Collaborative Filtering The collaborative filtering module estimates the user's sentiment towards a new unseen movie. This is done by leveraging a large dataset of external critics' ratings and reviews, which include critics similar to the current user, and the data about user sentiment for the movies mentioned in the conversation. The latter are used by converting the sentiments to ratings of the same scale as critics' ratings. 1

The CF techniques used in the collaborative filtering module are: KNN-based item-CF, SVD SVDpp and NCF

Domain Adaptation: Lastly, we apply Gradient Boosted Trees to perform domain adaptation. This model uses 3 inputs:
(1) CF predictions for the unseen movie;
(2) similarity between the user and critics calculated using earthmovers distance;
(3) similarity between the conversation and movies' metadata calculated by cosine similarity.

The collaborative filtering techniques used in the CF module are explained in detail below.

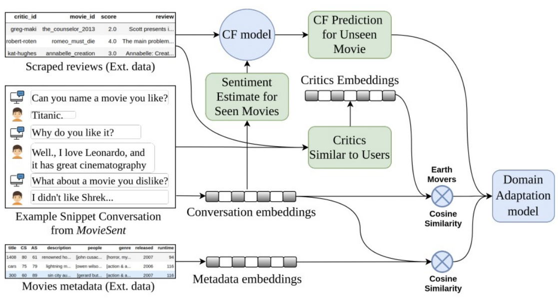

Figure 1: Overview of the recommendation system for conversational elicitation and prediction of movie preferences.

3.3 CF Approaches

3.3.1 KNN

For a baseline model we experimented with different KNN models namely:

KNNBaseline: A basic collaborative filtering algorithm taking into account a baseline rating. This is what (Volokhin et al., 2021) use.
KNNwithMeans: A basic collaborative filtering algorithm similar to KNNBasic which takes into account the mean ratings of each user.
KNNWithZScore: A basic collaborative filtering algorithm similar to KNNBasic which takes into account the z-score normalization of each user.x

3.3.2 SVD and SVDpp

SVD algorithm is based on matrix factorization (Takacs et al., 2008). When we use SVD, it takes the lower dimensional representation of movies such that people who like similar movies together are mapped together. In doing so, it discovers the latent factors that allow movies to be mapped onto the same space as a user. 2

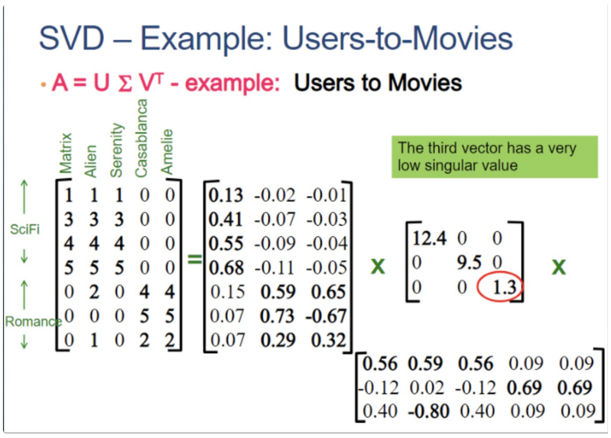

Figure 2: SVD in Movie Recommendation

SVDpp is an extension of SVD which also takes into account implicit ratings. An implicit rating describes the fact that a user rated an item, regardless of the rating value. SVDpp is the best performing CF algorithm from the three that we use.

The equation for the same is:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right)$$

3.3.3 Neural Collaborative Filtering

In Matrix Factorization, like SVD or SVDpp we use a simple and fixed inner product . However, this Matrix Factorization methods are not effective when it comes to estimating complex user-item interactions in the low-dimensional latent space. Neural Collaborative Filtering (NCF) overcomes this limitation of Matrix Factorization by replacing the user-item inner product most used in matrix factorization techniques to learn user-item interactions with a neural architecture.

Neural Collaborative Filtering (NCF) replaces the user-item inner product most commonly used in matrix factorization techniques to learn user-item interactions with a neural architecture.

Specifically, NCF ensembles Generalized Matrix Factorization (GMF) and Multi-Layer Perceptron (MLP) to unify the strengths of linearity of matrix factorization and non-linearity of MLP for modelling the user–item latent structures (He et al., 2017) 3.

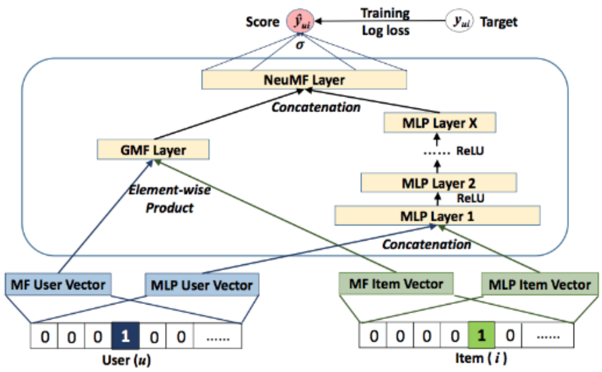

Figure 3: Neural Matrix Factorization Model

The way the ensembling is done is: Generalized Matrix Factorisation (GMF) and Multi-Layer Perceptron (MLP) have separate user and item embeddings. This ensures that both of them learn optimal embeddings independently. GMF then replicates the vanilla matrix factorization by element-wise product of the user-item vector; MLP takes the concatenation of user-item latent vectors as input. The outputs of GMF and MLP are concatenated in the final NeuMF(Neural Matrix Factorisation) layer.

3.4 GBRT

To integrate the results obtained from multiple weak-learners and for domain adaptation, we use GBRT to combine predictions of all the above models into one single hybrid recommendation system using Gradient Boosted Regression Trees.

4 Experiments

4.1 Evaluation Metrics

Mean Absolute Error measures the average magnitude of the errors in a set of predictions, without considering their direction. The MAE is a linear score which means that all the individual differences are weighted equally in the average. It tells us how big of an error we can expect from the prediction model on average. The MAE will always be lesser than or equal to RMSE. If all the errors have equal magnitude, then MAE = RMSE. The lower the value of MAE, the higher is the accuracy of the prediction model.

$$\frac{1}{n}|\sum_{i=1}^{n}(y_i - x_i)|$$

Root Mean Square Error is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data, that is, how close the observed data points are to the model's predicted values. It is a frequently used measure of the differences between values predicted by a model or an estimator and the values observed. The smaller an RMSE value, the closer the predicted and observed values are or the closer you are to finding the line of best fit.

$$(\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2)^{\frac{1}{2}}$$

4.2 Results

We perform hypterparameter tuning on the number of regressor trees for the final prediction of the $m + 1$ movie. We do this for all the CF models mentioned in Section 3.3. We plot the results of tuning the GBRT model estimators for SVDpp 4. We also plot the feature importances with respect to the movie metadata features for the best performing model of SVDpp 5. In comparison with the official results in the paper for GBRT which were RMSE = 1.09 and MAE = 0.84, we obtained a performance of RMSE = 1.07 and MAE = 0.81.

We also experimented with the number of estimators of Random Forest used in Sentiment Prediction and were able to acheive an RMSE of 0.686.

We experiment with Neural Collaborative Filtering as the CF method as an extension to (Volokhin et al., 2021). We also perform hyperparameter tuning on this, the plot for which can be seen in 6.

We also log the time it takes to run one epoch for each of these methods. For SVDpp, it takes 4 and a half hours per epoch and for NCF, it takes 4 hours. Additionally, we also notice that increasing the number of epochs in SVDpp has no significant effect to the performance.

The detailed results for other techniques are listed in Table 1.

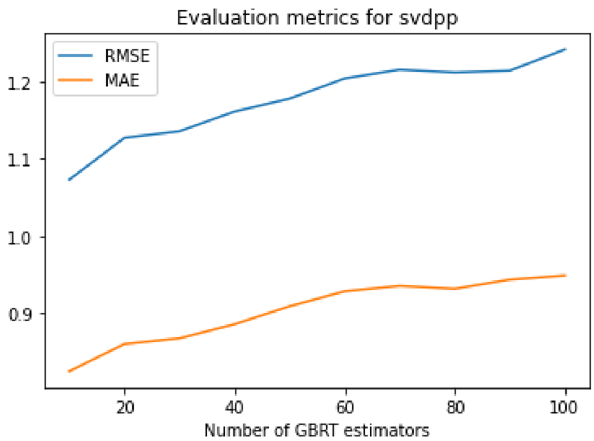

Figure 4: Hyperparameter Tuning for SVDpp + GBRT

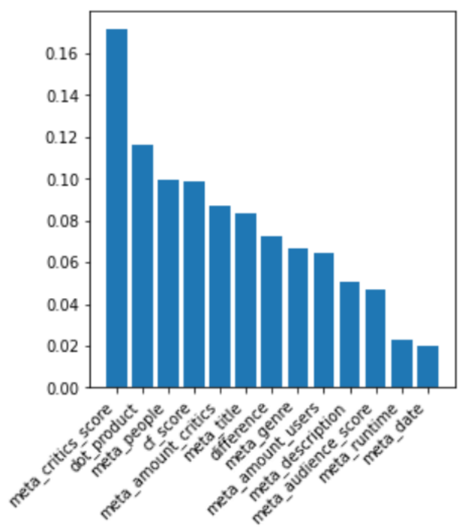

Figure 5: SVDpp Feature Importance for GBRT Estimators = 10

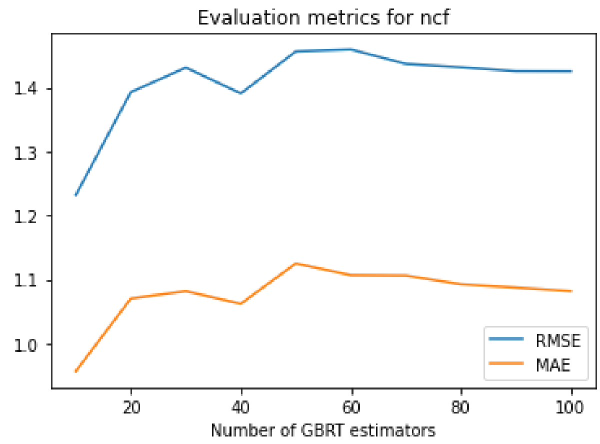

Figure 6: Hyperparameter Tuning for NCF + GBRT

We observed that for all the CF-Methods, we obtained better results in terms of RMSE and MAE when we set the number of estimators as low as 10. Also, we observed that meta critics score had the highest feature importance in the final Domain Adaptation GBRT model and movie metadata as release date and runtime had the least feature importance.

We observed that NCF didn't give us good results or wasn't able to beat SVDpp. We believe that the reason this happens is because the dataset size is small. There are only 500 odd conversation that satisfy the conditions mentioned above ($m = 2$).

Table 1: Quantitative Evaluation of different CF methods, with GBRT estimators = 10

CF-Method	RMSE	MAE
KNNBaseline	1.1118	0.8524
KNNwithMeans	1.1131	0.8527
KNNwithZScore	1.101	0.8383
SVD	1.1318	0.8301
SVDpp	1.07	0.81
NeuMf	1.232	0.9569

5 Code Implementation

The code for this project can be found here.

6 Conclusion

We explored the problem of movie recommendation from conversational data using domain adaptation using the approach described in (Volokhin et al., 2021) and successfully implemented the same from scratch, for all the methods described in the paper. We also performed hyperparameter tuning for all the components in the pipeline and determined better parameters that lead to an improvement in performance of the recommendation system, arguably due to overfitting of the one configured in (Volokhin et al., 2021). Going further, we extend on the paper by applying neural collaborative filtering on the given data and report results of **RMSE = 1.232 and MAE = 0.9569**

This is an exciting problem domain with lots of future scope of work. One specific task would be look at more such conversational datasets with huge amounts of data that would give better results with NCF. Also we can explore the tradeoff between the value of m, the number of movies the person engages in conversation about before the agent recommends a movie, given more rich dataset.

Acknowledgement

We are immensely grateful to Karl Stratos (Professor) and Vincent Zhang(Teaching Assistant) for their comments on an earlier version of the proposal, although any errors are our own and should not tarnish the reputations of these esteemed persons.

References

Jeffrey Dalton, Victor Ajayi, and Richard Main. 2018. Vote goat: Conversational movie recommendation. *SIGIR, SIGIR '18, pages 1285–1288, New York, NY, USA. ACM.*

Mehdi Elahi, Yashar Deldjoo, Farshad Bakhshandegan Moghaddam, Leonardo Cella, Stefano Cereda, and Paolo Cremonesi. 2017. Exploring the semantic gap for movie recommendations. *RecSys*, pages 326–330.

Ming He, Bo Wang, and Xiangkun Du. 2019. Hi2rec: Exploring knowledge in heterogeneous information for movie recommendation. *IEEE Access*, pages 30276–30284.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. *CoRR*, abs/1708.05031.

Rahul Katarya and Om Prakash Verma. 2017. An effective collaborative movie recommender system with cuckoo search. *Egyptian Informatics Journal, 18(2):105–112.*

Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. *CoRR*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

SenseTechnologies. AndChill. andchill.io.

G. Takacs, I. Pilaszy, B. Nemeth, and D. Tikk. 2008. Investigation of Various Matrix Factorization Methods for Large Recommender Systems. *Proc. ICDM*.

S. Volokhin, J. Ho, O. Rokhlenko, and E. Agichtein. 2021. You sound like someone who watches drama movies: Towards predicting movie preferences from conversational interactions. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*.

Zhou Zhao, Qifan Yang, Hanqing Lu, Tim Weninger, Deng Cai, Xiaofei He, and Yueting Zhuang. 2017. Social-aware movie recommendation via multimodal network learning. *IEEE Transactions on Multimedia, 20(2):430–440*.