An Analysis of Recommendation Methods in Movie Recommendations

A Technical Report
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Nikita Semichev

May 14, 2021

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Nikita Semichev

Technical advisor: Hongning Wang, Department of Computer Science

An Analysis of Recommendation Methods in Movie Recommendations

Introduction

State-of-the-art methods for recommendation systems are constantly being developed and improved. Analyzing different implementations and testing them on real data provides insight into how those algorithms learn from the data and specific factors that determine what gets recommended. This project uses a dataset of movies from MovieLens to compare different recommendation algorithms using different metrics. It is important to understand how different recommendation algorithms work to determine what works best and what could be improved.

Background Research

There is a wide range of different recommendation algorithms, however, this project focused on analyzing the performance of 7 algorithms that utilize neural recommendation approaches. This project tries to extend the work from a previous study[1], which specifically focused on the same 7 algorithms, using the same implementations and hyper-parameters for training, however, instead of using different datasets, this project uses a single dataset for all of the methods. The description of these 7 algorithms and important hyperparameters and their values for each algorithm are as follows (hyperparameters are shown in the tables for each method):

1. Collaborative Memory Networks (CMN)

Collaborative memory networks combine memory networks and neural attention mechanisms with latent factor and neighborhood models, and maintain three memory states: user-specific memory, item-specific memory, and a neighborhood state.[2]

epochs	50
epochs gmf	100
hops	3
neg samples	4
reg l2 cmn	1.00E-01
reg l2 gmf	1.00E-04
pretrain	True
learning rate	1.00E-03
verbose	False
batch size	128
embed size	50

2. Metapath based Context for RECommendation (MCRec)

In addition to learning the representations for users and items, a common practice in existing heterogeneous information networks (HIN), this model incorporates meta-paths as the context in an interaction between a user and an item, which allows for learning a better interaction function and can result in better recommendations.[3]

epochs	130
latent dim	128
reg latent	0
layers	[512, 256, 128, 64]
reg layes	[0, 0, 0, 0]
learning rate	1.00E-03
batch size	256
num negatives	4

3. Collaborative Variational Autoencoder (CVAE)

The collaborative variational autoencoder is a generative latent variable model, where the contents and ratings of the items are generated through latent item and content variables, which allows the model to learn deep latent representations and implicit relationships between items and users.[4] The hyperparameters table contains two columns of values, the first is for sparse setting (P=1), and the second is for dense setting (P=10).

epochs	5	35
learning rate vae	1.00E-02	1.00E-02
learning rate cvae	1.00E-03	1.00E-03
num factors	50	50
dimensions vae	[200, 100]	[200, 100]
epochs vae	[50, 50]	[50, 50]
batch size	128	128
lambda u	1.00E-01	1.00E-01
lambda v	10	10
lambda r	1	1
a	1	1
b	0.0100	0.0100
M	300	300

4. Collaborative Deep Learning (CDL)

Collaborative deep learning is a probabilistic feed-forward model, which applies deep learning techniques to learn a deep representation of content and collaborative information, and, since it is a hierarchical Bayesian model, it can be extended to use other auxiliary information to improve its

para lv	10	10
para lu	1	1
para ln	1000.0000	1000.0000
batch size	128	128
epoch sdae	200	200
epoch dae	200	200

performance.[5] Likewise for CDL, the first column of values corresponds to sparse setting, and the second column corresponds to dense setting.

5. Neural Collaborative Filtering (NCF)

Neural collaborative filtering generalizes matrix factorization by employing a neural architecture that can learn an arbitrary functions from the data, and the model in question, neural matrix factorization (NeuMF) combined two different layers into one, the generalized matrix factorization layer and the multi-layer perceptron layer.[6]

epochs	10
epochs gmf	10
epochs mlp	10
batch size	256
num factors	64
layers	[256, 128, 64]
reg mf	0.00E+00
reg layers	[0, 0, 0]
num negatives	4
learning rate	1.00E-03
learning rate pretrain	1.00E-03
learner	sgd
learner pretrain	adam
pretrain	True

6. Spectral Collaborative Filtering (SpectralCF)

Spectral collaborative filtering is the first collaborative filtering based method which directly learns from the spectral domains of user-item bipartite graphs, in which both proximity and connectivity information are revealed in the graphs.[7]

batch size	2048
embedding size	4
decay	3.06E-02
learning rate	8.83E-04
k	2
epochs	805

7. Variational Autoencoders for Collaborative Filtering (Mult-VAE)

A non-linear probabilistic model that utilizes variational autoencoders to extend their use to collaborative filtering for implicit feedback, which uses multinomial likelihood and estimates parameters through Bayesian inference.[8]

epochs	95
batch size	500
total anneal steps	200000

Presented next are the general advantages and disadvantages of the methods that allow for understanding why some methods may perform better than others if applied to different datasets, however, these are just guidelines and results may vary depending on the dataset. The

CMN method is easy to use and add new users, and works reasonably well on smaller datasets, however, the performance can become worse if the datasets are sparse. The MCRec method incorporates a co-attention mechanism, which takes the interaction relation into consideration, which not only can result in good performance, but also makes the recommendations interpretable. The CVAE method is more robust than other methods, such as the CDL, in making accurate recommendations. The CDL method adjusts reasonably well to a change in preferences from a user, however, sometimes changes in what is recommended in cases like this may not always be accurate. The NeuMF is a more complex model, which achieves decent results through deep learning, however, as it is more complex, it may be more difficult to use and properly structure to achieve the most accurate results. The SpectralCF method allows for discovering deep connections between users and items which can mitigate the cold-start problem for collaborative filtering, which is important for recommendation systems. Finally, the Mult-VAE method works well and is more robust than some other methods with sparse data, and, as it uses multinomial likelihood, can provide more accurate recommendation results than other models.

Methods and Results

The dataset that was used for this project is the MovieLens 100K dataset, which contains 100,000 ratings from 1000 users on 1700 movies. The implementations for all the algorithms mentioned previously all come from the authors of each respective algorithm. Using the MovieLens dataset, each algorithm was trained and evaluated using 3 different evaluation metrics, hit rate (HR), normalized discounted cumulative gain (NDCG), and recall (REC). The HR metric calculates the number of correct recommendations a user received. The NDCG metric takes into account the ranking of the correct recommendations, and is a ratio of discounted cumulative gain (DCG) and ideal discounted cumulative gain (IDCG). The REC metric measures the quota of correct recommendations over the true positive samples in the test data. For each of the methods, 80/20 split was used to separate the data into the training and testing sets. The hyperparameters for each of the methods were obtained from the results of the paper that compared the performance of many different recommendation algorithms and recorded the optimal values for each of the hyperparameters, which can be seen in the previous section.[9] The results are shown in the table below.

Method	**HR@10**	**NDCG@10**	**REC@100**
CMN	0.7940	0.5265	0.2517
MCRec	0.6141	0.2271	0.2386
CVAE	0.7632	0.4114	0.1528
CDL	0.5326	0.1527	0.1056
NeuMF	0.7153	0.3984	0.1920
SpectralCF	0.6290	0.1843	0.3143
Mult-VAE	0.6943	0.2481	0.5613

For the dataset this project used, the CDL had the worst performance, most likely due to the nature of the dataset, as CDL achieves better results in sparse datasets. MCRec and SpectralCF performed similarly and both were better than CDL. The MCRec had better performance as it is a meta-path based model that used information such as movie genres to achieve better results. SpectralCF was able to discover deep connections between users and graphs from the spectral domain, which allows this method to achieve better performance using the connectivity information. The next methods that performed better but similarly were NeuMF and Mult-VAE. NeuMF utilizes neural architecture that can learn an arbitrary function from the data, and achieved a much higher NDCG value than Mult-VAE, however, Mult-VAE achieved a much higher REC value. Mult-VAE uses a multinomial likelihood function and introduces a new regularization parameter, which seems to improve the REC metric by a significant amount, prioritizing the quota of correct recommendation (REC) and not the ranking of the correct recommendations (NDCG). CVAE is an extension of variational autoencoders, just as Mult-VAE, however, CVAE uses content data such as text in addition to just rating data. For this dataset, the CVAE performed better than most other methods in HR and NDCG metrics since movie genres and tags provide extra information and allow CVAE to achieve better and more accurate results. CMN was the best model in terms of HR and NDCG, which can be explained by the small size of the dataset, as CMN tends to perform better on smaller datasets, as well as the concept of memory layers or hops, which allows CMN to reassess recommendations based on most similar users between hops before producing the final result, which can be particularly useful for movie recommendations, as it can extract the information about the genres, which are most likely similar between similar users.

Conclusion

There are many different algorithms and methods for recommendations that are being used and developed. This project only looks at a small subset of those algorithms and provides their evaluations. Different algorithms could be studied and evaluated to compare the performance and determine which ones work the best in specific situations. Since only one data set was used in this project, using the same algorithms on different and potentially larger datasets could improve the understanding of algorithms' performance. In addition, more metrics could be used to look at the performance in different perspectives.

References

[1] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019 b. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. *Proceedings of RecSys '19*.
[2] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. In *Proceedings SIGIR '18*. 515–524
[3] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging metapath based context for top-n recommendation with a neural co-attention model. In *Proceedings KDD '18*. 1531–1540.
[4] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings KDD '17.* 305–314
[5] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings KDD '15*. 1235–1244
[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings WWW '17*. 173–182.
[7] Lei Zheng, Chun-Ta Lu, Fei Jiang, Jiawei Zhang, and Philip S. Yu. 2018. Spectral Collaborative Filtering. In *Proceedings RecSys '18*. 311–319.
[8] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings WWW '18*. 689–698
[9] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2019. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Algorithms Research. *Transactions on Information Systems*.