ISSN:0975-9646

Survey of Deep Learning Based Entertainment Oriented Recommendation Systems

Rohit Jacob Mathew[1], S Karthick[2], Pavamana Bhat[3]
[1,2,3]*Dept. of Software Engg.,SRMIST,India*

***Abstract*—In today's world, automated recommendation is a big field that large companies like Netflix, Spotify and even multiple e-commerce websites use. It is a tiresome and languid task to make sense of the vast and diverse information provided to make a choice. An ideal solution would be to use a recommender system to help ease the users' decision-making ability. Many techniques to perform recommendation such as content-based and collaborative recommender systems are present but due to limitations with respect to these traditional methods, a deep learning model provides better results. Deep learning help the system to gain a better perspective of the users and items and thus improves the accuracy of the recommendation. In this article, we give a brief summary of the traditional techniques and then survey a few deep learning recommendation systems.**

I. INTRODUCTION

With the advancement in technology, we see a large influx of products and information. Thus making a concise and informed decision is becoming hard for users. Recommender systems have proven to be a breakthrough. It is able to process huge quantity of data and support the user in their decision-making ability. In recommender systems items are suggested to users based on their history of purchases and past rating of other items. Traditional approaches to recommendation systems like Collaborative Filtering and Content-based recommendations are a step forward but with sparse data available, they were unable to provide optimal results. It is a tiresome and languid task to make sense of the vast and diverse information provided to make a choice. With the advancement in technology and the ability to do complex computations by means of machine learning and artificial intelligence, we are able to handle and interpret larger and more complex data better. Research is going into applying deep learning technology to build better and more optimal recommendation systems. Large companies like Netflix, Spotify and many e-commerce websites, have used some of these concepts. This paper is organized as follows: A brief introduction into the different traditional methods, a simple summary of the Deep Learning approaches for recommender systems followed by the study of deep learning recommender system. Finally, we end with concluding remarks and future works.

II. TRADITIONAL RECOMMENDER SYSTEMS

A. Collaborative Filtering (CF):

These techniques take the users past interactions or the similar tastes of the users and exclude the information about the user or item. It adheres to a simple outlook that users tend to buy items preferred by users with similar taste. The ratings either can be precise and explicit such as on a 1-5 scale or could be vague and implicit. This vague and implicit form of rating is generally feedback, such as purchases or viewed items, from users. CF techniques are broadly classified into two: Memory-based and Model-based.

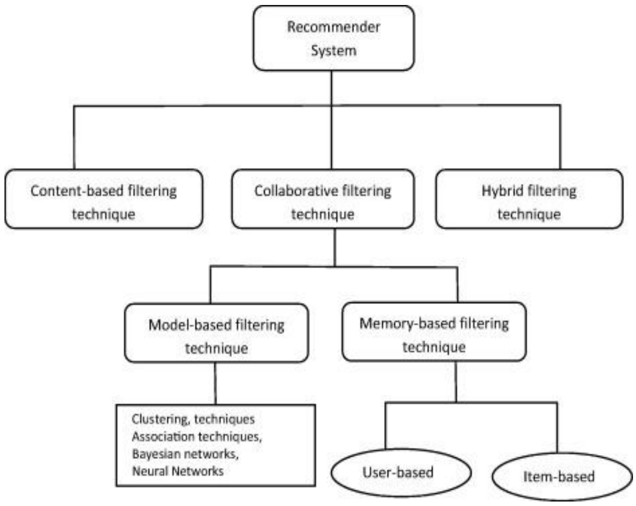

Figure 1: Summary of Recommendation Techniques

Memory-Based Collaborative Filtering:

This algorithm tries to find similarities between other users and the current user (Active user of the system) and then use the similar preferences and ratings to predict a result. There are a few advantage to this method such as we can scale it to fit a large scale of data, it is easy to implement (as we only calculate the similarity) and the arrival of new data can be easily handled. However, there are also limitations: the system is slow as it uses the entire database to make the prediction every single time and if there is sparse data and common items/ratings are very few then the obtained results are very unreliable and inaccurate.

Calculating the similarity between users or items is a very important process in this memory based collaborative filtering. Pearson correlation measures the extent to which two values relate to each other and is a popularly used technique [1, 2, 3, 11].For users' u and v the Pearson correlation is:

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}} \qquad (1.1)$$

Here *'i ε I'* sums over the items that are rated by both user u and user v.

Cosine Similarity is another popular approach used in collaborative filtering. Cosine Similarity measures the frequency of words in a document and then computes the cosine of the angle formed by the frequency vectors which results in the similarity of the documents [1, 2, 3, 4, 11]. For items i and j, the vector cosine similarity is:

$$w_{i,j} = cos(\vec{i}, \vec{j}) = \frac{\vec{i} \bullet \vec{j}}{\left\|\vec{i}\right\| * \left\|\vec{j}\right\|} \qquad (1.2)$$

Here'**•**' denotes the dot product of two vectors**.**

Model-based collaborative filtering:

The core concept behind this model revolves around Machine Learning or data mining. Here the model tries to find patterns in the training data between different ratings and then make a recommendation based on the learned model [3, 5, 11]. Some of the advantages of Model-based Collaborative Filtering is that it is able to predict better and more accurately than the Memory-based collaborative filtering, it is also scalable and can easily avoid overfitting. However, here too exists the limitation that with sparse data the model cannot accurately predict the recommendation/rating of the user. Adding data to the model is also hard as it is inflexible.

Model-based CF can use both classification as well as regression methods to achieve its purpose. Classification algorithms usually follow unsupervised learning and are used to cluster objects into categories. Clustering algorithms can be used to make categorical groups and regression models can be used to find a numerical value. Clustering algorithms like K-Means are popularly used in CF to cluster users or items into groups. Hereafter the probability of ratings can be calculated on the group and then the prediction made to achieve the model's purpose [5]. Other popular algorithms include Naive Bayes and Probabilistic Matrix Factorization**.**

B. Content based Recommender System:

This recommender system was built to handle the sparse data limitations of Model-based CF. Here information such as item description and user profiles are used to find correlations/latent factors and to help improve the accuracy of the predictions. Thus by means of this latent factors, the model will be able to predict recommendations/ratings even with sparse data.

This type of model has a few advantages: it has the ability to recommend items not yet found by the user, it helps the recommender system easily explain why/how it recommended a certain item and provides user independence in the rating process. These are highly advantages for both the user as well as the client [3, 5, 11]. There still exists limitations such as overspecialization of a certain product i.e. tendency to keep recommending a same type of item. User feedback is not simple to acquire, as the user does not rank the items. This results in difficulty in verifying the recommendation and in generating attributes for the items.

III. DEEP LEARNING AND RECOMMENDER SYSTEM

The emergence of Deep Learning as anew area of Machine Learning is revolutionary [6]. They generally consist of several layers of processing that are hierarchical and can be trained either by supervised or unsupervised approaches [3, 7, 11]. As you go into each layer, a more abstract representation of the input is obtained. The parameters of this processing can be obtained and optimized by training the model with data. Deep learning has shown effective implementation in many fields such as NLP, computer vision problems, speech recognition and language processing.

A. Architecute of Deep Learning:

Deep Learning architecture can be classified into three broader categories:

Generative deep architecture is where the algorithm looks to model how the data was generated in order to be categorized in a certain group. This generally refers to the probabilistic distribution of p(x, y) [3, 8, 9, 10, 11]. Here complex correlations between data is found for pattern analysis or for synthesis of data.

Discriminative deep architecture is where the data is categorized into a certain group without looking at how the data was generated. This generally refers to the probabilistic distribution of p(y | x) [3, 8, 9, 10, 11].

Hybrid deep Architecture is where the above two architectures are combined together to get a better and more optimized result. It ideally aims to bring about a discriminative architecture but looks to help the model improve and optimize itself better by means of the generative model or the criteria from a discriminative architecture are used to learn the parameters in a generative deep architecture [3, 8, 9, 11].

B. Deep Neural Networks (DNN):

These are multilayer neural networks with many hidden layers. Here, the weights are fully connected and most general implementations of a DNN have stacked Restricted Boltzmann Machines (RBM) or Deep Belief Networks (DBN) [3, 7, 11]. This model is able to handle a large hidden unit and can initialize parameters better. Thus, a DNN with many hidden layers is able to model the problem better but these DNN require to be trained on a large, detailed dataset and also require a lot of computational power. Hence, we have seen DNN become popular today with the improvement in technology [3, 7, 11].

Deep Auto Encoder (DAE):

It is one of the popular DNN models being used today. It is built by stacking many RBM together to build a stacked DBN with certain functionalities. Here the output target of the DAE is the data input itself. [12] proposes a good pre-training method to learn a deep autoencoder by considering each neighbor a two-layer RBM and thus in this manner, we have good parameter initialization and using back propagation, we will be able to fine-tune the model.

C. Convolutional Neural Networks (CNN):

It is a deep learning model that consist of a convoluted layer and a pooling layer in each module and these modules are then either stacked one on top of another or with a DNN on top of it to achieve a deep learning architecture [3, 11]. As seen in the diagram, the weights are shared between layers and the pooling layer sub-sampling thing output of the convolution layer leads to the optimization of the result. This is one of the most successful deep learning models and is extensively used in Computer Vision, classification and other MLPs where a biological aspect is required [3, 7, 11]. As it is a class of deep learning models, it learns complex features from within the data using the convolution and pooling functions. [13] details the initial base architecture based on which CNN was built on.

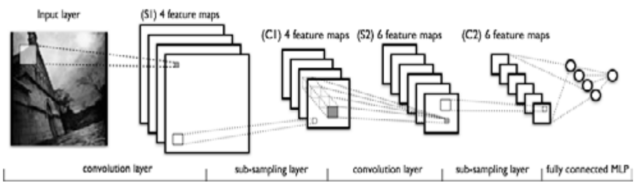

Figure 2: Architecture of CNN

IV. SURVEY OF DEEP LEARNING RECOMMENDER SYSTEM

A. Restricted Boltzmann Machine for Collaborative Filtering:

A shallow two layer Boltzmann Machine (BM) with no intra-layer communication is considered as a Restricted Boltzmann machine (RBM) [14, 19]. This model is seen to outperform the traditional methods and Matrix Factorization but as the model is not deep enough (consists of only two layers) it is not able to show great results. This model is used in modelling temporal data [3, 7, 15] and learning word embedding [3, 7, 16, 17]. More latent factors of the users/items and more accurate predictions of the ratings can be achieved by training a deeper RBM. An RBM does not use content information such as user profiles, item profiles, reviews and user feedbacks etc. RBM is typically used when we have a lot of missing data within our dataset and hence cannot deal with a user for whom no preference of item exists or recommend items that no user has rated yet [3, 7, 18].

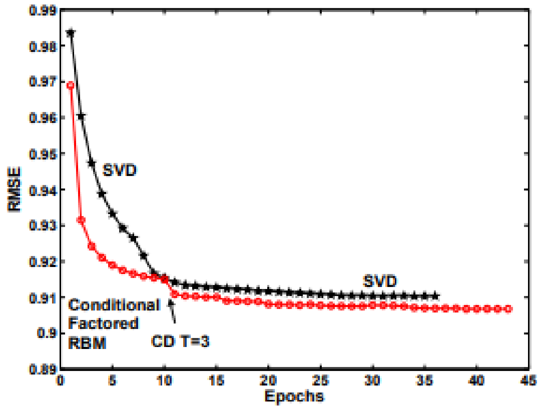

Figure 3: Performance of the RBM vs. SVD as illustrated in [19, Fig. 4]

B. Collaborative Deep Learning for Recommender Systems:

To utilize the data that the RBM was not able to use and to address the problems with the previous model, [20] introduced Collaborative Deep Learning (CDL). [20] introduces the algorithm of using a Stacked Autoencoder (SDAE) [21] and to make it do Collaborative Topic Regression (CTR) [22] to learn features of the items from review texts and model users from the subsequent Gaussian distribution.

This was the initial model that could learn from review texts but still has several drawbacks. It only models item review texts but a user’s review of the product is important to capture the user's feelings and his preferences. It also cannot grasp semantic meanings, which are useful in understanding the user’s attitude and item property [3, 7]. Semantic meanings refers to when you have two similar reviews but use different words, CDL might consider them different. Finally, word orders are important in text modelling applications [23] and CDL ignores it. Hence, to improve CDL performance, word order should be taken into consideration.

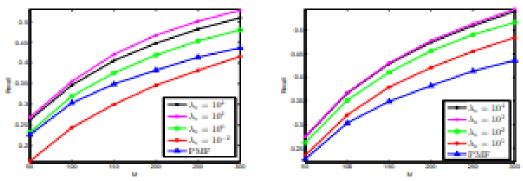

Figure 3: Performance of CDL based on recall @ M as illustrated in [20, Fig. 6]

C. Deep Content based Music Recommendation:

[23] shows us that Collaborative Filtering outperforms content-based methods but that is not true when it comes to cold-start problems i.e. items that have not been consumed before. In terms of the music, recommendations are generally done using metadata such as artist, genre and album but this is predictable. Ideally, recommender systems should suggest unknown and new items to the user and hence a better approach is to look at the music signal and hence analyze and recommend items that users have listened to [3, 7, 24]. To do this [24, 25] suggest we use a deep CNN model to learn latent factors from the music signals.

Some of the drawbacks of this model are that it uses Weighted Matrix Factorization which still is not accurate enough to be used to train a CNN and metadata is not used [3, 7, 23, 24] . A system that utilizes metadata along with latent factors from music signals can get better item features.

V. CONCLUSION

In this study, we briefly understood traditional recommender techniques as well as the basic deep learning models used to build recommender systems and then conducted a study of a few deep learning recommender system.

The summary of the traditional techniques shows that using a hybrid approach with both traditional techniques fused with deep learning features would give better results. The study also shows that many improvements are possible in both collaborative filtering as well as content-based techniques to get better results. Due to the many limitations regarding the traditional techniques, a deep learning based approach is adopted to enhance the recommendations. This is a result of the model being able to learn multiple features of the users/items and enhance upon its precision. With this model, we get a better understanding of the user's preferences and demands, thus giving us the opportunity to enhance the model to get better accuracy.

In Spite of deep learning influencing many areas, a lot of improvement is still possible. Research is underway on how to improve and build better deep learning models. Overall, limitation of the traditional recommendation approach has resulted in sub-par recommendation system and now with the advantages of deep learning coupled with recommendation techniques, we are seeing deep recommender systems that better understands the latent features and provide a more thorough and better quality recommendation.

REFERENCES

[1]] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. 1994. An open architecture for collaborative filtering of netnews. ACM on Computer Supported Cooperative Work.

[2] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. Department of Computer Science and Engineering, Florida Atlantic University.

[3] Lei Zheng. 2016. A survey and critique of deep learning on recommender systems. Department of computer science.

[4] G. Salton and M. McGill. 1983. Introduction to Modern Information Retrieval, McGraw-Hill.

[5] Poonam B. Thorat, R. M. Goudar, Sunita Barve. 2015. Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System, International Journal of Computer Applications.

[6] G. Hinton, S. Osindero, and Y. The. 2006. A fast learning algorithm for deep belief nets. Neural Computation

[7] Li Deng, Dong Yu. 2014. Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing

[8] Bengio Y. 2009. Learning deep architectures for AI. Foundations and Trends in Machine Learning

[9] Heigold, G., Ney, H., Lehnen, P., Gass, T., Schluter, R. 2011. Equivalence of generative and log-liner models. IEEE.

[10] Andrew Y. Ng and Michael I Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes.

[11] Basiliyos Tilahun Betru, Charles Awono Onana and Bernabe Batchakui. 2017. Deep Learning Methods on Recommender System: A Survey of State-of-the-art.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean.2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems.

[13] Yann LeCun. 1998. Gradient based learning applied to document recognition.

[14] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. 1994. An open architecture for collaborative filtering of netnews. ACM on Computer Supported Cooperative Work.

[15] Bengio, Y., Ducharme, R., Vincent, P. and Jauvin. 2000. A neural probabilistic language model. NIPS.

[16] Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. Computational Linguistics.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean.2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems.

[18] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. ACM.

[19] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. ACM. Machine learning.

[20] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. ACM

[21] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and PierreAntoine Manzagol. 2010. Stacked denoising autoencoders, learning useful representations in a deep network with a local denoising criterion. The Journal of Machine Learning Research.

[22] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. International conference on Knowledge discovery and data mining. ACM.

[23]] Hanna M Wallach.2006. Topic modeling: beyond bag-of-words. International conference on Machine learning. ACM.

[24] Xinxi Wang and Ye Wang. 2014. Improving content based and hybrid music recommendation using deep learning. ACM on Multimedia.

[25] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep contentbased music recommendation.