

Searching an Optimal Algorithm for Movie Recommendation System

Pavithra. M¹, Sowmiya. S², Tamilmalar. A³, Raguvaran. S⁴

^{1,2,3}Student of Final year, Bachelor of Engineering,

⁴Assistant Professor, Department of Computer Science, KPR Institute of Engineering and Technology, Arasur, Coimbatore, India.

Abstract - In the spread of information a movie recommendation is important in our social life due to its strength in providing enhanced entertainment. There are different genres, cultures and languages to choose from in the world of movies. Such a system can suggest a set of movies to users based on their interest, or the popularities of the movies. On an average of one year movie survey 600 movies are released in Hollywood. For streaming movie services like Netflix, recommendation systems are essential for helping users find new movies to enjoy. So far, a decent number of works has been done in this field. But there is always room for renovation. In this paper, we design and implement a movie recommendation system.

Key Words: Analysis, Movie Recommendation, K-means, Clustering, Machine learning, Neural networks.

1. INTRODUCTION:

Movies are a part and parcel of life. There are different types of movies like some for entertainment, some for educational purposes, some are animated movies for children, and some are horror movies or action films. Movies can be easily differentiated through their genres like comedy, thriller, animation, action etc.

Movie streaming services like Netflix, Hulu, Amazon Prime, and others are increasingly used by consumers to enjoy video content. For example, in 2017 Netflix subscribers collectively watched more than 140 million hours per day¹ and Netflix surpassed \$11 billion in revenue in 2017. In fact, roughly 80% of hours streamed at Netflix were influenced by proprietary recommendation system. Undoubtedly, movie streaming services have become an integral part of how we consume video content today, and the importance of movie recommendation systems cannot be understated—they are an integral part of how we consume video content today.

The objective of this paper is to separate users using clustering algorithm in order to find users with similar taste of movies. Machine learning approaches are used to guess what rating a particular user might give to a particular movie so that this information can be used to recommend movies to viewers.

RELATED WORK

GaurangiTilak [1] introduced MovieGEN, an expert system for movie recommendation. They implemented the system using machine learning and cluster analysis on the basis of hybrid recommendation method. Based on the Support Vector Machine prediction it selects movies from the dataset, clusters the movies and develops questions to the users. Based on the user's answers, it refines the movie set and finally recommends movies to the users.

Paterek[2] et al used Singular Value Decomposition (SVD) to reduce complexity of collaborative filtering method. As the number of items increases, the amount of computation of the collaborative filtering method becomes very large. By applying SVD, the dimension is reduced, and the collaborative filtering algorithm can be feasible. The performance is also improved.

Kula [3] et al used metadata to represent user and item as vector. The recommendation is created by using the user and item vector. To learn item vector, each property of item is embedded as vectors. The summation of each property vector is then used as item vector. User vectors are learned in a similar way using user-preferred metadata However, in this case, since the metadata embedding values are simply added, the weight of metadata cannot be considered.

Baatarjav, Phithakkitnukoon, and Dantu[4] introduced a group recommendation system for the popular social network Face book understanding the problem users go through to find right groups. They used a combination of hierarchical clustering technique and decision tree. They worked on understanding groups by analyzing the member profiles.

SURVEY ON MOVIE RECOMMENDATION SYSTEM

1.1 Deep Neural Network Model (DNN):

Deep Neural Networks (DNNs) have achieved great success in various fields, such as computer vision, speech recognition and natural language processing; however, there are few studies on recommendation systems with these technologies. Some researchers have recently proposed recommendation models based on deep learning, but most of these models use additional features, such as text content and audio information, to enhance their performances.

Considering that the above-mentioned information may be difficult to obtain for most recommendation system, recommendation model based on DNNs that does not need any extra information aside from the interaction between users and items.

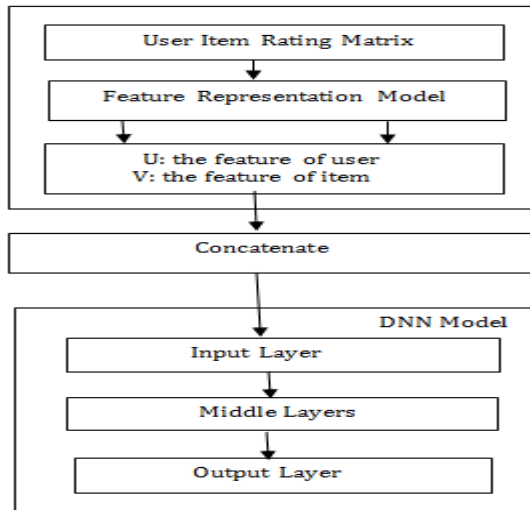


Fig-(a) Deep Neural Network Architecture

In the input layer, the input vector x_0 is concatenated by the latent features of users and items; therefore, for any record R_{ij} , we have

$$x_0 = \text{concatenate}(U_i, V_j)$$

Where the function concatenate() is used to concatenate two vectors. When x_0 passes through the first hidden layer, the output of the first hidden layer is obtained by the following equation

$$x_1 = \text{activation}(W_1x_0 + b_1)$$

We can obtain the output at the l -th hidden layer:

$$x_l = \text{ReLU}(W_lx_{l-1} + b_l)$$

1.2 Collaborative filtering:

Collaborative filtering system recommends items based on similarity measures between users and/or items. The system recommends those items that are preferred by similar kind of users.

Collaborative filtering has many advantages:

1. It is content-independent i.e. it relies on connections only.
2. Since in CF people makes explicit ratings so real quality assessment of items are done.
3. It provides serendipitous re commendations because recommendations are base on user's similarity rather than item's similarity.

Recommender system area, unit a very most important part of the knowledge and ecommerce system. They represent a strong technique for facultative users to filter through giant info and products areas. Nearly twenty years of analysis on cooperative filtering have junction rectifier to a varied set of algorithms and an expensive assortment of tools for evaluating their performance. Analysis within the field is occupy the direction of a richer understanding of however the recommender technology which is embedded in specific domains. The differing personalities exhibited by completely different recommender algorithms shows that suggestion isn't a one size fits all downside. Specific tasks, info desires, and item domains represent distinctive issues for recommenders, and style and analysis of recommenders has to be done supported by the user tasks. Effective deployments should begin with careful analysis of prospective users and their goals. Supporting this analysis, system designers have a number of choices for the selection of algorithmic rule and for its embedding within the close user expertise. In concert of the foremost productive approaches to assembling recommender systems, cooperative filtering (CF) uses the legendary preferences of a bunch of users to create recommendations or predictions of the unknown preferences for different users. During this paper, we have a tendency to 1st introduce CF tasks and their main challenges, like information scantiness, measurability, synonymy, gray sheep, shilling attacks, privacy protection, etc., and their attainable solutions. We have a tendency to then gift 3 main classes of CF techniques: memory-based, model-based, and hybrid CF algorithms, with examples for representative algorithms of every class, and analysis of their prognostic performance and their ability to deal with the challenges. From basic techniques to the progressive, we have a tendency to conceive to gift a comprehensive survey for CF techniques, which might be served as a roadmap for analysis and follow during this space. This paper discusses a large kind of the alternatives obtainable and their implications, planning to offer each researcher's with an introduction to the vital problems underlying recommenders and current best practices for addressing these problems.

Why Collaborative based filtering?

Some movie recommendation system had certain drawbacks such as, it asks a series of questions to users which was time taking. On the other hand it was not user friendly for the fact that it proved to be stressful to a certain extent.

Keeping in mind these shortcomings, we have developed a movie recommendation system that recommends movies to users based on the information provided by the users themselves. In the present study, a user is given the option to select his choices from a set of attributes which include actor, director, genre, year and rating etc. We predict the users choices based on the choices of the previous visited

history of users. The system has been developed in PHP and currently uses a simple console based interface.

1.3 Clustering k-means algorithm:

The processes of clustering and collaborating are mentioned below:

1. The user enters the genre of the movie (for eg: action, romance etc).
2. Partition of the data set is done according to the genre of the movies. Partition reduces the data size considerably, and it provides the platform for clustering.
3. K-Means clustering is applied on the partitioned data set, to form cluster of movies having the same genre and characteristics.
4. The user can then choose his movie from the clusters obtained.

The first task carried out to achieve an efficient and quick execution is the partitioning of the sparse item set of movies into partitions based on the genre requested by the user. The partitioning reduces one large-dimensionality item space into a set of smaller-dimensionality spaces with fewer items, less ratings, and often less users. As a result of this the time to compute prediction decreases since there is less data to consider. The prediction accuracy also increases as the ambiguities generated by ratings on items of dissimilar content have been removed. This is followed by the clustering of the requisite partition in order to divide the movies set into clusters of similar movies. The clustering is achieved by considering parameters of the user's age and the rating provided by those users to the movies under the partition. The age of the user has been considered as a parameter in clustering the movies following the fact that movies of different genres attract users from different age groups. Thus clusters with much more accurate results are likely to be found under such an approach.

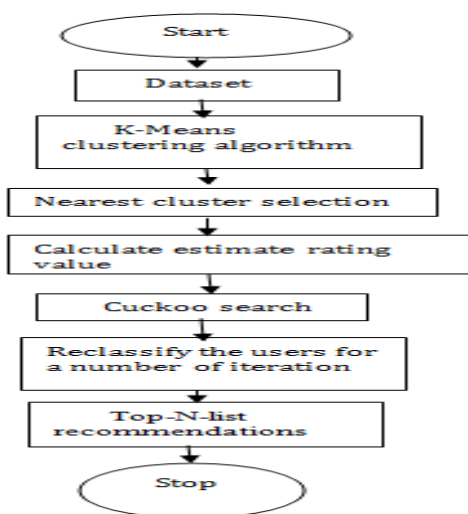


Figure (b) Clustering Framework

Meta data based word2vec algorithm:

The Word2vec algorithm is one of the deep learning methods widely used in various tasks related to document analysis such as document clustering and recommendation. Values of these metadata are embedded as vector and are used as input and output of proposed Word2Vec network. Movie embedding is also used as input and output with meta data embedding.

Word2Vec algorithm uses movie embedding and metadata embedding as input. The input and output pairs can be obtained from the user's movie purchase and viewing history data. Movies which is rated as highest score from a same user are set to a positive movie set, and randomly selected movie pairs from the set are used as input and output of the Word2Vec Network.

The movie and metadata vectors given as inputs are initialized with pre-trained embedding using the Word2vec algorithm. Two methods are used to obtain the pre-trained metadata vector. First, we use the same method as describe above. That is, for the movies rated highest score by the same user, pairs of metadata included in the movies are used as input and output of the Word2Vec network. Second, if there are different metadata values included in a same movie, the pair of the metadata values are used as input and output data.

1.4 Content based filtering:

A content based recommender works with the knowledge that the user provides, either by rating or by clicking on a link. Supported that knowledge, a user profile is generated, which is then used to make suggestions to the user. As the user provides additional inputs or takes actions on the recommendations, the engine becomes more and more accurate content-based filtering is about concerning attributes to items, in order that the algorithmic rules is aware of one thing concerning the content of every item within the info. However where do these attributes come from? And the solution depends largely on what we are attempting to recommend. If we are working on with text ,we may be able to programmatically extract keywords using Natural Language Processing techniques, though these approaches have their own drawbacks. Alternative types of content might come across with different amounts of metadata, though such incomplete knowledge is usually incomplete and covers solely the most basic reasonable kind of knowledge, making it of limited value when it comes to recommendations.

Why content based filtering?

Collaborative filtering may be the state of the art when it comes to machine learning and recommender systems, but content-based filtering still has a number of advantages, especially in certain circumstances.

- Results tend to be highly relevant.
- Recommendations are transparent.
- Users can get started more quickly.
- New items can be recommended immediately.
- It's technically easier to implement.

1.5 Demographic filtering:

It aims to classify the user based on personal attributes and build recommendations based on demographic categories. The users area unit will be divided into demographic categories in terms of their personal attributes. These categories function the input file to the recommendation method. The target of this method is to seek out the categories of individuals sort of a bound product. If individuals from category C like products and there's person c (this user belongs to category C), who has not seen nonetheless products, then this product will be recommended to person c. the purchasers offer the personal knowledge via surveys that they fill in throughout the registration method or will be extracted from the purchasing history of the users. This method might not require assembling the complicated knowledge like history of users' purchases and ratings. However, the weaknesses of DF area unit that the classification will be too general and this results in lose the individuality of the users, this methodology uses knowledge that area unit provided by users. This knowledge will be either incomplete or untrue and the classification is formed per the customer's interest, that's tends to vary over time. DF does not support the adoption of the user profile to changes.

Classification based on algorithm and dataset:

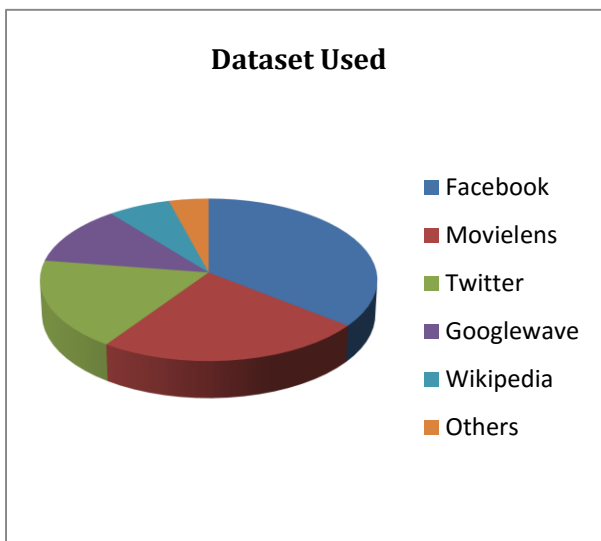


Fig –(c): Classification based on dataset

Various publicly available datasets are available for researchers. Figure(c) shows the references for datasets and domains. Notably, most of the research papers use more than one dataset termed as “Multiple Dataset”. For example, datasets from Delicious.com and Movie Lens are used more. The “Others” and “Custom” category includes the private network for university data, business transaction data website, some entertainment sectors, travel, television programs and small private sectors, which are maintained by their own organizations. Movie Lens is the mostly used publicly available dataset. However, Face book, Flicker, Stack Overflow, Google wave, and Wikipedia are the least used datasets.

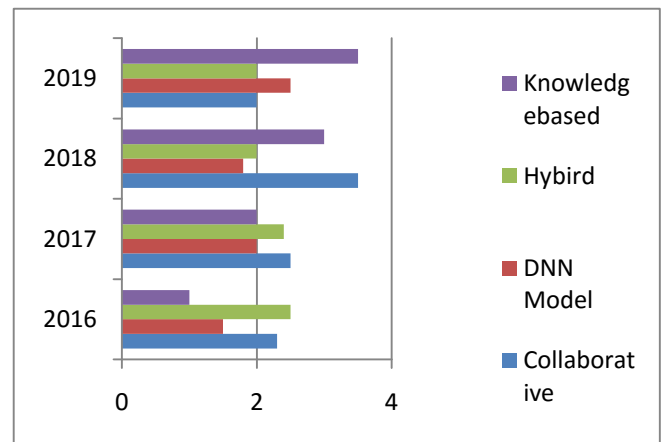


Fig -(d): Classification based on algorithm

2. Knowledge based recommendation system:

A knowledge based recommender system is one that uses knowledge about users and products to pursue a knowledge-based approach to generating a recommendation, reasoning about what products meet the users need. Each of these approaches has its strengths and weaknesses. As a collaborative filtering system collects more ratings from more users, the probability increases that someone in the system will be a good match for any given new user. However, a collaborative filtering system must be initialized with a large amount of data, because a system with a small base of ratings is unlikely to be very useful. Further, the accuracy of the system is very sensitive to the number of rated items that can be associated with a given user. These factors contribute to a “ramp-up” problem: until there is a large number of users whose habits are known, the system cannot be useful for most users, and until a sufficient number of rated items has been collected, the system cannot be useful for a particular user. A similar ramp-up problem is associated with machine learning approaches to recommendation. Typically, good classifiers cannot be learned until the user has rated many items. A knowledge-based recommender system avoids some of these drawbacks. It does not have a ramp-up problem since its recommendations do not depend on a base of user ratings. It

does not have to gather information about a particular user because its answers are independent of individual tastes. Knowledge-based recommender systems perform a needed function in a world of ever-expanding information resources. Unlike other recommender systems, they do not depend on large bodies of statistical data about particular rated items or particular users. Our experience has shown that the knowledge component of these systems need not be prohibitively large, since we need only enough knowledge to judge items as similar to each other. The knowledge base is one of the main components of Knowledge Based Systems. Depending on the kinds of KBRS, the nature of the knowledge base varies. Knowledge of the recommendation system can take the form of a simple data base, or it can contain domain ontology, formalized (expert) knowledge, or the knowledge can also amount to a case base. The nature of the knowledge base and the recommendation strategy are closely related and influence one another. Because a KBRS provides a personalized suggestion to the user, a user profile needs to be created. The content of such a profile depends on the methodology and recommendation strategy considered. Overall, we can say that the profile is composed of the user preferences, tastes, interests, needs, etc. The information needed for the identification of the user requirements regarding the particular recommendation problem is gathered in the profile. This info can be collected explicitly or implicitly. The former implies for example, questionnaires handed to the users. The latter implies, on the other hand, an analysis of the user behavior over time in order to extract information about the preferences.

3. CONCLUSIONS

Different techniques has been incorporated in recommender systems. Associate analysis of various types recommender system supported different factors is additionally done. The potency of recommender system is analyzed taking different datasets. Trust within the recommender system has been exaggerated in recommender system exploitation varied algorithms.

Approximately all the analysis has been done on cooperative or hybrid recommender system but terribly less work is finished on content based mostly recommender system. Thus the matter of "information overload" has been resolved by the Recommender system. This expeditiously provides personalized list of recommendations supported the users behavior and interests, therefore the users need not relay on someone to advocate what he needs to do next.

Most of the recommender system totally supported users interests associated to advocate an item to specific user we tend to choose similar users. Further, knowledge-based recommender systems actually help users explore and thereby understand an information space. Users are an integral part of the knowledge discovery process,

elaborating their information needs in the course of interacting with the system. One need only have general knowledge about the set of items and only an informal knowledge of one's needs; the system knows about the tradeoffs, category boundaries, and useful search strategies in the domain. Therefore knowledge based recommender systems are strongly trusted than other types of recommender systems.

REFERENCES

- [1]Barkan, Oren, and Noam Koenigstein. "Item2vec: neural item embedding for collaborative filtering." *Machine Learning for Signal Processing (MLSP)*, 2016 IEEE 26th International Workshop on. IEEE, 2016.
- [2] Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 2016.
- [3]Davidson, James, et al. "The YouTube Video Recommendation System."
- [4]B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-a case study,"DTIC Document, Tech. Rep., 2000.
- [5]A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization,"in *Advances in neural information processing systems*, 2008, pp. 1257-1264.
- [6]Naren Ramakrishnan; Benjamin J. Keller; Batul J. Mirza; Ananth Y. Grama; George Karypis (2001). "Privacy Risks in Recommender Systems". *IEEE Internet Computing*. Piscataway, NJ: IEEE Educational Activities Department. 5 (6): 54-62. doi:10.1109/4236.968832. ISBN 1-58113-561-0.
- [7].B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, and J. Riedl. *Movielens unplugged: experiences with an occasionally connected recommender system*. In *Proceedings of the 8th international conference on Intelligent user interfaces*.
- [8].An effective collaborative movie recommender system with cuckoo search Author RahulKatarya & Om PrakashVerma <https://doi.org/10.1016/j.eij.2016.10.002>.
- [9]Rashid A, Albert I, Cosley D, Lam S, McNee S, Konstan J, Riedl J (2002) Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the proceedings of international conference on intelligent user interfaces*.
- [10]Balabanovic M, Shoham Y (1997) Fab: content-based collaborative recommendation. *Commun ACM* 40(3):66-72.
- [11]"Facebook, Pandora Lead Rise of Recommendation Engines - TIME". *TIME.com*. 27 May 2010. Retrieved 1 June 2015.

[12]HoseinJafarkarimi; A.T.H. Sim and R. Saadatdoost A Naïve Recommendation Model for Large Databases, International Journal of Information and Education Technology, June 2012.

[13]Rise of the Netflix Hackers Archived January 24, 2012, at the Wayback Machine.