International Journal of Research in Engineering, Science and Management
Volume 4, Issue 7, July 2021
https://www.ijresm.com | ISSN (Online): 2581-5792

Hybrid Movie Recommender System

Pushpa Ganapati Padti[1*], Kavya Hegde[2], Prasanna Kumar[3]
[1,3]*Student, Department of Computer Science and Engineering, Srinivas Institute of Technology, Mangalore, India*
[2]*Professor, Department of Computer Science and Engineering, Srinivas Institute of Technology, Mangalore, India*

***Abstract*: The need for Recommendation Systems is increasing day by day as companies using recommender systems focus on increasing sales as a result of very personalized offers and an enhanced customer experience. Recommendations typically speed up searches make it easier for users to access the content they're interested and surprise them with offers they would have never searched for. This paper demonstrates the usage of machine learning algorithms in the movie recommender system. In this paper, we have introduced two types of filtering, content-based and collaborative-based filtering. Furthermore, we have given the analysis of the hybrid model.**

***Keywords*: Collaborative-based filtering, Content-based filtering, Hybrid model.**

1. Introduction

A recommendation system is a type of information filtering system that faces the challenge of assuming a user's priorities and making recommendations based on those priorities. The consumer has access to a broad variety of recommendation system applications. The popularity of recommendation systems has steadily grown, and they have recently been introduced in almost all popular online platforms. Films, podcasts, books, and videos, as well as colleagues and stories on social media, commodities on e-commerce platforms, and people on commercial and dating websites, all contribute to the content of such a scheme.

These systems will also retrieve and process data regarding a user's interests, and then use that information to enhance their recommendations in the future. Users have come to expect positive outcomes as recommender systems have progressed. They have a downside when it comes to programs that are unable to make effective recommendations. If a music streaming program can't predict and play the songs that the user wants, the user will just avoid using it. As a result, professional companies put a high priority on improving their recommendation systems. The Content based recommendation procedure searches for the user's likes and dislikes and constructs a User-based Profile. We look for item profiles and their corresponding user ratings when creating a user profile. The user profile is made up of the total of the item profiles, with the combination representing the ratings that the customer or user has provided.

We estimate the resemblance of the user profile with all the items in the database using cosine resemblance between the user-generated profile and item profile after the user profile has been generated. Since the noticed ratings are often highly connected across multiple users and objects, the basic methodology of collaborative filtering systems is that these undetermined ratings should be credited. Assume there are two users, who have very similar tastes.

If the scores, as both have stated, are quite similar, the fundamental algorithm can be used to determine their similarity. In such instances, there's a fair chance that the scores, even though only one of them has a definite value, would be identical. This similarity can be used to draw inferences regarding values that are only partially defined. For the calculation process, almost all collaborative filtering projects emphasize using either item associations or user associations. Both forms of correlations are used in this model.

2. Related Works

There are many recommender systems are available. These systems use different approaches, such as CF, CBF, and hybrid to recommend the preferred items. These approaches are discussed as follows:

Baolin Yi et al [1] proposes a Deep Learning based recommendation system that aims to integrate hybrid learning and feedback systems using multiple side information. All of these RS are based on CF models and hence to improve the accuracy of the predictions/suggestions these models have many kinds of side information like Social Recommendation [6] which uses social relations or trust relations. The above resource uses DMF (Deep Matrix Factorization) which basically means to merge two or more factors of CF (Collaborative filtering) using deep learning and we implement the Implicit Feedback(IF) technique that seeks to avoid this bottleneck by inferring something similar to the ratings that a user would assign from observations that are available to the system. Such an approach could greatly extend the range of applications for which recommender systems would be useful. But the cost and complexity of

Implementation of the feedback system are immensely difficult. Shanshan Wan et al., [3] proposes an e-learning recommender system that uses a hybrid filtering(HF) recommendation approach combining learner influence model (LIM), self-organization based (SOB) recommendation strategy, and sequential pattern mining (SPM) for

*Corresponding author: pushpapadti12@gmail.com

recommendation learning objects (LOS) to learners.

These results show that the model can provide personalized and diversified recommendations, and it also shows promising efficiency and adaptability in e-learning scenarios. The only drawback is that the RS will keep on recommending top-rated products as both CF and CBF, the two factors that make the Hybrid system only concentrate on purchase history and product features alone.

Shuai Zhang et al., [4] proposes a technique called Factorized Metric Learning (FML), which finds a low-rank structure in metric vector spaces. In this approach, users and items are represented as points in a multidimensional coordinate system (i.e., metric vector space). This technique used is more accurate than metric learning-based models. Recommendations are made based on user and item closeness, defined in metric space. However, unlike other metric learning approaches, the key feature behind our model is that it factorizes the metric space. Its performance may be hindered by the simple choice of interaction. The result proposed the FML model which treats users and items as points in a metric vector space, thus learning the positions of these points via factorization, which enabled FML to address the two classic recommendation tasks: rating prediction and item ranking. Experiments on several real-world datasets from different domains showed that this model performs better than neural network and metric learning-based models.

3. Types of Recommendation System

A. Content-based Filtering

Content-based recommendation systems analyze item descriptions to spot items that are of particular interest to the user. Because the small print of advice systems differ supported the representation of things. Content-based recommendation systems try to recommend items similar to those a given user has liked in the past. Indeed, the basic process performed by a content-based recommender consists of matching up the attributes of a user profile in which preferences and interests are stored, with the attributes of a content object (item), to recommend to the user new interesting items. The Content-Based Recommendation procedure searches for the user's likes and dislikes and constructs a User-based Profile. The user profile is made up of the total of the item profiles, with the combination representing the ratings that the customer or user has provided. Benefits of a content-oriented procedure include the fact that other users' information or data is not required, and the recommender system will recommend new goods or something that has not yet been evaluated; However, the recommender system would not recommend products that are not in the same category as the items for which the user has provided ratings. Content-based filtering algorithms usually fail if the user profile or the location descriptions are not well structured.

B. Collaborative Filtering

Collaborative Filtering is that the process of filtering or evaluating items using the opinions of others. While the term collaborative filtering (CF) has only been around for a bit over a decade, CF takes its roots from something humans are doing for hundreds of years - sharing opinions with others. Collaborative filtering (CF) predicts user preferences in item selection based on the known user ratings of items. As one of the most common approaches to recommender systems, CF has been proved to be effective for solving the information overload problem. CF can be divided into two main branches: memory-based and model-based. Most of the present research improves the accuracy of Memory-based algorithms only by improving the similarity measures. But few Types of research focused on the prediction score models which we believe are more important than the similarity measures.

C. Hybrid Filtering

Hybrid recommendation systems are a mix of single recommendation systems as sub-components. This hybrid approach was introduced to cope with the problem of conventional recommendation systems. This approach combines two or more recommendation approaches in different ways to benefit from their advantages. Generally, in hybrid systems, collaborative filtering is combined with another technique in a weighted way.

4. Proposed Work

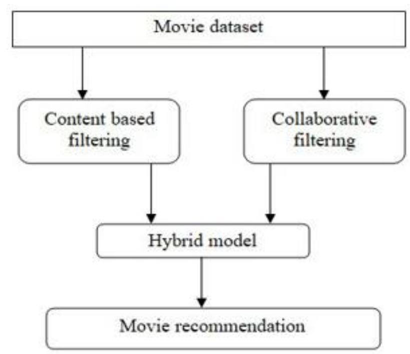

Fig. 1. Block diagram of the system

The above block diagram is the proposed system, which uses a content-based and collaborative filtering method to build the hybrid. So through a hybrid model user will get a better recommendation based on the ratings and interests. There are many popular public databases available, which have been widely used to recommend movies and other entertainment media. We conducted Experiments using public databases, such as MovieLens.

In this proposed model MovieLens dataset is given to both content-based filtering and collaborative filtering.

Content-based filtering makes recommendations by using keywords and attributes assigned to objects in a database and matching them to a user profile. The user profile is created based on data derived from a user’s actions, such as purchases, ratings (likes and dislikes), downloads, movies searched for on a website, and clicks on movie links. As the user provides more inputs or takes actions on those recommendations, the engine becomes more and more accurate.

Collaborative filtering is based on the assumption that people like things similar to other things they like, and things that are liked by other people with similar tastes. It doesn't need anything else except users' historical preference on a set of items. Because based on historical data, the assumption here is that the users who have agreed in the past tend to also agree in the future. This model takes user-id as input and processes this information using content-based filtering and collaborative filtering. Based on the profile of the user the hybrid model will recommend the top 10 movies.

5. Implementation

A. Datasets

We have used the MovieLens dataset, which is publicly available. MovieLens provide a full data set that contains 26,000,000 ratings and 750,000 tag applications applied to 45,000 movies by 270,000 users.

Table 1
Attributes in the dataset

Attributes	Description
movieId	Id gave to the specific movie
title	Title of the movie
genres	Genres the movie belongs to
userId	Id gave to the specific user
rating	Ratings are given by the specific user to a specific movie

The figure 2 is shows the distribution of the user rating in the movie lens dataset.

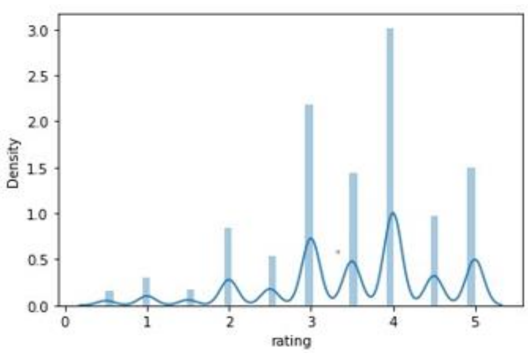

Fig. 2. Distribution of user ratings in dataset

B. Algorithm Used

Content-based filtering relies on assigning attributes to database objects so the algorithm knows something about each object. These attributes depend primarily on the movies that are being recommended. User profiles are another element crucial to content-based recommender systems. Profiles include the database objects the user has interacted with—purchased, browsed, watched.

Attributes appearing across multiple objects are weighted more heavily than others. This helps establish a degree of importance because not all of an object's attributes are equal to the user. Based on attribute weightings and histories, the recommender system produces a unique model of each user's preferences. The model consists of attributes the user is liable to like or dislike based on past activities, weighted by importance. User models are compared against all database objects, later assigned scores based on their similarity to the user profile.

1) Creating a TF-IDF Vectorizer

The TF*IDF algorithm is used to weigh a keyword in any document and assign the importance to that keyword based on the number of times it appears in the document, the higher the TF*IDF score (weight), the rarer and more important the term, and vice versa. Each word or term has its respective TF and IDF score. The product of the TF and IDF scores of a term is called the TF*IDF weight of that term.

The TF (term frequency) of a word is the number of times it appears in a document;

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

The IDF (inverse document frequency) of a word is the measure of how significant that term is in the whole corpus;

IDF(t) = log_e (Total number of documents / Number of documents with term t in it).

TF-IDF Calculation:

$W_{x,y} = tf_{x,y} * \log(N/df_x)$, where;
$tf_{x,y}$ = frequency of x in y
df_x = number of documents containing x
N = total number of documents

2) Vector Space Model

In this model, each item is stored as a vector of its attributes (which also are vectors) in an n-dimensional space, and therefore the angles between the vectors are calculated to work out the similarity between the vectors.

The method of calculating the user's likes/dislikes/measures are calculated by taking the cosine of the angle between the user profile vector and also the document vector. the rationale behind using cosine is that the worth of cosine will increase because the angle between the vectors decreases, which signifies more similarity.

```
cosine_similarities = linear_kernel(tfidf_matrix,
tfidf_matrix) results = {}
for idx, row in ds.iterrows():
    similar_indices = cosine_similarities[idx].argsort()[:-
100:-1]
    similar_items = [(cosine_similarities[idx][i], ds['id'][i])
for i in similar_indices]
    results[row['id']] = similar_items[1:]
```

We've calculated the cosine similarity of every item with every other item within the dataset, then arranged them consistent with their similarity with the item i, and stored the values in 'results'.

Collaborative filtering is predicated on the idea that folks like things like other things they like, and things that are liked by people with similar tastes. It doesn't need anything except users' historical preference on a group of things. Because it's supported historical data, the core assumption here is that the users who have agreed within the past tend to also agree within the future. In terms of user preference, it's usually expressed by two categories. Explicit Rating could be a rate given by a user to an item on a wage schedule, like 5 stars for Titanic. this is often the foremost direct feedback from users to indicate what quantity they like an item. Implicit Rating, suggests user's preference indirectly, like page views, clicks, purchase records, whether or to not hear a music track, and so on. hottest approaches are supported low-dimensional factor models (model-based matrix factorization).

The CF techniques are broadly divided into 2-types,

3) Memory based approach

Memory-Based Collaborative Filtering approaches are often divided into two main sections: user-item filtering and item-item filtering. User-item filtering takes a selected user, finds users that are like that user supported similarity of ratings, and recommends items that those similar users liked. In contrast, item-item filtering will take an item, find users who liked that item and find other items that those users or similar users also liked. It takes items and outputs other items as recommendations.

The closest user or items are calculated only by using Cosine similarity coefficients, which are only supported arithmetic operations. a standard distance metric is cosine similarity. The metric can be thought of geometrically if one treats a given user's (item's) row (column) of the rating matrix as a vector. For user-based collaborative filtering, two users' similarity is measured as the cosine of the angle between the two users' vectors.

4) Model-based approach

In this approach, CF models are developed using machine learning algorithms to predict user's ratings of unrated items. The algorithms in this approach can further be broken down into 3 sub-types.

5) Matrix Factorization (MF)

Matrix decomposition can be reformulated as an optimization problem with loss functions and constraints. The idea behind such models is that the attitudes or preferences of a user can be determined by a small number of hidden factors. We can call these factors Embeddings; intuitively, we can understand embeddings as low dimensional hidden factors for items and users.

To see how a matrix is being factorized, the first thing to understand is Singular Value Decomposition (SVD). Based on Linear Algebra, any real matrix R can be decomposed into 3 matrices U, Σ, and V. Using movie example, U is an $n \times r$ user-latent feature matrix, V is an $m \times r$ movie-latent feature matrix. Σ is an $r \times r$ diagonal matrix containing the singular values of the original matrix, simply representing how important a specific feature is to predict user preference.

6) Non-parametric approach (KNN)

The idea is the same as that of memory-based recommendation systems. In memory-based algorithms, we use the similarities between users and/or items and use them as weights to predict a rating for a user and an item. The difference is that the similarities in this approach are calculated based on an unsupervised learning model, rather than cosine similarity. In this approach, we also limit the number of similar users as k, which makes the system more scalable.

6. Result

<table>
<tr><th></th><th>movieId</th><th>title</th><th>genres</th><th>svd_rating</th></tr>
<tr><td>520</td><td>608</td><td>Fargo (1996)</td><td>Comedy|Crime|Drama|Thriller</td><td>4.669302</td></tr>
<tr><td>27</td><td>28</td><td>Persuasion (1995)</td><td>Drama|Romance</td><td>4.593841</td></tr>
<tr><td>2355</td><td>3114</td><td>Toy Story 2 (1999)</td><td>Adventure|Animation|Children|Comedy|Fantasy</td><td>4.307476</td></tr>
<tr><td>702</td><td>920</td><td>Gone with the Wind (1939)</td><td>Drama|Romance|War</td><td>4.268571</td></tr>
<tr><td>815</td><td>1073</td><td>Willy Wonka & the Chocolate Factory (1971)</td><td>Children|Comedy|Fantasy|Musical</td><td>4.256211</td></tr>
<tr><td>725</td><td>945</td><td>Top Hat (1935)</td><td>Comedy|Musical|Romance</td><td>4.219327</td></tr>
<tr><td>791</td><td>1034</td><td>Freeway (1996)</td><td>Comedy|Crime|Drama|Thriller</td><td>4.198149</td></tr>
<tr><td>2329</td><td>3086</td><td>Babes in Toyland (1934)</td><td>Children|Comedy|Fantasy|Musical</td><td>4.163076</td></tr>
<tr><td>24</td><td>25</td><td>Leaving Las Vegas (1995)</td><td>Drama|Romance</td><td>4.139079</td></tr>
<tr><td>17</td><td>18</td><td>Four Rooms (1995)</td><td>Comedy</td><td>4.133271</td></tr>
<tr><td>304</td><td>346</td><td>Backbeat (1993)</td><td>Drama|Musical</td><td>4.068576</td></tr>
</table>

Fig. 3. Output of the hybrid model

The above figure shows the final outcome of the hybrid model that is recommended movies to the user. This hybrid model recommends top-10 movies to the user using content and collaborative-based filtering methods.

7. Conclusion and Future Scope

Recommender systems have the effect of guiding users in a personalized way to interesting objects in a large space of possible options. In this paper, we proposed a hybrid recommendation system which will give better result than individual method. The content-based method will filter the individual user interest based on that user history and ratings that they have given to some movies. Collaborative filtering will find another user who has the same taste as the user. By doing so it will be helpful to recommend movies to each other. So by using a combined method we can recommend better than usual. In the future, we are planned to use a method with better efficiency and accuracy. The supplementary feature that can be added to our proposed system is to avail users of a full-fledged user interface. Lastly, developing a well-integrated web application that can recommend whenever users want it to will complete the project.

References

[1] Baolin Yi, Xiaoxuan Shen, Hai Liu, Zhaoli Zhang, Wei Zhang, Sannyuya Liu, and Naixue Xiong, "Deep Matrix Factorization with Implicit Feedback Embedding for Recommendation System", IEEE Transactions on Industrial Informatics, vol.15, pp. 4591-4601, 2019.

[2] Shanshan Wan, Zhendong Niu, "A Hybrid E-learning Recommendation Approach Based on Learners' Influence Propagation", IEEE Transactions on Knowledge and Data Engineering, vol. 32, pp. 827-840, 2019.

[3] Shuai Zhang, Lina Yao, Bin Wu, Xiwei Xu, Xiang Zhang, and Liming Zhu, "Unraveling Metric Vector Spaces with Factorization for Recommendation", IEEE Transactions on Industrial Informatics, vol. 16, pp. 732-742, 2019.

[4] Shuiguang Deng, Longtao Huang, Guandong Xu, Xindong Wu, Zhaohui Wu, "On Deep Learning for Trust-Aware Recommendations in Social Networks", IEEE Transactions on Neural Networks and Learning Systems, vol. 28, pp. 1164-1177, 2016.

[5] V. Subramaniyaswamy, R. Logesh, M. Chandrashekhar and Anirudh Challa, "A personalised movie recommendation system based on collaborative filtering", Int. J. High Performance Computing and Networking, vol. 10, nos. 1/2, 2017.

[6] F Furtado, A Singh, "movie Recommendation System Using Machine Learning," vol. 10, no. 1, 2020.