Collaborative Filtering-based Recommendation System for Movies

ANURATHI BALA, Northeastern University
ARADHANA JAYAPRAKASH, Northeastern University

1 INTRODUCTION

Movie recommendation systems are a necessity for any movie streaming platform's survival and success. There are thousands of distractions to the daily movie watcher who can view what catches their eye or scroll through a selection of other movies. As the number of titles increase day by day, it can be difficult for viewers to decide what they would want to watch. Hence, in the perspective of a streaming service's success, it is important to increase customer engagement by identifying the customer's wants and needs. In order to keep these customers from joining a competitor streaming service, it is important that these platforms have a solid recommendation system that accurately predicts a user's interests.

There are several different approaches that can be used when building recommendation systems: Content-Based Filtering, Collaborative Filtering, and Hybrid Models. Content-Based Filtering focuses on a user's interest specifically and recommends items that a user likes rather than what the majority likes. Collaborative Filtering is based on the idea that a person likes things similar to what he or she has liked in the past and (or) people with similar interests. Latent-Factor Models are a type of model-based Collaborative filtering method used to capture meaningful latent connections between users and items inferred from rating patterns. Hybrid recommendation models can be used where two or more recommendation methods are implemented to boost performance.

This report explains how an accurate hybrid recommendation system is built according to a user's interests on the MovieLens data set [1]. Collaborative filtering has been proven to have the best results out of the other methods, hence this method is incorporated to reduce the probability of incorrect predictions and improve the performance of the model. The proposed system uses clustering, where a user's rating average based on a cluster was swapped with the missing ratings for a given movie by that user. Then Latent-Factor modeling is performed in which the user-movie matrix is fed to a Singular Value Decomposition (SVD) model which is used to predict ratings that a user would give for specific movies. Additionally, this method is compared with the SVD model output of a row-mean imputed user-movie matrix to have a baseline for comparison. Next, memory-based collaborative filtering using cosine similarity is performed on both dense user-movie matrices and the error between them is evaluated.

In the perspective of the contributions made in the scope of the project, the proposed missing value imputation method has a level of user-based specificity that will yield more accurate results than traditional methods of imputing missing values with row-mean. This method hasn't been used before and makes sense because the average rating of each cluster for a given user are used to impute missing values of movies of specific clusters.

The following report is organized as follows. Section 2 describes well-known approaches in regards to the development of Movie recommendation systems. Section 3 goes into more detail about the science, terminology, and algorithm behind different types of recommendations systems. Section 4 presents the proposed approach. Section 5 shows the evaluation of the proposed system. Lastly, Section 6 summarizes the results and future work that can be done.

2 RELATED WORK

In the past, several researchers have designed Collaborative Filtering recommendation systems to help recommend movies from a large set of already existing movies. Katarya Rahul [2] developed a hybrid recommendation system that used K-means clustering in conjunction with the bio-inspired artificial bee colony (ABC) optimization technique using the MovieLens data set. Additionally, Arisara Pornwattanavichai et al. [3] developed a hybrid Tweet recommendation system using Latent Dirichlet Allocation and matrix factorization. The main inspiration behind this study follows [4], where Collaborative Filtering is combined with K-means clustering of movies by genre attributes and demographic attribute segmentation.

Accurate imputation methods have explained that replacing missing values with the row-mean is acceptable for predicting ratings. However, there may be a better way to impute these values more accurately according to types of movies (genres) that a user likes. The proposed approach explores more efficient ways to provide movie recommendations using clustering and machine learning methods and to see if better accuracy can be achieved.

3 BACKGROUND INFORMATION

3.1 DIMENSIONALITY REDUCTION

Algorithms dealing with dimensionality reduction aim to transform data from a high dimensionality space to a low dimension by reducing the number of features in the data while still retain all the information represented. This method helps in processing and performing other functionalities on the data set while maintaining cost efficiency and run-time and preventing data loss.

3.2 CLUSTERING

Clustering algorithms group similar data points into a specified number of clusters such that data points within a cluster are more similar to the points that are in the same cluster than those of other clusters. In this project, K-means clustering is used where the number of clusters is determined by k and each cluster has a cluster centroid. The presence of a data point in a specific cluster depends on the proximity of the data point to the cluster centroid. The repetitive manner of allocating data points to clusters, forms the K-means clustering algorithm.

3.3 RECOMMENDER SYSTEMS

Recommendation systems use algorithms to predict preferences to users based on user activity history, item similarity etc. Content-Based recommendation systems recommend items to users based on previous items rated by the user. Collaborative Filtering recommender systems find a set of similar users and recommend items based on this set of users preferences. The focus of this project is on Collaborative filtering, specifically.

3.4 MEMORY-BASED COLLABORATIVE FILTERING

Memory-Based Collaborative filtering makes use of Cosine Similarity measures to find similar items or users based on user rating data to recommend items. They consist of two types, where the user-item filtering finds similar users and recommends items based on his or her preferences. Whereas, item-item filtering finds users who liked a particular item and recommends other items that those users liked.

3.5 MODEL-BASED COLLABORATIVE FILTERING

A Model-Based Collaborative filtering approach deploys matrix factorization to obtain latent factors that can be used for predictions. This method also helps in representing the user-item matrix in a lower dimensional space, adding in

the feature of dimensionality reduction to its methodology.

4 PROPOSED APPROACH

The problem at hand is that user-movie matrices are sparse. Thus, in order for a recommendation system to work properly, proper imputation of unrated movies by a user must be done before any recommendation system implementation can be performed. Missing values cannot be replaced with 0 because that would throw off the entire accuracy of the recommendation system. Hence, the proposed clustering method provides a more refined approach to imputing those missing values and should yield accurate results.

The key components of building the recommendation system are organized as follows: data pre-processing, implementing genre-based movie clustering, imputing missing values of the user-movie matrix with user-cluster averages, building a Model-Based and Memory-Based Collaborative filtering model, and finally evaluating error metrics using the baseline model. Figure 1. explains the layout of the recommendation system.

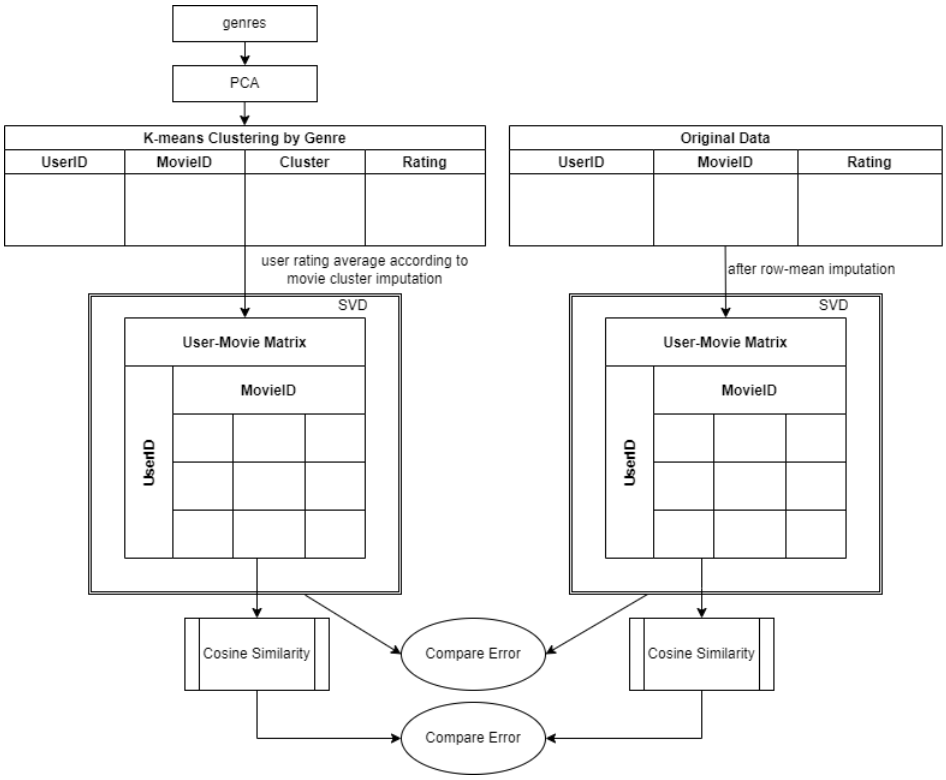

Fig. 1. Proposed Architecture

Using the MovieLens dataset, Principal Component Analysis (PCA) will be used to reduce the dimensionality of the number of genres present amongst all the movies. Next, movies are clustered by genre using the K-means clustering method. Merging this with the ratings data set by MoviedID, each user's average cluster rating is computed and replaces every NA value for a given (UserID, MovieID) pair. This matrix is then pivoted into a user-movie matrix where there is a rating for every (UserID, MovieID) combination, hence the matrix is now dense and Singular Value

Decomposition is performed. Next, in order to have a baseline comparison to check efficiency, SVD is performed on a user-movie matrix imputed by the row-mean. To explain further, the original ratings data set is transformed to a user-movie matrix. However, this matrix is sparse as there are unrated movies by some users. To address this issue, the row-mean is computed for each individual user (represented by a row) and the missing values are imputed by these row-mean values, leaving a now dense user-movie matrix. Again, SVD is performed and the results of the two methods are evaluated. In the same way, Cosine Similarity is performed on both of the dense user-movie matrices after their respective imputations and the error is evaluated.

5 EXPERIMENTS

The data set used to validate our proposed approach for the recommendation system is the MovieLens data set, which provides 1,000,209 movie ratings from over 6040 users and a selection of over 3706 movies. The MovieLens data set consists of two individual data sets: Ratings and Movies. In the Ratings data set, the features found are 'UserID', 'MovieID' and 'Rating'. The ratings are given on a scale of 1 to 5. Whereas in the Movies data set, the features found are 'MovieID', 'Title', and 'Genres'. These two data sets are used in conjunction with each other to understand genres that a user likes and the given ratings for movies for a given user.

The first step in the experiment is to import the movies data set. From the 'Genres' column of the movies data set, the individual genres are extracted and one-hot encoding is performed to convert the text representation of the genres to numerically indicate the presence of each genre in a specific movie. Additionally, the 'Title' and 'Genres' column is removed as the 'MovieID' indication is enough to differentiate between each unique movie. The next step reduces dimensions of the data set by performing Principle Component Analysis (PCA) to minimize the the number of genres from 18 genres to 12 components. In order to do this, the 18 genres are extracted and reduced to 12 components based on the 'elbow method' using cumulative explained variance. Following this step, K-means clustering is used to cluster movies based on genres into 6 separate clusters. Again, the 'k' value is determined by the elbow method using distortion. The cluster value or 'class' is assigned for each movie and the data frame is then merged with the ratings data set which consists of columns 'UserID', 'MovieID' and 'Rating'. This merged table is then organized such that a user's average rating for the cluster of that specific movie's cluster imputes the missing values for a given user-movie pair. Hence, a dense user-movie matrix is formed from this computation. Next, Singular Value Decomposition is performed and a 'k' value is selected by the elbow method once again. The error of the predicted rating matrix is computed by a specific comparison to a user-movie matrix which will be explained next. In order to have a baseline comparison to the specific clustering method proposed in this project, the ratings data set is used to form the user-movie matrix where ratings are given for each (user, movie) pair. Row-mean imputation is performed to replace the missing values in the user-movie matrix. Similarly, Singular Value Decomposition (SVD) is performed and a 'k' value is selected by the elbow method. Like mentioned before, both predicted models are compared with each other and the various error metrics are calculated. For Memory-Based Collaborative filtering, Cosine Similarity is computed for both dense matrices, and the same comparison between the two user-movie matrices are returned.

For Principle Component Analysis, the cumulative explained variance is plotted against k values and an optimal k, 12, is decided using the elbow method, as shown.

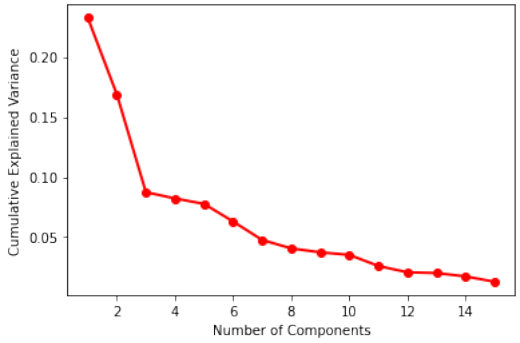

Fig. 2. Variance of 15 components on genres of movies data set (Elbow Method)

Similarly, the 'elbow method' is used K-means clustering to determine an optimal number of clusters to cluster movies by genre to. The Singular Value Decomposition model also performs the same check for an optimal number of components.

After computing the SVD model, the user-movie matrix for both with and without K-means clustering is returned. Similarly, after computing the Cosine Similarity model, the user-movie matrix for both with and without K-means clustering is returned. The user-movie matrix shows the predicted values that each user would give for each and every movie present in the data set. The result of the experiment is presented in Table 1 for a random user. Additionally, Table 2. shows the the error metrics computed for the models built.

Table 1. Predicted Ratings for UserID 1

UserID 1 Predicted Ratings Comparison	
	MovieID 1, 2, 3 Predicted Ratings
Model-Based with Clustering	4.30, 4.23, 3.99
Model-Based	4.24, 4.18, 4.18
Memory-Based with Clustering	3.56, 3.54, 2.92
Memory-Based	4.19, 4.19, 4.19

Table 2. Error Metrics for Memory-Based and Model-Based Collaborative Filtering

Error Metrics		
	RMSE	MAE
Memory Based	0.837	0.424
Model Based	0.693	0.427

For the proposed Memory-Based approach compared with the baseline model, the RMSE and MAE were 0.837 and 0.424, respectively. For the proposed Model-Based approach compared with the baseline model, the RMSE and MAE were 0.693 and 0.427, respectively. The error values were low for both methods, hence it can be concluded that the

proposed approach is plausible. One of the unexpected findings from the study was that the Memory-Based approach yielded lower error than the Model-Based approach by 0.837 - 0.693 = 0.144. This is surprising because most studies have found that the Model-Based approach outperforms the Memory-Based approach, yielding lower error.

6 CONCLUSION AND FUTURE WORK

This work proposes an algorithm using the unsupervised machine learning algorithm of K-means clustering with both a memory-based and model-based approach of collaborative filtering methods in the field of movie recommendation systems. Initially, by PCA dimensionality reduction, 18 features of genres were reduced to 12 by the elbow method for k selection. Then movies were clustered by genre according to K-means clustering which again used the elbow method approach to determine the optimal 'k' number of clusters. After deriving the user-movie matrix, it was evident that there were several missing values, leaving the matrix sparse. Since rating estimation of a sparse matrix is difficult, there had to be a way to impute the missing values accurately such that rating prediction could be continued. For every missing rating, that specific movie was routed back to the cluster it belonged to. For the missing rating for that given user, the user's cluster average for which the movie belongs to was found and imputed the missing values in user-movie matrix. Following this, SVD model-based Collaborative Filtering was performed give powerful recommendations. Additionally, the sparse user-movie matrix was imputed with the row-mean as a baseline comparison to the clustered approach. Again, the same SVD was computed to output predicted user ratings. For the computed dense matrices, cosine similarity was also computed and the predicted ratings were outputted. The results shows that both error metrics for both Memory-Based and Model-Based Collaborative Filtering were minimal. However, the Memory-Based approach showed slightly higher accuracy than the Model-Based approach. Hence, the clustering based approach proves to be feasible.

One of the main limitations of the study was the lack of user ratings. For a user that gives only a few ratings, if that user has given an extreme rating like 1 or 5, this rating will overpower through the rest of the predicted ratings, which will lead to inaccurate rating predictions. Hence, if more rating data is available for a given user, then issue of few ratings in the system for model-based approaches might be resolved.

The findings from this research show that Memory-Based Collaborative filtering is more accurate. Additionally, the error metrics yielded small values which confirms that the proposed hybrid recommendation system approach works. The method works because each cluster contains movies of similar genres. If a user rates a romance movie as 5, then chances are that he or she will vote for another romance movie close to 5.

The study performed can be improved by finding better methods of filling missing ratings in a user-movie matrix. Clustering could possibly be done on a different basis other than genres and be checked to see if it is more accurate. Also, real users that have given ratings in the data set could be used to validate whether the rating prediction works. Further studies can be done to continue to explore and improve other techniques used in recommendation systems.

REFERENCES

[1] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (dec 2015), 19 pages. https://doi.org/10.1145/2827872

[2] Rahul Katarya. 2018. Movie recommender system with metaheuristic artificial bee. *Neural Computing and Applications* 30, 6 (2018), 1983–1990.

[3] Arisara Pornwattanavichai, Pongsakorn Jirachanchaisiri, Janekhwan Kitsupapaisan, Saranya Maneeroj, et al. 2020. Enhanced Tweet Hybrid Recommender System Using Unsupervised Topic Modeling and Matrix Factorization-Based Neural Network. In *Supervised and Unsupervised Learning for Data Science*. Springer, 121–143.

[4] AFOUDI Yassine, LAZAAR Mohamed, and Mohammed Al Achhab. 2021. Intelligent recommender system based on unsupervised machine learning and demographic attributes. *Simulation Modelling Practice and Theory* 107 (2021), 102198. https://doi.org/10.1016/j.simpat.2020.102198