

Movie Recommendation System using K-Means

Kranti Kumari: 202SP011, Nikhil Bobate - 202SP017
Under the guidance of Dr. Aparna P

Department of Electronics and Communication Engineering
National Institute of Technology, Surathkal
Karnataka-575025, India

May 2021

1. Abstract

Recommender system is a filtration program whose prime goal is to predict the “rating” or “preference” of a user towards a domain-specific item or item. In our case, this domain-specific item is a movie, therefore the main focus of our recommendation system is to filter and predict only those movies which a user would prefer, given some data about the user himself or herself. This project aims at implementing a Movie Recommendation system using Memory based, item-to-item Collaborative Approach of K-Means algorithm to reduce the human effort by suggesting movies based on the user’s interests.

2. Introduction

During the last few decades, with the rise of YouTube, Amazon, Netflix and many other such web services, recommender systems have taken more and more place in our lives. They are so commonplace now that many of us use them without even knowing it. Because we can’t possibly look through all the products or content on a website, a recommendation system plays an important role in helping us have a better user experience, while also exposing us to more inventory we might not discover otherwise.

Different filtration strategies are Content based filtering and Collaborative filtering.

In Content Based filtering, the filtration strategy is based on the data provided about the items. The algorithm recommends products that are similar to the ones that a user has liked in the past. This similarity (generally cosine similarity) is computed from the data we have about the items as well as the user’s

past preferences.

In Collaborative filtering, the filtration strategy is based on the combination of the user's behavior and comparing and contrasting that with other users' behavior in the database. The history of all users plays an important role in this algorithm.

It has two variations: memory based and model based approaches. Memory based approaches directly works with values of recorded interactions, assuming no model, and are essentially based on nearest neighbours search (for example, find the closest users from a user of interest and suggest the most popular items among these neighbours). Model based approaches assume an underlying "generative" model that explains the user-item interactions and try to discover it in order to make new predictions.

Figure 1 depicts the Illustration of Content based filtering and Collaborative filtering:

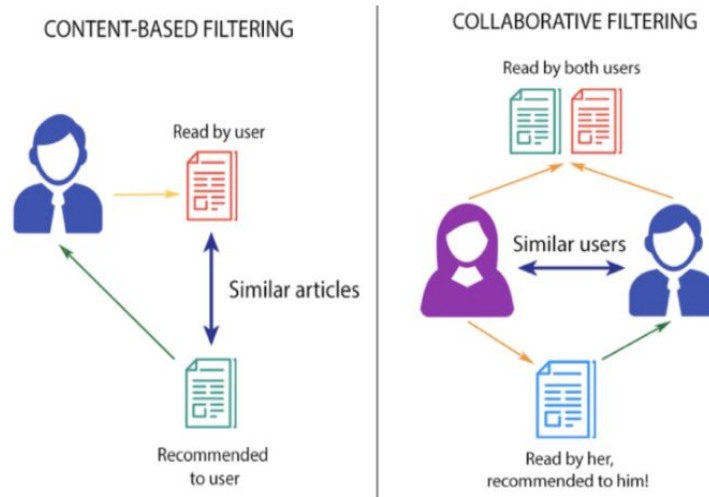


Figure 1: Filtration Strategies

For this model, we are using Movie Lens Small Latest Data-set from Kaggle.

3. Recommendation System

Our aim is to reduce the human effort by suggesting movies based on the user's interests.

3.1. Problem Statement

Content-based recommendation systems are constrained to people, these systems don't prescribe things out of the box. These systems work on individual users' ratings, hence limiting your choice to explore more. While our system which is based on a collaborative approach computes the connection between different clients and relying upon their ratings, prescribes movies to others who have similar tastes, subsequently allowing users to explore more. It will give progressively explicit outcomes contrasted with different systems that are based on the content-based approach.

3.2. Methodology

Collaborative filtering works based on users that have similar tastes.

Figure 2 depicts the Flowchart of our recommendation system:

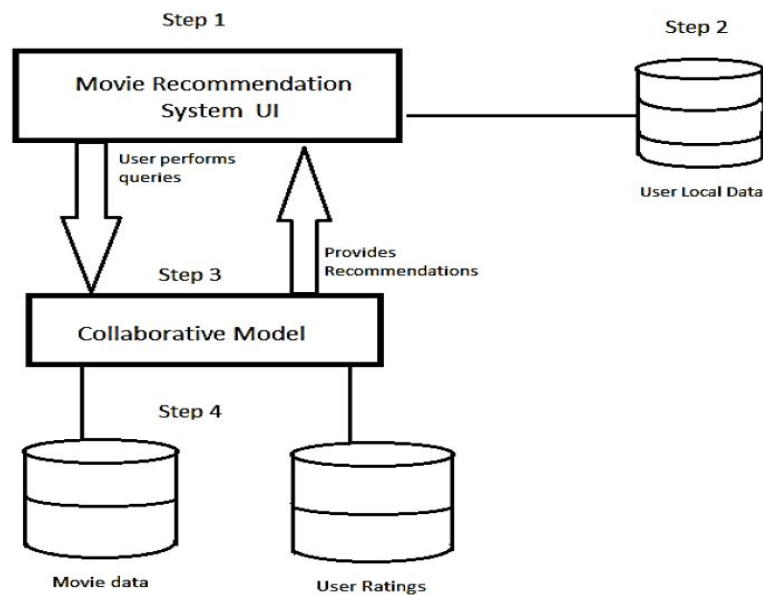


Figure 2: Architecture of movie recommendation system

Steps involved in building this model are:

1. Cleaning the dataset
 - Remove dashes from dataset and identify unique movies that have been rated.
 - Remove users who haven't at least rated 55 movies.

2. Merge datasets
 - Combine the two datasets genres and tags for metadata.
3. Perform dimension reduction
 - This dataset gives what we call a sparse matrix. So, perform singular value decomposition (SVD) for dimension reduction which will identify variables that have the most variance.
4. Run the algorithm for the target film
 - Run the collaborative filtering algorithm

4. Results

The main focus of our recommendation system is to filter and predict only those movies, which a user would prefer, given some data about the user himself or herself.

Figure 3 depicts the variance plot to see what latent dimensions to use:

```
# Plot variance to see what latent dimensions to use  
  
explained = svd.explained_variance_ratio_.cumsum()  
plt.plot(explained, '.-', ms = 10, color = 'red')  
plt.xlabel('Singular Value Components', fontsize = 12)  
plt.ylabel('Cumulative percent of Variance', fontsize = 12)  
plt.show()
```

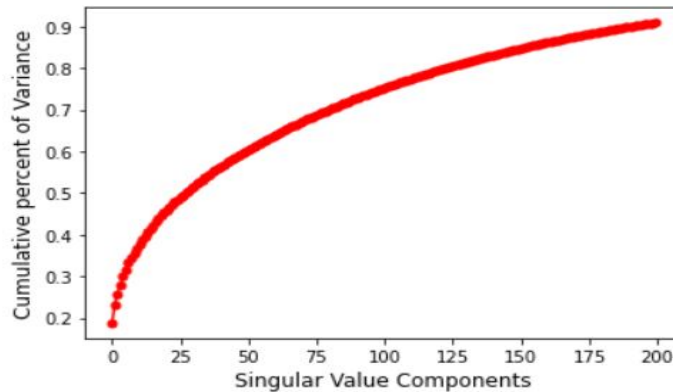


Figure 3: Variance plot

We then run the collaborative model for the target film. Figure 4 shows the output of our recommendation model for the selected movie 'Toy Story (1995)'.

	collaborative
Appleseed (Appurushido) (2004)	-0.033840
Rent-A-Cop (1988)	-0.023536
Meatballs 4 (1992)	-0.023536
House Party 3 (1994)	-0.023536
Hell in the Pacific (1968)	-0.022253
Violets Are Blue... (1986)	-0.022253
Supercop 2 (Project S) (Chao ji ji hua) (1993)	-0.022116
Stray Dog (Nora inu) (1949)	-0.018263
Equus (1977)	-0.018263
Shakes the Clown (1992)	-0.017138
Sugar Hill (1994)	-0.016789

Figure 4: Recommendation for Toy Story (1995)

5. Conclusion

This recommendation system recommends different movies to users. Since this system is based on a collaborative approach, it will give progressively explicit outcomes contrasted with different systems that are based on the content-based approach. Content-based recommendation systems are constrained to people, these systems don't prescribe things out of the box. These systems work on individual users' ratings, hence limiting your choice to explore more. While our system which is based on a collaborative approach computes the connection between different clients and relying upon their ratings, prescribes movies to others who have similar tastes, subsequently allowing users to explore more.

6. References

- [1] F. Furtado, A, Singh, A "Movie Recommendation System Using Machine Learning" : 2020, International Journal of Research in Industrial Engineering, Vol. 9, No. 1 (2020) 84–98
- [2] <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>.
- [3] Duda, Richard O., Peter E. Hart and David G. Stork., "Pattern Classification", John Wiley and Sons, 2012.