A Profile Based Movie Recommendation System

Mr. Devam Shah
devam.s1@ahduni.edu.in

Ms. Nancy Radadia
nancy.r@ahduni.edu.in

Mr. Varshil Shah
varshil.s@ahduni.edu.in

Mr. Parth Sarkhelia
parth.s3@ahduni.edu.in

***Abstract*—Our project centers around building a movie recommendation system based on Collaborative filtering. In particular we look at implementing two algorithms of item based collaborative filtering which are K-Nearest Neighbours (KNN) and Alternating Least Squares (ALS) and compare their performance in terms of standard measures.**

***Index Terms*—Recommendation System, Machine Learning, Item-based Collaborative Filter, K Nearest Neighbours, Alternating Least Square**

I. Introduction

Recommendation systems are one of the most prominent applications of Machine Learning and part of everyday life. They are well-studied and proven to provide tremendous value to internet businesses and their consumers. Recommender systems can be loosely broken down into three categories: content-based systems, collaborative filtering systems, and hybrid systems, out of which we are going for item-based collaborative filtering systems. Item-based shared filtering recommends a product based on the browsing habits of other users who looked at the same item as me. (users who looked at my item also looked at these other items)

This project uses a movie-lens dataset for training the recommendation models. MovieLens dataset contains two files, movies.csv, and rating.csv. This dataset contains 9000 unique movies and 600 unique users. Total ratings in this dataset are around 300,000.

II. Literature survey

The literature search for the recommendation system was mostly based on descriptors ”Collaborative filtering”, ”Contents filtering” and ”Hybrid Recommendation Systems”. Content-based filtering needs a lot of information about the item to recommend it, on the other hand, collaborative filtering doesn’t need anything else except the user’s historical preference on a set of items.

KNN algorithm is a simple implementation of collaborative filtering [1]. It s based on feature similarity, as to how closely a sample data point feature correlates with our training dataset, thus determining the classification of the sample data point.

Though the KNN model has some limitations like it recommends a movie that is popular among similar users without allowing for personalization, also it has scalability issues.

The second approach is the Alternating least squares algorithm which is a matrix factorization technique that seeks to decompose the rating matrix into two lower dimensional matrices. This method uses the iterative approach. During each iteration, one of the factor matrices is held constant and another one is tuned using the least square method. For the next iteration, the newly solved matrix is held constant and another matrix is tuned. This runs in a loop until the loss function is minimized.

III. Implementation

A. KNN Algorithm

When it comes to recommending items/movies, we are interested in recommending only top K items/movies to user [2]. KNN relies on labeled input data to learn a function that generates appropriate output for unlabelled data. Our first step is to clean the given dataset and reduce the number of user and movies while keeping dimension of rating constant and reshape the dataset into a format which can be given as parameters. Reducing the number of users and movie is necessary as it reduces sparsity and more importantly, KNN doesn’t work properly in case of a large number of dimension. When a movie name is given as input, the KNN model selects movies from a dataset which have a fuzz ratio of more than 50 for the given input. It then calculates Euclidean distance for each selected movie for the input movie and selects K nearest movies with the smallest Euclidean distance as per calculation. These K nearest movies are the recommended movies for the user.

B. ALS Algorithm

In collaborative filtering, matrix factorization is one of the solutions for sparse data problems. Alternating Least Square (ALS) [3] is also a matrix factorization algorithm that minimizes two loss functions alternatively obtained from matrix factorization.

In matrix factorization, the main goal is to divide the Rating Matrix into two lower dimension matrix which are the User matrix and Item Matrix. This algorithm assumes that there are k attributes or features which define each user and similarly there are k attributes or features that define each movie. Multiplying each feature of the user by the corresponding feature of the movie and adding it together gives a reasonable estimate of the user’s rating. Mathematically, we can represent the matrix factorization as follows-

$$r_{ui} = x_u^T \cdot y_i = \sum_k x_{uk} y_{ki} \tag{1}$$

where, u is number of users, i is number of item, r is rating matrix, x is user matrix, y is item matrix, k is latent factors.

The loss function can be given as:

$$L = \sum_{u,i \in S} (r_{ui} - x_u^T \cdot y_i)^2 + \lambda_x \sum_u \|x_u\|^2 + \lambda_y \sum_u \|y_i\|^2 \tag{2}$$

Now, we aim to minimise above loss function by using ALS. During each iteration, ALS holds one set of latent vectors as constant and tune the other set of latent vectors in order to minimise loss. For the next iteration , the newly solved matrix is held constant and another matrix is tuned.
Let's assume at first we hold item vectors(y_i) constant.So we take derivative of loss function with respect to user vector(x_u). Differentiating equation (2) with respect to User vector(x_u) , by assuming item vector(y_i) as constant gives-

$$\frac{\partial L}{\partial x_u} = -2 \sum_i (r_{ui} - x_u^T \cdot y_i) y_i^T + 2\lambda_x x_u^T = 0$$

Solving this we get,

$$x_u^T = r_u Y(\ Y^T Y + \lambda_x I)\ ^{-1} \tag{3}$$

Similarly, we can obtain the derivation for the item vectors by assuming solved-user vectors as constant.

$$y_i^T = r_i X(\ X^T X + \lambda_y I)\ ^{-1} \tag{4}$$

IV. Result

A. KNN Algorithm

Given below is the output of recommended movies to the user when a movie is given as an input string. The recommended movies seem similar to the given input movie.

```
makeRecommendation('Avengers',itemUserMatrixSparse,
                   recommendationModel,movieToIndex,10)
```

```
system is working....

Viewer who watches this movie  Avengers also watches following movies.
Sherlock Holmes: A Game of Shadows (2011)
X-Men: First Class (2011)
Captain America: The First Avenger (2011)
Wreck-It Ralph (2012)
Mission: Impossible - Ghost Protocol (2011)
Looper (2012)
Star Trek Into Darkness (2013)
Thor (2011)
Guardians of the Galaxy (2014)
Iron Man 3 (2013)
```

Fig. 1. KNN

B. ALS Algorithm

In the ALS algorithm, the recommended movies are generated based on the previous preferences and ratings given by the user.
The given figure depicts the list of the movie which are preferred by user id 100.

100th User's Actual Preference:

```
ratings.join(movies, on='movieId').filter('userId = 100')
.sort('rating', ascending=False).limit(10).show()
```

```
+-------+------+------+--------------------+--------------------+
|movieId|userId|rating|               title|              genres|
+-------+------+------+--------------------+--------------------+
|   1035|   100|   5.0|Sound of Music, T...|     Musical|Romance|
|   2628|   100|   5.0|Star Wars: Episod...|Action|Adventure|...|
|   1380|   100|   5.0|       Grease (1978)|Comedy|Musical|Ro...|
|    150|   100|   5.0|    Apollo 13 (1995)|Adventure|Drama|IMAX|
|   3578|   100|   5.0|    Gladiator (2000)|Action|Adventure|...|
|    364|   100|   5.0|Lion King, The (1...|Adventure|Animati...|
|   2167|   100|   5.0|        Blade (1998)|Action|Horror|Thr...|
|   1028|   100|   5.0| Mary Poppins (1964)|Children|Comedy|F...|
|   2565|   100|   5.0|King and I, The (...|Drama|Musical|Rom...|
|    261|   100|   5.0| Little Women (1994)|               Drama|
+-------+------+------+--------------------+--------------------+
```

Fig. 2. Preferences of user

The below figure depicts the list of movie which are recommended to user id 100 based on their preference and rating.

100th User's ALS Recommendations:

```
nrecommendations.join(movies, on='movieId').filter('userId = 100').show()
```

```
+-------+------+---------+--------------------+--------------------+
|movieId|userId|   rating|               title|              genres|
+-------+------+---------+--------------------+--------------------+
|  81191|   100|5.0516715|Waiting for 'Supe...|         Documentary|
|   1014|   100| 4.616705|    Pollyanna (1960)|Children|Comedy|D...|
|  52767|   100|4.5567102|        21 Up (1977)|         Documentary|
| 104319|   100|4.5465045|First Grader, The...|               Drama|
|  80969|   100|4.5424123|Never Let Me Go (...|Drama|Romance|Sci-Fi|
|   1542|   100|4.4762063|  Brassed Off (1996)|Comedy|Drama|Romance|
|   1546|   100|4.4236083|  Schizopolis (1996)|              Comedy|
|   3330|   100|4.4175496|Splendor in the G...|       Drama|Romance|
|   7566|   100| 4.386091|        28 Up (1985)|         Documentary|
|     73|   100|4.3668256|Misérables, Les (...|           Drama|War|
+-------+------+---------+--------------------+--------------------+
```

Fig. 3. Recommendation for user

V. Conclusion

- KNN is simple to implement which do not derive any function from the training data but doesn't work properly in case of a large number of dimension. Also, the KNN model recommends a movie that is popular among similar users without allowing for personalization.
- ALS deduce proper recommendations of movies even from the large datasets and it recommends the movie based on personalization and taste of user's genres.

References

[1] Harrison, Onel. "Machine Learning Basics with the K-Nearest Neighbors Algorithm." Medium, Towards Data Science, 14 July 2019, towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.

[2] Gupta, Meenu, et al. "Movie Recommender System Using Collaborative Filtering." 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE, 2020.

[3] Li, Jung-Bin, et al. "Implementation of an Alternating Least Square Model Based Collaborative Filtering Movie Recommendation System on Hadoop and Spark Platforms." International Conference on Broadband and Wireless Computing, Communication and Applications. Springer, Cham, 2018.