IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol.-6, Issue-11, Nov 2018 E-ISSN: 2347-2693

Building a Movie Recommendation System using SVD algorithm

Asoke Nath [1*], Adityam Ghosh [2], Arion Mitra[3]

[1,2,3]Dept. of Computer Science, St. Xavier's College (Autonomous), Kolkata, India

**Corresponding author: asokejoy1@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 06/Nov/2018, Published: 30/Nov/2018

Abstract - Recommendation System predicts or recommends a set of products or items based upon the preference of the user. Recommender systems are utilized in variety of areas including movies, music, news, books search queries in general. This paper focuses on the design and development of a movie recommendation system using the SVD (Singular Value Decomposition) algorithm where we see that how sparse data are in real life situation and thereby predefined strategies such as collaborative or content-based filtering cannot be applied to these data for better results. Our objective is to reduce the sparsity of the data using dimensionality reduction by the SVD algorithm and hence recommend a list of movies based on the given input parameters.

Keywords: Recommendation System, SVD Decomposition, Netflix, Dimensionality reduction

I. INTRODUCTION

During the past few years the number of active internet users have increased manifold. With this, companies started investing more on collecting these ginormous user data and building automated system over it so that they can facilitate both their users and their business also. Whenever we go for e-shopping , at the bottom of the page ,we usually see these kind of recommendations,

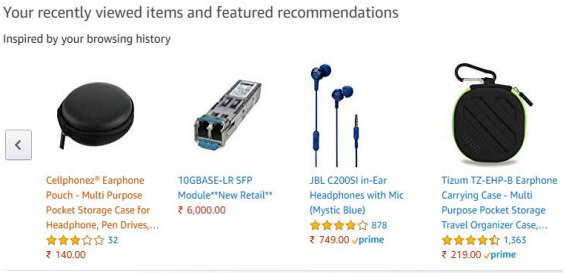

Fig. 1: Recommendation by amazon.in

Now the question is how do companies like Amazon, Flipkart, BigBasket etc. they come to know about the stuffs which their customer would prefer? or how can they recommend products like this when no-one has told them about their personal choices? Well, the answer to all of these question is, they continually collect their user's data , the amount of data that we have generated by surfing or spending time in their websites and selecting products and buying a few , from these data they tend to know our choices and based on that they build a system that recommends the best products to us and hence comes the advent of Recommendation System.

So what is a *Recommendation System*? A recommendation system or a recommender system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item.

Recommender System are utilized in a variety of areas including movies, music, news, books, research-articles, search-updates etc.

Recommendation system is broadly classified into three types:

- *Collaborative filtering*
- *Content-Based filtering*
- *Hybrid Recommender System*

A *Collaborative filtering system* is a widely used recommender system algorithm to predict the "rating" or "preference" a user would give to any particular product. This method is built on analyzing a large amount of information on user's behavior, activities or preferences and predicting what users will like based on similarity with the other users. This whole process is built either using a User-Item Ranking Matrix or Item-Item Ranking Matrix and then other algorithms are applied over it like k-nearest neighbor (KNN) and the Pearson-Correlation.

A *Content-Based filtering* approach is based on the description of the item and a profile of the user's preferences. In a content-based recommender system keywords are used to describe an item and a user profile is built to indicate the type of item the user likes. In other words these algorithms try to recommend items that are similar to those that a user liked in the past (or is examining in the present). Basically various candidate items are compared with items that were previously rated by the user and the best-matching items are recommended. This approach has its roots in information retrieval and information filtering research. Algorithms such as Bayesian Classifier, cluster analysis, decision trees and artificial

neural networks are used in order to estimate the ranks or the best recommended items

A *Hybrid Recommender System* is one in which both the methods of Collaborative Filtering and Content-Based Filtering are applied to recommend items to a user. It's like keeping one part in content-based algorithmic approach and the other in collaborative approach and at the end combining them to form a hybrid approach. Typical example being Netflix. The website makes recommendations by comparing the watching and searching habits of similar users (i.e., collaborative filtering) as well as by offering movies that share characteristics with films that a user has rated highly (content-based filtering).

This paper is organized as follows ,Section I contains the introduction, section II contains the methodology where we discuss the various steps of the proposed algorithm, section III contains the results obtained from our system while some input is given, and section IV concludes our paper with some future scopes.

II. METHODOLOGY

To build this system we collected sample data from MovieLens website. These data consists of 20 million movie ratings along with user-ids and genres and tags. Our very first step was to visualize the whole data-set.
After visualization, the next step was to build the user-item matrix which is a matrix of the ratings a user has given to each movie they have seen, the column here is the user Id and the row here is the movieId.

userId	1	2	3	4	5	6	7	8	9	10	...	202	203	204	205	206	207	208	209	210	211
movieId_r																					
1	0.0	0.0	4.0	0.0	0.0	5.0	0.0	4.0	0.0	4.0	...	5.0	0.0	5.0	4.0	0.0	0.0	4.0	4.0	0.0	0.0
2	3.5	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	4.0	0.0	0.0	0.0	3.0	3.0	5.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	3.5	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	3.0	0.0	0.0	0.0	3.0	0.0	0.0	...	0.0	0.0	0.0	4.5	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	5.0	3.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	3.5	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	4.0	0.0	0.0	0.0	4.0	0.0	0.0	...	3.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	5.0	0.0	4.0	0.0	0.0	4.0	...	3.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0

Fig. 2: Matrix form of sample data obtained

Upon building this user – item matrix we see that the user-item matrix is a sparse matrix. Now a sparse matrix is a matrix with maximum number of zero values in it. So we check the sparsity of the data

Sparsity Check

```
In [36]: n_users = df.userId.unique().shape[0]
         n_movies = df.movieId_r.unique().shape[0]
         print 'Users: %d | Movies: %d' %(n_users,n_movies)

         Users: 211 | Movies: 5258

In [37]: sparsity = round(1.0 - len(df)/float(n_users * n_movies),3)
         print 'The Sparsity level of the dataset is: ', str(sparsity * 100), '%'

         The Sparsity level of the dataset is:  97.5 %
```

Fig. 3: Sparsity Check of the matrix of sample data

As we can see that the sparsity level is quite high in the above matrix hence methods like Collaborative filtering or Content-based filtering on this dataset will not produce satisfactory results. So the suitable option is to first reduce the dimensionality of the matrix and then predict the correct ratings that can be given to a movie by a specific user and depending on the predicted rating we shall recommend that movie to the user. To do this we now need to apply the SVD algorithm. In SVD algorithm, a matrix is decomposed into three other matrices :

$$\boldsymbol{A = USV^T}$$

Where:

- A is a $m \times n$ matrix
- U is a $m \times n$ *orthogonal* matrix
- S is a $n \times n$ *diagonal matrix*
- V is a $n \times n$ orthogonal matrix

The steps to design the algorithm for this problem are given below:

- MAX_ROWS,MAX_COLUMNS = matrix.shape
- We define a function computeSVD( ) which takes in parameters the user-rating matrix and the dimensional reduction value K:

1. U,s,Vt= sparsesvd(urm,K)
2. dim = (len(s),len(s))
3. S = np.zeros(dim,dtype = np.float32)
Loop is set from 0 to len(s) and sqrt() values of s[i] is stored in S[i,i]
4. U=csc_matrix(np.transpose(U),dtype = np.float32)

5. S = csc_matrix(S,dtype = np.float32)
6. Vt = csc_matrix(Vt, dtype = np.float32)
7. return U , S ,Vt

The above function returns the reduced matrix(in decomposed form)

- Next We define another function called computeEstimatedRatings() which takes in parameters like the user-rating matrix, U, S, Vt , test_userid, K and a Boolean value test

*1. rightTerm = S*Vt*
2.estimatedRatings=np.zeros(shape=(MAX_Rows, MAX_Colum ns), dtype=np.float16)
3. for userTest in uTest:
*prod = U[userTest, :]*rightTerm*
we convert the vector to dense format in order to get the indices of the movies with the best estimated ratings
4. estimatedRatings[userTest,:]= prod.todense(recom= (estimatedRatings[userTest,:]).argsort()[:10]

5. return recom

This function returns the final recommended list of movies based on the input parameters.

III. RESULTS AND DISCUSSION

After we build the system as per the algorithm designed we then test it for some random user id. Let us suppose say we test the system for an arbitrary user having user id 4. To test this system we need to pass data to it as follows:

```
K = 25
matrix_modified = csc_matrix(matrix , dtype = np.float32)
U, S, Vt = computeSVD(matrix_modified,K)
uTest = [4]
print "Recommendation for User id: " , uTest[0]
print "Predicted MovieIds: "
uTest_recommended_items = computeEstimatedRatings(matrix_modified,U,S,Vt,uTest,K,True)
print uTest_recommended_items
print uTest_recommended_items.shape
for i in uTest_recommended_items:
    #print i
    print movies_data[movies_data['movieId'] == i]
```

Fig. 4: Input parameters for our system

When the given data is passed to the following function we get some predicted movies for that specific user id and then we print the best ones as follows:

```
Recommendation for User id:  4
Predicted MovieIds:
[222 902 306 412 890 612 591 503 328 502]
(10,)
     movieId                    title        genres
219      222  Circle of Friends (1995)  Drama|Romance
     movieId                          title        genres
885      902  Breakfast at Tiffany's (1961)  Drama|Romance
     movieId                                          title genres
303      306  Three Colors: Red (Trois couleurs: Rouge) (1994)  Drama
     movieId                           title genres
408      412  Age of Innocence, The (1993)  Drama
     movieId                              title    genres
873      890  Baton Rouge (Bâton rouge) (1988)  Thriller
     movieId                       title  genres
606      612  Pallbearer, The (1996)  Comedy
     movieId                        title                genres
585      591  Tough and Deadly (1995)  Action|Drama|Thriller
     movieId                   title genres
499      503  New Age, The (1994)  Drama
     movieId                                             title  \
324      328  Tales from the Crypt Presents: Demon Knight (1...
```

Fig. 5: Output from our system

The top 10 predicted or recommended movies are shown for the "user id 4" in descending order sequence where the first movie is of the highest rating and the last one is of the lowest rating as per the user's choice. This prediction of the movies based on the preferred genres of the user id 4 was computed from the data-set that was collected initially from the movie-lens website. Thus our system succeeded in reducing the sparsity of the matrix obtained and then apply SVD to finally obtain the predicted list of movies based upon the user's preference.

IV. CONCLUSION AND FUTURE SCOPE

This algorithm is mechanized to recommend the best 10 movies to the user based on their choices although this approach is better for very sparse data but it has got some limitations in the real life cases. For example suppose a new movie is released and there is no rating data available for that movie from any user. Then this algorithm will not recommend the movie to any user as this system works and evaluates data already prepared in the past. In that case this algorithm needs to be incorporated along with hybrid recommender systems to increase the efficiency.

REFERENCES

[1] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=<http://dx.doi.org/10.1145/2827872>

[2] B.M. Sarwar, et, "Application of Dimensionality Reduction in Recommender System—A Case Study," *Proc. KDD Workshop on Web Mining for e-Commerce: Challenges and Opportunities* (WebKDD), ACM Press, 2000.

[4] Bobadilla, J., Ortega, F., Hernando, A., Gutierrez, A.: Recommender systems survey. Knowledge-Based Systems 46(0), 109–132 (2013)

[5]To view full code visit the following link:
https://github.com/lucifermorningstar1305/machine_learning/blob/master/Codelogicx/Codelogicx/RecommenderSystemFinal. ipynb

Author's Profile

Dr. Asoke Nath currently attached with t St. Xavier's College, Kolkata. Apart from his teaching assignment he is actively involved in doing research (i) Cryptography and Network Security, (ii) Visual Cryptography, (iii) Steganography, (iv)Big data analytics, (v) Data science, (vi)Green Computing, (vii)Mathematical modeling of Social networks, (viii)MOOCs. He has published 237 research publications in Journals and Conference proceedings.

Mr. Adityam Ghosh is a **final year BSC(Computer Science)** student at **St.Xavier's College,** Kolkata, India. Throughout his college and school days he has exercised himself with various experimentation on different computer languages like **Python, C/C++, JAVA, C#, Shell Programming, R programming, Android Apps, Machine Learning and AI, SQL**.

Mr. Arion Mitra is currently in final year of Bachelor's degree in Computer Science from St. Xavier's College, Kolkata. He is having interests in the field of **Artificial Intelligence, Neural Networks and cryptography**.

