PES University, Bangalore

(Established under Karnataka Act No. 16 of 2013)

MAY 2020: IN SEMESTER ASSESSMENT (ISA) B.TECH. IV SEMESTER

UE18MA251- LINEAR ALGEBRA

MINI PROJECT REPORT

ON SIMPLE MOVIE RECOMMENDER USING SVD

Submitted by

1. NAVYA P HEGDE PES2201800017
2. NAYANA C S REDDY PES2201800510
3. LINCY CATHERINE SANNY PES2201800273

Branch & Section : CSE 4 ‘E’

PROJECT EVALUATION

(For Official Use Only)

Sl. No.	Parameter	Max Marks	Marks Awarded
1	Background & Framing of the problem	4	
2	Approach and Solution	4	
3	References	4	
4	Clarity of the concepts & Creativity	4	
5	Choice of examples and understanding of the topic	4	
6	Presentation of the work	5	
	Total	25	

Name of the Course Instructor : Mrs. SathyaVani N L

Signature of the Course Instructor :

TABLE OF CONTENTS

ABSTRACT

Data analysis has become a very important area for both companies and researchers as a consequence of the technological developments in recent years. Companies are trying to increase their profit by analyzing the existing data about their customers and making decisions for the future according to the results of these analyses. Parallel to the need of companies, researchers are investigating different methodologies to analyze data more accurately with high performance. Recommender systems are one of the most popular and widespread data analysis tool. A recommender system applies knowledge discovery techniques to the existing data and makes personalized product recommendations during live customer interaction. However, the huge growth of customers and products especially on the internet, poses some challenges for recommender systems, producing high quality recommendations and performing millions of recommendations per second. In order to improve the performance of recommender systems, researchers have proposed many different methods. Singular Value Decomposition (SVD) technique based on dimension reduction is one of the methods which produces high quality recommendations, but has to undergo very expensive matrix calculations. In this project, we propose and experimentally validate some contributions to SVD technique which are based on the user and the item categorization. Besides, we adopt tags to classical 2D (User-Item) SVD technique and report the results of experiments. Results are promising to make more accurate and scalable recommender systems.

1. Introduction:

Movies are the most complex and powerful art form in the present world. It can help us to understand our own lives, the lives of those around us and even how our society and culture operates. It is the combination of technology, business, entertainment and aesthetics. It has an important role in the present-day world. In this fast-paced society where the need for entertainment is growing rapidly there is also a need to understand the user's likes and dislikes. A movie recommendation system is personalized to a specific person, based on what he has expressed his interest in. Our project is based on "**SIMPLE MOVIE RECOMMENDER USING SVD**".

Recommender systems apply knowledge discovery techniques to the problem of making personalized product recommendations during live customer interaction. Recommender systems have shown a lot of awareness in the past decade. Due to their great business value, recommender systems have also been successfully deployed in business, such as product recommendation at Flipkart, HomeShop18, and music recommendation at Last.fm, Pandora, and movie recommendation at Flixstreet, MovieLens, and Jinni. But in reality, when we are asking our friends or looking for opinions, reviews for recommendations of Mobile or heart touching music, movies, electronic gadgets, restaurant, book, games, software Apps, we actually use social information for recommendation. Singular Value Decomposition (SVD)-based recommendation algorithms can quickly produce high quality recommendations.

In linear algebra, the Singular Value Decomposition (SVD) of a matrix is a factorization of that matrix into three matrices. It has some interesting algebraic properties and conveys important geometrical and theoretical insights about linear transformations. This classical method from linear algebra is getting popular in the field of data science and machine learning. This popularity is because of its application in developing recommender systems. There are a lot of online user-centric applications such as video players, music players, e-commerce applications, etc., where users are recommended with further items to engage with. The assumption is that people with similar movie tastes are most likely to give similar movie ratings. So, if I'm looking for a new movie and I've watched *The Matrix*, this method will recommend movies that have a similar rating pattern to *The Matrix* across a set of users. Finding and recommending many suitable items that would be liked and selected by users is always a challenge. There are many techniques used for this task and SVD is one of those techniques.

2. Review of Literature:

The essence of SVD is that it decomposes a matrix of any shape into a product of 3 matrices $A=USV^T$.

The result of the decomposition leaves us with an ordered matrix of singular values which encompass the variance associated with every direction. We assume that larger variances mean less redundancy and less correlation and encode more structure about the data. This allows us to use a representative subset of user rating directions or principal components to recommend movies.

A recommender system is an intelligent system that predicts the rating and preferences of users about products. The primary application of recommender systems is finding a relationship between user and products in order to maximize the user-product engagement. The major application of recommender systems is to suggest related video or music for generating a playlist for the user, when they are engaged with a related item.

Mustansar Ali Ghazanfar and Adam Prugel-Bennett in The Advantage of Careful Imputation Sources in Sparse Data-Environment of Recommender Systems: Generating Improved SVD-based Recommendations states: -An example of the recommender system is the Amazon recommender engine, which can filter through millions of available items based on the preferences or past browsing behavior of a user and can make personal recommendations. Some other well-known examples are YouTube (www.youtube.com) video recommender service and MovieLens (www.movielens.com) movie recommender system, which recommend videos and movies based on the person’s opinions. Recommender systems helps E-commerce sites in increasing their sales by making useful recommendation—items a customer/user would most likely to consume. In these systems, the history of user’s interactions with the system is stored, which shape user’s preferences. The history of the user can be gathered by explicit feedback, where the user rates some items in some scale, or by implicit feedback, where the user’s interaction with the system is observed—for instance, if a user purchases an item then this is a sign that they like that item.

There are two main types of recommender systems: collaborative filtering (CF) and content-based filtering recommender systems.

Collaborative filtering recommender systems recommend items by taking into account the taste (in terms of preferences of items) of users, under the assumption that users will be interested in items that users similar to them have rated highly. Examples of these systems include the GroupLens system, and Ringo (www.ringo.com). Collaborative filtering can be classified into two sub-categories: memory-based CF and model-based CF. Memory-based approaches make a prediction by taking into account the entire collection of previous rated items by a user. Examples of these systems include GroupLens recommender systems. Model based approaches use rating patterns of users in the training set, group users into different classes, and use ratings of predefined classes to generate recommendation for an active user1 on a target item2. Examples of these systems include item-based CF, Singular Value Decomposition (SVD) based models , bayesian networks, clustering models, and Kernelmapping recommender. [1]

Content-based filtering recommender systems recommend items based on the content information of an item, under the assumption that users will like similar items to the ones they liked before. In these systems, an item of interest is defined by its associated features, for instance, NewsWeeder , a newsgroup filtering system uses the words of text as features. The textual description of items is used to build item profiles. User profiles can be constructed by building a model of the user's preferences using the descriptions and types of the items that a user is interested in, or a history of user's interactions with the system is stored (e.g. user purchase history, types of items they purchased together, etc.). The history of the user can be gathered by explicit feedback or implicit feedback. Furthermore, hybrid recommender systems have been proposed which combine individual recommender systems to avoid certain limitations of individual recommender systems.

Rajeev Kumar, B. K. Verma and Shyam Sunder Rastogi in Social Popularity based SVD++ Recommender System states: -

SVD-based approach can generate results that were much better than a traditional collaborative filtering algorithm most of the time when tested on a MovieLens dataset. But there are some serious limitations when we apply SVD-based approach for recommending which make its less suitable for large scale deployment in e-commerce system. The matrix factorization step is computationally very expensive because it takes a lot of memory and time to factorize a matrix. This is a major problem towards achieving a high scalability while producing good predictive accuracy. In real world e-commerce application, a large number of customers only buy or rate a very small percentage of products, which is real problem. Dimensionality reduction in recommender system is used due to this problem. That will help to improve the precision of recommendations and reduce the complexity of real time computations.

The weakness of Collaborative filtering-based approach for large, sparse databases motivated us to investigate alternative recommender system algorithms. Latent Semantic Indexing (LSI) is used to reduce the dimensionality of user-item ratings matrix. LSI uses singular value decomposition (SVD) as its fundamental dimensionality reduction algorithm, maps well into the collaborative filtering based recommender system challenges. However, SVD cannot be applied to explicit rating in the collaborative filtering based approach because user does not rate most of product so user-item rating matrix have lot of missing values. Furthermore, only few known entries may cause of overfitting. Recent work says that we can fills in missing ratings values and make user-item rating matrix dense. But it is more expensive as compared to other method. Therefore, more recent works recommended that modeling with only the observed ratings, while avoiding over fitting through sufficient regularized model. [2]

3. Report on Present Investigation:

The Singular Value Decomposition (SVD), a method in linear algebra has been generally used as a dimensionality reduction technique in machine learning. SVD is a matrix factorization technique, which reduces the number of features of a dataset by reducing the space dimension from N-dimension to K-dimension (where K<N). In the context of the recommender system, the SVD is used as a collaborative filtering technique. It uses a matrix structure where each row represents a user, and each column represents an item. The elements of this matrix are the ratings that are given to items by users.

The factorization of this matrix is done by the singular value decomposition. It finds factors of matrices from the factorization of a high-level (user-item-rating) matrix. The singular value decomposition is a method of decomposing a matrix into three other matrices as given below:

$A=USV^T$.

Where A is a m x n utility matrix, U is a m x r orthogonal left singular matrix, which represents the relationship between users and latent factors, S is a r x r diagonal matrix, which describes the strength of each latent factor and V is a r x n diagonal right singular matrix, which indicates the similarity between items and latent factors. The latent factors here are the characteristics of the items, for example, the genre of music. The SVD decreases the dimension of the utility matrix A by extracting its latent factors. It maps each user and each item into a r-dimensional latent space. This mapping facilitates a clear representation of relationships between users and items.

Let each item be represented by a vector x_i and each user is represented by a vector y_u. The expected rating by a user on an item $\hat{r}_{ui}$ can be given as: $\hat{r}_{ui} = x_i^T.y_u$

Here, $\hat{r}_{ui}$ is a form of factorisation in singular value decomposition. The x_i and y_u can be obtained in a manner that the square error difference between their dot product and the expected rating in the user-item matrix is minimum. It can be expressed as:

$$Min(x, y) \sum_{(u,\, i)\in K} (r_{ui} - x_i^T.y_u)^2$$

In order to let the model, generalise well and not overfit the training data, a regularisation term is added as a penalty to the above formula.

$$Min(x,\ y) \sum_{(u,\ i) \in K} (r_{ui} - x_i^T . y_u)^2 + \lambda(||x_i||^2 + ||y_u||^2)$$

In order to reduce the error between the value predicted by the model and the actual value, the algorithm uses a bias term. Let for a user-item pair (u, i), μ is the average rating of all items, b_i is the average rating of item i minus μ and b_u is the average rating given by user u minus μ, the final equation after adding the regularisation term and bias can be given as:

$$Min(x,\ y,\ b_i,\ b_u) \sum_{(u,\ i) \in K} (r_{ui} - x_i^T . y_u - \mu - b_i - b_u)^2 + \lambda(||x_i||^2 + ||y_u||^2 + b_i^2 + b_u^2)$$

The above equation is the main component of the algorithm which works for singular value decomposition-based recommendation system.

CODE

Below is an implementation of singular value decomposition (SVD) based on collaborative filtering in the task of movie recommendation. For simplicity, MovieLens 1M Dataset has been used. This dataset has been chosen because it does not require any preprocessing.

The ongoing researches on this topic are aided with the Python libraries such as

- numpy as ‘np’
- pandas as ‘pd’.

The steps used to achieve the above-mentioned system are:

1) Read the files with pandas:

```
data = pd.io.parsers.read_csv('data/ratings.dat',  names=['user_id', 'movie_id', 'rating',

'time'], engine='python', delimiter='::')

movie_data = pd.io.parsers.read_csv('data/movies.dat',names=['movie_id', 'title', 'genre'],
```

engine='python', delimiter='::')

2) Create the ratings matrix of shape (m×um×u) with rows as movies and columns as users.

```
ratings_mat = np.ndarray(shape=(np.max(data.movie_id.values),
np.max(data.user_id.values)), dtype=np.uint8)
ratings_mat[data.movie_id.values-1, data.user_id.values-1] = data.rating.values
```

3) Normalise matrix (subtract mean off).

```
normalised_mat = ratings_mat - np.asarray([(np.mean(ratings_mat, 1))]).
```

4) Compute SVD

```
A = normalised_mat.T / np.sqrt(ratings_mat.shape[0] - 1)
U, S, V = np.linalg.svd(A)
```

5) Calculate cosine similarity, sort by most similar and return the top N.

```
def top_cosine_similarity(data, movie_id, top_n=10):
    index = movie_id - 1 # Movie id starts from 1
    movie_row = data[index, :]
    magnitude = np.sqrt(np.einsum('ij, ij -> i', data, data))
    similarity = np.dot(movie_row, data.T) / (magnitude[index] * magnitude)
    sort_indexes = np.argsort(-similarity)
    return sort_indexes[:top_n]
```

```
# Helper function to print top N similar movies
```

```
def print_similar_movies(movie_data, movie_id, top_indexes):
    print('Recommendations for {0}: \n'.format(
    movie_data[movie_data.movie_id == movie_id].title.values[0]))
    for id in top_indexes + 1:
    print(movie_data[movie_data.movie_id == id].title.values[0])
```

6) Select k principal components to represent the movies, a movie id to find recommendations and print the top n results.

```
k = 50
movie_id = 1 # Grab an id from movies.dat
top_n = 10
sliced = V.T[:, :k] # representative data
indexes = top_cosine_similarity(sliced, movie_id, top_n)
print_similar_movies(movie_data, movie_id, indexes)
```

7) Print the top N similar movies.

```
print_similar_movies(movie_data, movie_id, indexes)
```

<u>The output for the following code is as follows:</u>

Recommendations for Toy Story (1995):

Toy Story (1995)
Toy Story 2 (1999)
Babe (1995)
Bug's Life, A (1998)

Pleasantville (1998)

Babe: Pig in the City (1998)

Aladdin (1992)

Stuart Little (1999)

Secret Garden, The (1993)

Tarzan (1999)

4. Results and Discussion:

A few points to be noted include- SVD is faster when compared to co-variance matrix and singular values from SVD are sorted (we have to sort the eigenvalues in ascending order).

A few other points to be noted include:

- Approximation of missing values in the sparse user-item rating matrix has an important role in SVD based recommendations.
- Content-based filtering are not very accurate as compared to the collaborative filtering ones in the recommender system domain.
- Although content-based filtering has successfully been applied to text categorization and it gives accurate results.
- Text categorization and recommender system problems are quite different from each other due to:

  →a user rates the same item differently under different context and the reason of rating might be complex. Similarly, the positive feedback given by a user, e.g. purchased an item is dependent on the context; for example, a user might purchase an item as a gift, hence we cannot predict that they will purchase other similar items.

  → The user feedback in a recommender system is noisy, the observations such as customer not buying an item, or not watching a movie does not necessarily mean that the user is not interested in that item or movie. It can be that the user likes that item or movie but has not purchased or watched it.

  → the evaluation criteria for both is different- recommender system usually provides a list of top items a user would like to consume, whereas text categorization classify a given document to set of pre-defined categorization. Furthermore, in text categorization a document belongs to a single or a very few categories, whereas a user in recommender system might be interested in a large number of different items.

  →Fourth, a user might change their taste over time and this temporal change in profile is not shared by the text categorization tasks. Making accurate recommendation given the noisy input is different and more difficult as compared to the text categorization task.

5. Summary and Conclusion:

Recommender systems play an important role in identifying the interesting items for users and try to solve the problem of information overload. There has been some work, in the literature to overcome the scalability problem of recommender system using SVD. Scalability and accuracy problems can be eliminated by using a suitable imputation source, as a pre-processing step, with SVD. We have considered the two-dimensional Users × Items space, by recommending items to users based on the information only about users and items. It has been claimed that taking the additional context information (such as time, place, and the company of a user) into account, either by extending the user-item rating matrix into multiple dimensions or using reduction-based recommendation approach, might increase the performance of the recommender systems.

To summarize, the study was conducted on the recommendations which one individual gets while using Netflix, Amazon prime or any other applications using the method of *singular value decomposition*, an application of *linear algebra*. Hence, we can use *singular value decomposition* method in simple movie recommender system.

In future we need to identify why SVD works fine for some recommender applications, and not so fine for other applications. SVD can be implemented in different ways to solve recommender system problems. SVD can be used to create low dimensional visualizations of user-item rating matrix or SVD can help to identify important items or products that will help to recommend right items or products. The use of the implicit sentiment analysis within the CF in social web will be main concern in future.

6. Bibliography:

WEBSITES

1.https://alyssaq.github.io/2015/20150426-simple-movie-recommender-using-svd/

2.https://analyticsindiamag.com/singular-value-decomposition-svd-application-recommender-system/

3.https://www.youtube.com/watch?v=ZspR5PZemcs&list=PL35hOxiWAs3TvdJK0LJEOfACn1Lha0Xll&index=6&t=0s

4. https://www.youtube.com/watch?v=3ecNC-So0r4&list=PL35hOxiWAs3TvdJK0LJEOfACn1Lha0Xll&index=8

JOURNALS

[1] Mustansar Ali Ghazanfar and Adam Prugel-Bennett, "The Advantage of Careful Imputation Sources in Sparse Data-Environment of Recommender Systems: Generating Improved SVD-based Recommendations states"

[2] Rajeev Kumar, B. K. Verma and Shyam Sunder Rastogi, Social Popularity based SVD++ Recommender System states, International Journal of Computer Applications (0975 – 8887) Volume 87 – No.14, February 2014