BUILDING A MOVIE RECOMMENDATION SYSTEM USING NON NEGATIVE REGULARIZED MATRIX FACTORIZATION ON MOVIE LENS DATASET

BY

SHADMAN SHAFIN
151-15-5433

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Raja Tariqul Hasan Tusher
Lecturer
Department of CSE
Daffodil International University

Co-Supervised By

Aniruddha Rakshit
Lecturer
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH

December 2018

APPROVAL

This Project titled **"Building a movie recommendation system using Non negative regularized matrix factorization on movie-lens dataset"**, submitted by Shadman Shafin to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering (BSc) and approved as to its style and contents.

BOARD OF EXAMINERS

____

Dr. Syed Akhter Hossain — **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

Dr. Sheak Rashed Haider Noori — **Internal Examiner**
Assistant Professor and Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

MD Zahid Hasan — **Internal Examiner**
Assistant Professor and Coordinator of MIS
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

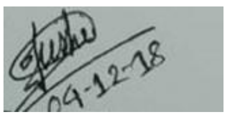
____

Dr.Muhammad Shorif Uddin — **External Examiner**
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

I hereby declare that, this project has been done by us under the supervision of **Raja Tariqul Hasan Tusher , Lecturer, Department of CSE** Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Raja Tariqul Hasan Tusher
Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:

Aniruddha Rakshit
Lecturer
Department of CSE
Daffodil International University

Submitted by:

Shadman Shafin
151-15-5433
Department of CSE

ACKNOWLEDGEMENT

First I express my heartiest thanks and gratefulness to almighty Allah for His divine blessing makes me possible to complete the final year project successfully.

I really grateful and wish our profound my indebtedness to **Raja Tariqul Hasan Tusher**, **Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge and keen interest of my supervisor in the active learning model design influenced to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Syed Akhter Hossain, Head, Department of CSE**, for his kind help to finish my project and also to other faculty members and the staffs of CSE department of Daffodil International University.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

ABSTRACT

Many E-commerce, Social Network, Entertainment sites uses recommendation system frequently to increase their revenue. There are various methods of recommending an item to a specific user. Collaborative filtering, content-based filtering are very popular. Matrix factorization is one of the most widely used method to predict the ratings of an item whether to recommend it to a user or not. But traditional matrix factorization only considers the user-item ratings matrix. It doesn't consider any extra information about any user or any item. Sometimes the information associates with the item or user is very important to predict a rating of item by a particular user. To do so, I propose a model which will consider the textual information about items and users provided with the dataset. Thus it will be helpful to design a better and more accurate recommendation system which will give almost accurate prediction of a missing rating.

TABLE OF CONTENTS

CONTENS **PAGE**

CONTENS	PAGE
Board of examiners	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
Chapter	
CHAPTER 1: Introduction	**1-3**
1.1 Introduction	1
1.2 Objectives of the project	1
1.3 Motivation	2
1.4 Rationale of the Study	2
1.5 Research Questions	2
1.6 Expected Output	3
1.7 Report Layout	3
Chapter 2: Background	**4-7**
2.1 Introduction	4
2.2 Related Works	4-6
2.3 Research Summary	7
2.4 Challenges	7
Chapter 3: Research Methodology	**8-17**
3.1 Introduction	8
3.2 Research Subject and Instrumentation	8
3.3 Data Collection Procedure	8-12
3.4 Theoretical Consideration	12-17
3.5 Implementation Requirements	17

Chapter 4: Experimental Results and Discussion **18-23**

4.1 Introduction 18

4.2 Dataset & Data format 18

4.3 Experimental Results 19

4.4 Statistical Graphs 20-23

4.5 Summary 23

Chapter 5: Summary, Conclusion, Recommendation and Implication for Future Research **24**

5.1 Summary of the Study 24

5.2 Implication for Further Study 24

5.3 Conclusions 24

References 25-26

List of Figures

Test and train error	20
Impact of alpha	21
Impact of beta	22
Impact of gama	23

List of Tables

Data Statistics 18

Format of Dataset 20

CHAPTER 1

INTRODUCTION

1.1 Introduction

Recommendation engine is applied by many social sites, online applications and ecommerce sites and suggests music, film, articles or products. Research says that 10%20% revenue is increased by recommendation. This is because, user does not know what to choose from a huge collection of items. If some items are suggested, then he can easily pick one up. Thus, items or products get noticed by users and their sale or rating increases.

Most popular techniques for recommendation engine are – Content based, Collaborative filtering and their hybrid. Non-negative matrix factorization is another technique for recommending an item. This technique considers the user-item ratings matrix and decompose it into two matrices so that the multiplication of those matrix results in the previous matrix. It does so by considering the user latent features and item latent features. But it doesn't take into account the actual attribute of user and items. The information about an item or a user remains unnecessary in this technique to recommend an item. But in practical I know that the nature of an item and a user is a vital issue for recommending an item to a user. A model is necessary to design which will take into account user-item attribute.

1.2 Objectives of the project:

The main objective of this thesis is to develop a recommendation engine which will increase the efficiency of rating prediction of an item given by a user. For this I use a modified version of non-negative matrix factorization. This modified version will take into account the textual information of user or item. Traditional matrix factorization only considers the user-item ratings matrix. The attributes of a user or an item is ignored. But this attribute information is very much useful in predicting ratings. By observing these attribute values, I can identify the nature of a user or an item. But these facilities are absent in traditional matrix factorization. It only considers the latent features. To give importance to both latent feature and actual attribute a modified version of matrix factorization is proposed.

1.3 Motivation

Recommendation systems help users to find and select items (example: books, movies, restaurants& many other related items) from the massive number existing on the web or in other electronic information sources. Given a huge set of items and a description of the user's needs, they present to the user a small set of the items that are well suited to the description. Recent work in recommendation systems contains intelligent aides for predicting and choosing web sites, news, TV listings and other info.

Besides this, we see that today's world is so much focusing on recommendation system. Users expect everything that the better things will be recommended to them by the system. To make a system to be recommendation capable must have the ability to take decision by itself. To take decision by itself must need to have the data mining capability.

These made me interested to do such kind of research based work. My work is fully related with machine learning techniques and has some data mining procedures too.

1.4 Rationale of the Study

In this study, I have worked for modification of Matrix Factorization to model a recommendation system as well as recommend items for users. That means, a full recommendation model is my scope of study. But I have not considered dynamically added users or items. I have worked with fixed and previously given items and users. Also, I did not consider local or global optima for my error testing due to computation complexity.

1.5 Research Question

- Will the recommender system be able to predict the ratings correctly?
- Will it be able to give importance to both latent feature and actual attribute?
- Can the Machine Learning process correctly predict Ratings?

1.6 Expected Output

Expected outcome of this research based project is to predict movies for users correctly with respect to the built model of trained dataset.

1.7 Report Layout

The report will be followed as follows

Chapter 1 provides the summary of this research based project. Introductory discussion is the key term of this first chapter. Apart from, what motivated us to do such a research based project is explained well in this chapter to. The most important part of this chapter is the Rationale of the Study. Then, what are the research questions and what is the expected outcome is discussed in the last section of this chapter.

Chapter 2 covers the related works and methods regarding to my project and very last the root obstacles or challenges of this research are explained.

Chapter 3 covers the theoretical discussion on this research work. To discuss the theoretical part of the research, this chapter elaborates the current and my proposed methods of this work And in the last section of this chapter

Chapter 4 is related with the outcome of the whole research and the project. Experimental results are presented in this chapter to make realize the project.

Chapter 5 is based on conclusion topics of the project. This chapter is responsible to show the whole project report adhering to recommendation. The chapter is closed by showing the future scope of others who want to work in this field.

CHAPTER 2
BACKGROUND

2.1 Introduction

This chapter reflects the related works that already done by some researchers in the previous time in this field. Besides, giving a clear explanation of this, this chapter will show what the limitations of these works were and lastly, this chapter describes scope of our research as well as the challenges of it.

2.2 Related Works

Very few works on this field has accomplished by this time though in the present time, working on this field is increasing day by day. There are enough resources for recommender system.

2.2.1 Collaborative Filtering (CF)

Collaborative filtering (CF) is a technique used by recommendation systems. Collaborative filtering is a technique of making automatic predictions about the interests of a user by collecting preferences or likings information from many users. The underlying theory of the collaborative filtering approach is that if a person A has the same view as a person B on an issue, A is more likely to have B's view on a different issue than that of a randomly selected person. For example, a collaborative filtering recommendation system for television tastes could make assumptions about which movie a user should like given a partial list of that user's likes or dislikes. These assumptions are specific to the user, but use information collected from many users. This differs from the simpler approach of giving an average score for each item of interest, for example based on its number of votes.

2.2.2 Content based Filtering

Another common approach when designing recommender systems is content based filtering. Content based filtering methods are based on a description of the item and a profile of the user’s preferences. In a content-based recommender, keywords are used to define the items and a user profile is built to indicate the type of item this user likes. In other words, these algorithms tries to recommend items that are similar to those, that a user liked in the past or present. In particular, various items are compared with items previously rated by the user and the best-matching items are recommended. This approach has its origins in information retrieval and information filtering research.

To abstract the features of items in the system, an algorithm can be applied. A commonly used algorithm is the term frequency–inverse document frequency (tf-idf) representation.

These methods use an item profile (a set of discrete characteristics and features) describing the item within the system. The system creates a content based profile of users based on a weighted vector of item features. The weights signify the importance of each feature to the user and can be calculated from individually rated content vectors using a variety of techniques. Simple methods use the average values of the rated item vector while other sophisticated methods use machine learning techniques such as Bayesian Classifier, cluster analysis, decision trees, and artificial neural networks in order to estimate the probability of that the user is going to like the items.

By getting direct feedback from a user in the way of a like or dislike button, can be used to assign higher or lower weights on the importance of certain attributes. (Roccio Classification techniques).

An issue with content based filtering is whether the system is able to learn user preferences from users' actions regarding one contented source and use them across other content types. When the system is limited to recommending content of the same type as the user is already using, the rate from the recommendation system is significantly less than when other content types from other services can be recommended. For example, recommending news articles based on browsing of news is useful, but would be much more useful when music, videos, products, discussions etc. from different services can be recommended based on news browsing.

2.2.3 Recommendation Methods

Collaborative filtering(CF) is one of the most popular approaches taking advantage of user rating history to predict users' interests. CF method mainly involves user-based CF and item-based CF. The basic knowledge of user based CF is to recommend the interesting items to the active user according to the interests of the other users who have close relationships. Also, item-based CF tries to recommend the active user the potentially interested items having close similarities with the historical items that the active user likes. Although the wide adoption in many real applications, e.g., Amazon, the effect of CF is sharply decreased for new users and items. This is partially because when the rating matrix is very sparse, for new users and items, it is extremely difficult to get the relationships between users and items. This limitation partially motivates us to consider other relations between users and between items; if I can get the users' or items' relationships no matter whether I have ample rating data, it may greatly improve the effectiveness of recommendations. The semantic relations between items discussed in this paper can overcome the limitation.

As per the most accurate single models for CF, Matrix Factorization(MF) is a latent factor model which effectively estimates latent factor vectors of users and items. Specifically, MF approach tries to decompose the rating matrix to user latent matrix and item latent matrix. Then the estimated rating is predicted by the multiplication of the two decomposed matrices. With the advent of social network, many researchers started to analyze social RS and proposed various models integrating social networks.

Social friendship is an outstanding obvious factor to improve the effectiveness of recommendation system. However, not every web site has social or trust mechanisms. This explicit social gap strongly motivates us to explore other valuable relations between items and between users to improve recommendation qualities. Indeed, the discussed semantic relations between items in this paper are helpful for making reasonable recommendations over sparse rating matrix.

Content based techniques are another effective methods which recommend relevant items to users according to users' personal interests. Usually, attributes and free texts are the two kinds of content in RS. Content based methods often assume that item's attributes are independent which is not always held in reality. Actually, several research outcomes such as have been proposed to handle the challenging issues. However, these papers still did not consider the text information as complementary content, which greatly encourages us to consider textual semantic relations between items to improve recommendation algorithms.

2.3 Research Summary

In this section I have introduced with many terms related to this thesis. I have come to know about data scarcity, I know about collaborative and content-based filtering. At the very last I have seen some related works on this topic done by others earlier. This information will help us in the next section of the paper.

2.4 Challenges

The main challenges of this work are dealing with the huge movie lens datasets.

Privacy: Feeding private/personal information to the recommender systems results in better recommendation services but may lead to problems of data privacy and security.

Sparsity: The obtainability of huge size of data about items the file and the hesitation of users to rate items to make an isolated profile matrix leading to less correct prediction.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This chapter mainly deals with the theoretical knowledge of the research work. It will give the clear understanding of the concept of work. To make it more clear, very first, Research Subject and Instrumentation is explained shortly. Then we know that in the data mining or machine learning process. For this reason, data collection process is described in this section. The chapter is being closed by giving the explanation of our project's statistical theories and besides, giving the clear concept of the implementation requirements.

3.2 Research Subject and Instrumentation

We mean by research subject is that research area that is being studied and researched for clear understandings. Not only for clear understanding, but also research subject is responsible for giving the right knowledge of various research parameters. On the other hand, Instrumentation refers to the required instruments or tools that are used by the researchers.

3.2 Data Collection Procedure

To research on specific field, the fast and foremost thing is the Data. Data is, actually, considered as the heart of the machine learning process. Movie-Lens data sets are widely used datasets for Recommendation Systems. I have used 100k dataset from movie lens website. I have edited the dataset for my project.

3.3 Theoretical Consideration

3.3.1 Gradient Decent & Stochastic Gradient Decent:

Gradient descent is an algorithm which reduces functions. In s given function defined by a set of parameters, gradient descent starts with an initial set of parameter values and iteratively moves toward a set of parameter values that minimizes the function. This iterative minimization is achieved using calculus, taking steps in the negative direction of the function gradient. Gradient descent is best used when the parameters cannot be calculated logically (using linear algebra) and must be searched for by an optimization algorithm.

Stochastic gradient descent (SGD), also known as incremental gradient descent. It is a stochastic approximation of the gradient descent optimization and iterative method for reducing an objective function that is written as a sum of differentiable functions. SGD tries to find minima or maxima by iteration. Since Gradient descent is slow to run on very large datasets. As one iteration of the gradient descent algorithm requires a prediction for each instance in the training dataset, it can be time-consuming when we have many millions of instances.

In SGD, the update to the constants is performed for each training instance, rather than at the end of the batch of instances. The first step of the procedure requires that the order of the training dataset is randomized. This is to mix up the order that updates are made to the coefficients. Because the coefficients are updated after every training instance, the updates will be noisy jumping all over the place, and so will the corresponding cost function. By mixing up the order for the updates to the coefficients, it harnesses this random walk and avoids it getting distracted or stuck. The learning can be much faster with stochastic gradient descent for very large training datasets and often you only need a small number of passes through the dataset to reach a good or good enough set of coefficients.

Explanation:

Suppose, an objective function $f(x) = \sum_{i=1}^{n} f_i(x)$. I have to find optimum x that minimizes (x). When used to minimize the above function, method would perform this iteration:

$$x = x - \alpha \nabla f(x)$$

When the algorithm sweeps through the training set, it performs the above update for each training dataset. Several passes can be made over the training set until the algorithm converges. After this is done, the data can be shuffled for each pass to stop cycles. Typical operations may use an adaptive learning rate so that the algorithm converges.

Pseudo code:

1. Choose an initial vector of parameter x and learning rate .
2. Recap until an approximate minimum is obtained:

I. Randomly shuffle examples in the training set. II. For $i = 1\ to\ n$, do:

$$x = x - \alpha \nabla f(x)$$

When the learning-rate α decrease with an appropriate rate, and subject to relatively mild assumptions, stochastic gradient descent joins almost surely to a global minimum when the objective function is convex or pseudo-convex and otherwise converges almost surely to a local minimum.

3.3.2 Matrix Factorization (MF) :

Assume now I have 5 users and 10 items, and ratings are integers ranging from 1 to 5. So, the task of predicting the missing ratings can be considered as filling the blanks (the hyphens in the matrix) such that the values would be reliable with the existing ratings in the matrix. The perception behind using matrix factorization to solve this problem is that there should be some latent features that determine how a user rates an item. For example, two users would give high ratings to a certain movie if they both like the actors or actresses in the movie, or if the movie is an action movie, which is a genre preferred by both users. So, if we can discover these latent features, we should be able to predict a rating with respect to a certain user and a certain item, because the features associated with the user should match with the features associated with the item. In trying to discover the different features, I also make the assumption that the number of latent features would be smaller than the number of users and the number of items.

MF decompose the rating matrix R into P and Q such that:

R' = P*Q^T where P = user*K and Q = movie*K K must be less than number of user and movie.

To obtain optimum P and Q, I have to minimize the error defined as:

$$e^2 = \frac{1}{2} |R - P * Q^T|^2$$

For reducing the error, I have to know in which direction I have to adjust the values of P and Q. In other words, I need to know the gradient at the current values and therefore I differentiate the above equation with respect to these two variables separately. So, this the objective function.

For avoiding over fitting regularization, a new parameter β is introduced for this. Now the objective function is like

$$\boldsymbol{L} = \underset{\boldsymbol{P},\boldsymbol{Q}}{\boldsymbol{min}} \frac{1}{2}\sum_{(u,i)\epsilon C}(\boldsymbol{R}_{u,i} - \boldsymbol{p}_u \boldsymbol{q}_i^T) + \frac{\beta}{2}\left(\left|\left|\boldsymbol{P}\right|\right|_F^2 + \left|\left|\boldsymbol{Q}\right|\right|_F^2\right) \quad (1)$$

The new parameter β is used to control the magnitudes of the user-feature and item-feature vectors such that P and Q would give a good estimation of R without having to contain large number. To learn P and Q, stochastic gradient decent (SGD) is often applied to optimize the objective function in eqn1. The derivatives of the L with respect to P and Q is as follows:

$$\frac{\partial L}{\partial q_i} = \sum_u \boldsymbol{I}_{u,o_i}\left(\boldsymbol{p}_u \boldsymbol{q}_i^T - \boldsymbol{R}_{u,o_i}\right)\boldsymbol{p}_u + \boldsymbol{\beta}\boldsymbol{q}_i \quad (2)$$

$$\frac{\partial L}{\partial p_u} = \sum_{o_i} \boldsymbol{I}_{u,o_i}\left(\boldsymbol{p}_u \boldsymbol{q}_i^T - \boldsymbol{R}_{u,o_i}\right)\boldsymbol{q}_i + \boldsymbol{\beta}\boldsymbol{p}_u \quad (3)$$

P and Q will be updated using the following equations.

$$\boldsymbol{p}_u \leftarrow \boldsymbol{p}_u + ((\boldsymbol{R}_{u,} - \ ) - \boldsymbol{\beta}\boldsymbol{p}_u) \quad (4)$$

$$\boldsymbol{q}_i \leftarrow \boldsymbol{q}_i + \boldsymbol{\alpha}((\boldsymbol{R}_{u,o_i} - \boldsymbol{p}_u \boldsymbol{q}_i^T)\boldsymbol{p}_u - \boldsymbol{\beta}\boldsymbol{q}_i) \quad (5)$$

Algorithm:

1.Initialize:

Initialize P and Q randomly

2. Repeat until converge for i in all user for j in all movie if R[i][j] > 0 : eij = X[i][j] - P[I,:]* Q[:,j]

#compute the gradient from the error for k = 1 to K:

$$P[i][k] = P[i][k] + \alpha(eij * Q[k][j] - (\beta * P[i][k]))$$

$$Q[k][j] = Q[k][j] + \alpha(eij * P[i][k] - (\beta *Q[k][j]))$$

3. Calculate Error error = 0 for i in all user for j in all movie if R[i][j] > 0:

$$error += (R[i][j] - (P[i, :], Q[: , j]))^2$$

4. return P, Q.T

In this MF model, semantic relationship among items are ignored. It only considers the user-rating. The existing MF algorithms have still not fully taken the essential relations between items and users, especially the semantic relations. Therefore, I mix the semantic relations between items into MF to improve the performance of the system.

3.4 Proposed Method

Matrix Factorization has been used widely used technique to recommend an item to a user. It does so by decomposing a matrix into two matrices. The multiplication of both matrices will approximate to previous one. But it has some limitations. In this chapter, I shall discuss how traditional MF works, where it has limitations. Then I will discuss my proposed model, how it works and how it improves performance.

Methodology:

The notation used in my experiments are as following:

X = term frequency matrix

R = Rating matrix of dimension user*movie

Each element r_{ij} means the rating on item j given by user i.

Alpha, α = learning rate which is multiplied with gradient decent.

Beta, β = bias controlling parameter

Lambda = Control semantic relationship among item

Now I shall explain my proposed modified method of matrix factorization. The semantic relation between items is computed first according to textual matrix factorization. Then items' semantic relation and users' rating are combined together. This model is advantageous in following aspects:

1. The semantic relations between items are analyzed by the first application of matrix factorization, which are able to remedy the problem of lacking informative rating knowledge, further to improve quality of recommendation.

2. Users' subjective rating is also combined in the learning modem by second application of matrix factorization.

Obtaining similarity of items:

Similar to basic non-negative MF, a textual MF is applied on term frequency matrix X. X is obtained by term frequency. In my dataset, term indicates the genres or categories of movies. Rows of X represents words i.e genres and columns of X represents the sentences i.e number of movies. Each value x_{ij} of X represents the value of words or genres for item i. Matrix X is decomposed into A and B. Dimension of A is term*k and that of B is movie*k where k = number of latent feature. Here, I have used nonnegative MF. A non-negative MF is same normal MF except that, it factorizes a matrix into two matrices which do not contain any negative value. It makes the resulting matrices easier to inspect. Also, in applications like processing of audio spectrograms of muscular activity, non-negativity is inherent to the data being considered.

Now semantic similarity between item oi and oj is calculated by using cosine similarity of sentence i.e movie i and movie j.

Semantic similarity is a metric defined over a set of documents or terms, where the idea of distance between them based on likeliness of their meaning or semantic content as opposed to similarity which can be estimated regarding their syntactical representation. These are mathematical tools for estimating the strength of the semantic relationship between units of language, concepts or instances, through a numerical description obtained according to the comparison of information supporting their meaning or describing their nature. I have obtained similarity from decomposed matrix B which contains the item features. It is calculated according to following equation:

$$\boldsymbol{S(o_i, o_j) = S(s_i, s_j) = cos < B_i, B_j > = \frac{B_i . B_j}{||B_i|| \, ||B_j||}} \tag{6}$$

Now how relationship between items is incorporated with a traditional matrix factorization will be discussed. Again, I apply MF on rating matrix but this time the update of decomposed matrix will be different. Semantic relationship will be now added during updating P and Q. The learned rating values should be as close as possible to the observed rating values, and the predicted item profiles should be similar to their neighborhoods as Ill, which are derived from their semantic relations. textual MF models relations between words and sentences by accommodating the impact of missing words, which improves the performance of semantic analysis.

Objective function:

The objective function is a equation to be optimized given certain condition or no condition and with variables that need to be minimized pf maximized using non-linear programming techniques. An objective function can be the result of an attempt to express a business goal in mathematical terms for use in decision analysis, operations research or optimization studies.

The objective function of my proposed method is:

$$\boldsymbol{L_u = \frac{1}{2} \sum_{(u,o_i) \epsilon K} \left(R_{u,o_i} - \widehat{R}_{u,o_i}\right)^2 + \frac{\beta}{2}\left(||P||_F^2 + ||Q||_F^2\right) + \frac{\lambda}{2} \sum_{o_i} (q_i - \sum_{o_j} S(o_i, o_j)\, q_i)(q_i - \sum_{o_j} S(o_i, o_j)\, q_i)^T} \tag{7}$$

As I can see, semantic relationship and user rating preference is combined here. So, rating is predicted by both of them.

Update Of P,Q

I have to find optimum values for P and Q such that the value of objective function is minimum. I can do it by partial derivation. It is assumed that partial derivation is equal to zero,then the derivation is solved for P and Q.

The partial derivatives of my objective function with respect to P and Q are

$$\frac{\partial L_u}{\partial p_u} = \sum_{o_i} I_{u,o_i} \left(p_u q_i^T - R_{u,o_i}\right) q_i + \beta p_u \qquad (8)$$

$$\frac{\partial L_u}{\partial q_i} = \sum_u I_{u,o_i} \left(p_u q_i^T - R_{u,o_i}\right) p_u + \beta q_i + \lambda \left(q_i \sum_{o_j} S(o_i, o_j)\, q_j \right) - \lambda \sum_{o_j} S(o_i, o_j) \left(q_j - \sum_{o_k} S(o_j, o_k)\, q_k\right) \qquad (9)$$

The derivatives are solved and I find the equations of P and Q. P and Q will be update as follows:

$$p_u \leftarrow p_u + \alpha(\left(R_{u,i} - p_u q_i^T\right) q_i - \beta p_u) \qquad (10)$$

$$q_i \leftarrow q_i + \alpha((R_{u,i} - p_u q^T{}_i) p_u - \beta q_i - \lambda(q_i - \sum_{oj} S(o_i, o_j)\, q_j) + \lambda \sum_{oj} (o_i, o_j)\, (q_j - \sum_{ok} (o_j, o_k)\, q_k)) \qquad (11)$$

I also have used bias in my method. For example, I can assume that when a rating is generated, some biases may also contribute to the ratings. In particular, every user may have his or her own bias, meaning that he or she may tend to rate items higher or lower than the others. In movie ratings, if a user is a serious movie watcher, he or she may tend to give lower ratings, when compared to another user who generally enjoys movies as long as they are not too boring. A similar idea can also be applied to the items being rated.

Hence, in the equal of predicting a rating, I can also add these biases in order to better model how a rating is generated

$$R_{ui} = b + [u] + [i] + p_u q_i \qquad (12)$$

Here b = global bias b_u = user bias and b_i is item bias.

Using similar equation mentioned in eqn4 and 5, b_u and b_i is updated.

The update rule for bias is

$\boldsymbol{b_u}\,[\boldsymbol{u}] \leftarrow \boldsymbol{b_u}\,[\boldsymbol{u}] + \boldsymbol{\alpha}(\boldsymbol{e_{u,\mathrm{i}}} - \boldsymbol{\beta b_u}\,[\boldsymbol{u}])$

$\boldsymbol{b_i}\,[\boldsymbol{i}] \leftarrow \boldsymbol{b_i}\,[\boldsymbol{i}] + \boldsymbol{\alpha}(\boldsymbol{e_{u,\mathrm{i}}} - \boldsymbol{\beta b_i}[\boldsymbol{i}])$

So, my procedure starts with factorizing the term frequency matrix X. When P and Q is learned, rating can be predicted by eqn11.

Algorithm of Modified MF:

1. Initialize: Initialization: Initialize P and Q with random number from normal distribution

2. While $((L^{i+1} - L^{i}) < \in)$ For all genre of a movie

   For all movies

   Compute and update A and B using equation mentioned previously

3. Computer similarity from B
4. Repeat until converge for u in all user for i in all movie if R[u][i] > 0 :

$e_{ui} = R_{ui} - p_u\, q_i$

#compute the gradient from the error

$$p_u = p_u + \alpha(e_{ui}q_i - (\beta p_u))$$

$$q_i = q_i + \alpha(e_{ui}p_u - \beta q_i - \lambda\,(q_i - \textstyle\sum_{oj} S(o_i, o_j)\, q_j) + \lambda \textstyle\sum_{oj} S(o_i, o_j)\,(q_j - \textstyle\sum_{ok} S(o_j, o_k)\, q_k))$$

5. Calculate Error error = 0 for i in all user for j in all movie if R[i][j] > 0:

error += (R[i][j] − (P[i, :] ∗ Q[: , j]) ^{2} 6. return P, Q.T

Complexity:

More time is consumed when P and Q is being learned. Because for each iteration before convergence, for each item and user the gradient decent is computed. The complexity during learning latent feature of P and Q is then O (N*K*n + N*K*n*n) where n is number of neighbor for each item, N is the number of given rating in training set of data and K is the number of latent feature.

3.5 Implementation Requirements

After the proper analysis on all necessary statistical or theoretical concepts and methods, a list of requirement has been generated that must be required for such a work for building a recommendation system. The probable necessary things are:

Hardware/Software Requirements

- Operating System (Windows 7 or above)
- Hard Disk (minimum 256GB)
- Ram(more than 2 GB)

Developing Tools

- Python Environment
- Anaconda
- Pycharm
- Notepad ++
- CMD

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction

This chapter 4 mainly focuses on the descriptive analysis of the data used in the research as well as the experimental results of my project.

4.2 Dataset & Data format

I have used movielens 100k dataset. It contains 100000 ratings of approximately 8927movies given by 718 users. The ratings are in range of 0 to 5. Each user has given at least 20 ratings. The movie titles, categories are also provided here in a different file. Ratings.csv file is the rating file. Every line of this file represents one rating of one movie by one user and has the resulting format: userId , movieId, rating, timestamp.

Movie information is contained in the file movies.csv. Each line of this file after the header row represents one movie, and has the following format: movieId , title , genres

Data Format:

Table 1: Format of data set

User id	Movie id	Ratings	Timestamp

Data Statistics:

Table 2: Data statistics

No. of users	No. of movies	No. of Ratings
718	8927	100000

4.3 Experimental Result:

Data has divided into training and test sets. Training set has 75% data and test set has 25% data. I also have construct term document matrix (TDM) using term frequency of movie from movie description. This TDM is used to obtain similarity matrix of movies which is needed in update of P and Q.

I have used root squared mean error in my experiment.

Root mean square error, RMSE = $\sqrt{\frac{1}{N}\sum_{(u,i)\in R_{test}}(r_{ui} - r_{ui}^{'})^2}$

Where R_{test} is the test set. The error is calculated for non-zero value of R_{test}.
For comparing my method, I have taken user-based CF, item-based CF, SVD and conventional MF.

In my experiment I have set Latent feature K 8 and it is fixed. Again, Due to time complexity, I could not check for different values of K.

Comparison of Results:

I have obtained errors of user and item-based CF and SVD by implementing these separately. It is seen that, for the given data, rmse of user-based CF is 3.1225 and my proposed method gives rmse of 1.03 that means 67% improvement. Item based gives rmse of 3.4487. My TDM based MF gives 70% better result.

SVD has rmse of 2.7131 using MovieLens 100k data. So my result is 62% better than SVD. For, SVD I have set latent feature 8. And For Modified MF, I have used 8 latent features.

The improvement is the contribution of similarity of item categories.

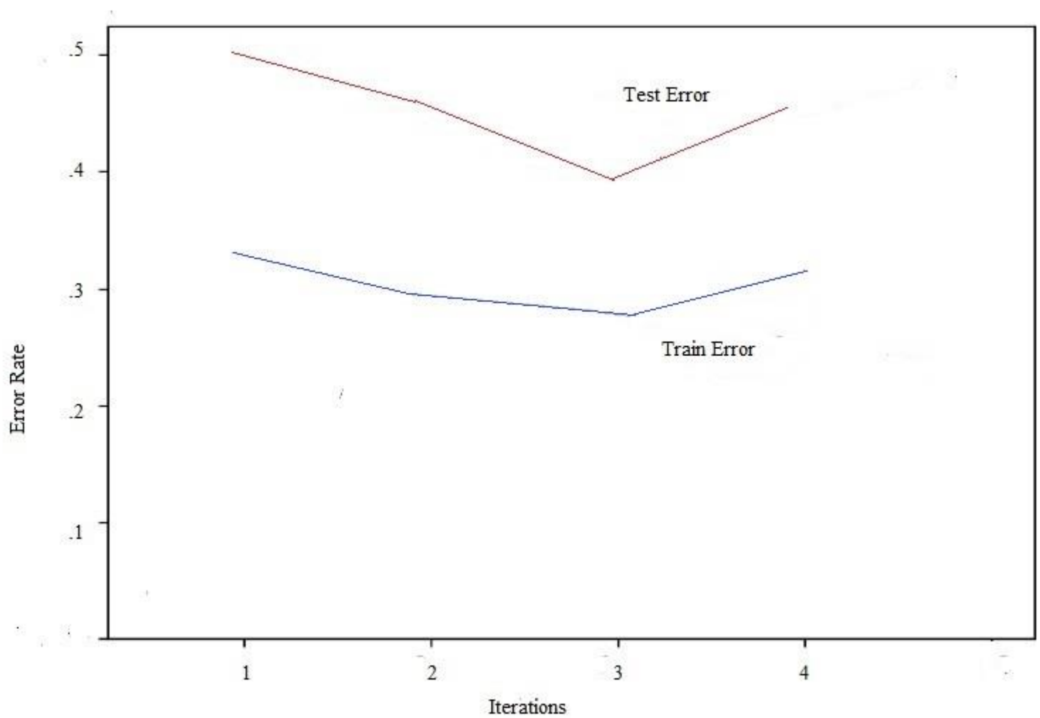
In table 3, the summary is given:

Table 3: Format of data set

Metric	User-Based CF	Item-Based CF	SVD	TDM-Similarity based MF
RMSE	3.1225	3.4487	2.7131	1.03
Improvement	67%	70%	62%	

4.4 Statistical graphs

The following graph shows the error with increase of iteration. But since we have taken few number of iteration, our result may not be in global optima. This graph shows, that after first 3 iterations, error decreases and then error increases (Test&Train Data). So, from this curve, the error is minimum for 3 iterations.



Impact of alpha:

Alpha is learning rate. The fraction of gradient decent to be taken is defined by alpha. Larger value of alpha indicates a larger contribution of gradient decent. To find optimum value of alpha, I have tested for different value of alpha in range 0.0002. The optimum value for which rmse is smallest is 0.0010. From fig 4.1 I can see the impact of alpha. Iteration number is set to 10.

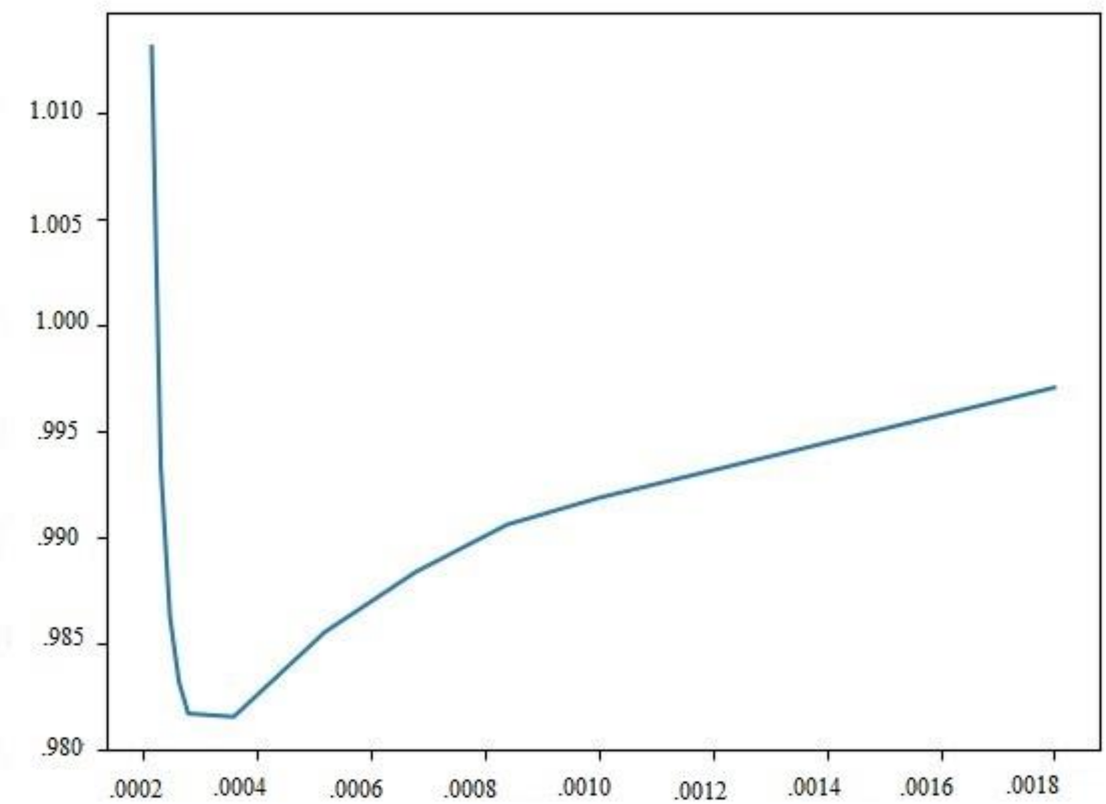


Figure 4.1: Impact of Alpha

Impact of beta:

Beta is a parameter that is used for regularization. It controls the magnitude of user and item feature vectors. It is usually set to an order of 0.02. To get optimum value of beta, I have test for values in range 0.02 to 0.18. I have set iteration 10 while testing for beta. From the graph, I see that, beta = 0.12 is optimum value.

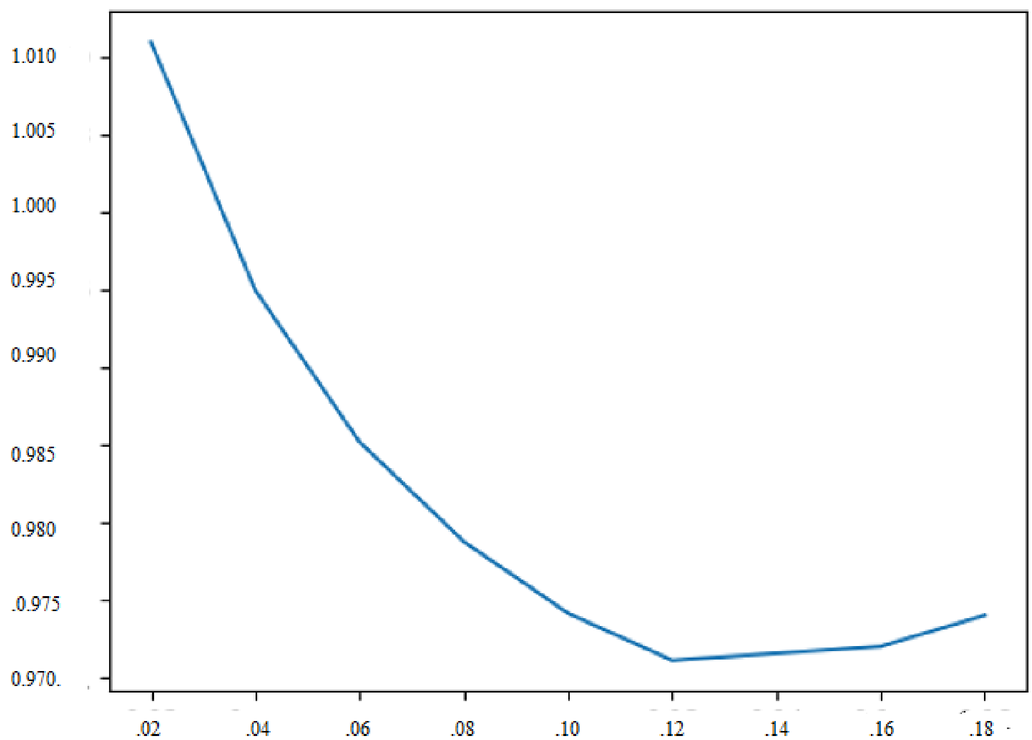


Figure 4.2: Impact of beta

Impact of lambda:

Value of lambda effects the relationship among movies. The higher the lambda, the higher the impact of similarity betIen movies. I have checked the rmse for different values of lambda. For a certain value of lambda, the iteration will continue until the rmse converge. I have taken lambda in range of 0 to 1. The test for different values of lambda is done to get optimum value of alpha.

And from fig-4, I see , optimum value for lambda is 0.2 for which rmse is smallest. I have set iteration = 30 while testing for lambda.

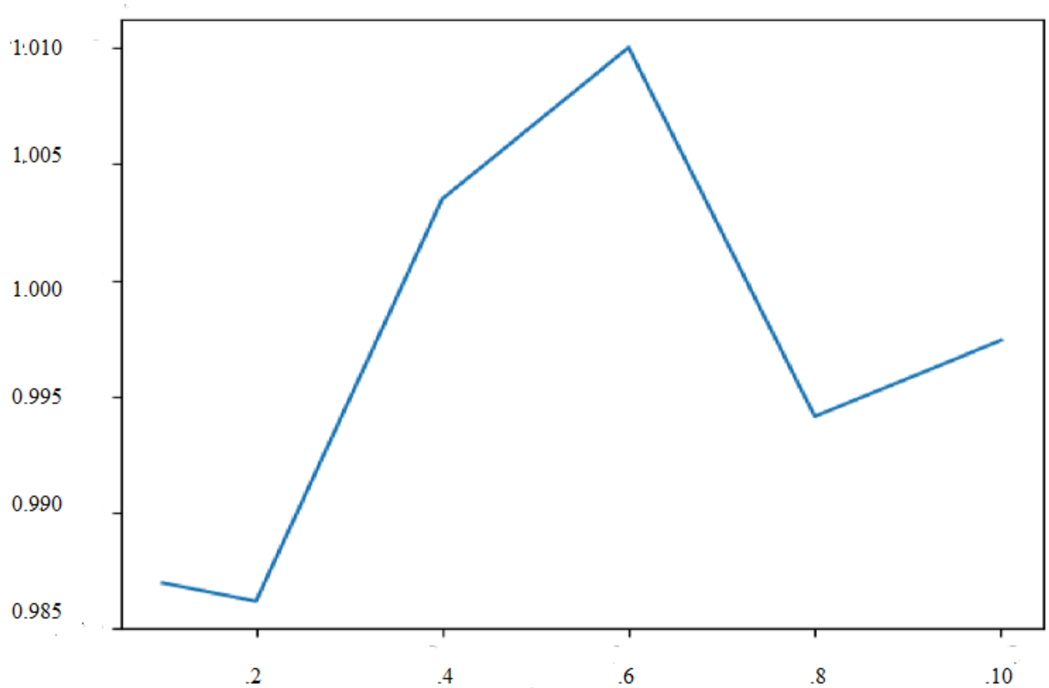


Figure 4.2: Impact of beta

4.4 Summary

I have discussed performance of proposed method. The result shows that my method works much better than some Ill-known conventional method for the MovieLens dataset which is a sparse dataset.

CHAPTER 5

SUMMAY, CONCLUSION, RECOMMENDATION AD IMPLECATION FOR FUTURE RESEARCH

5.1 Summary of the Study

In my experiment, I have proposed a modified version of MF where MF is used twice. Once MF is used to get feature vector from term document matrix. Then again MF is used to obtain two feature matrixes from rating matrix. Thus, two-time MF is applied. A decomposed matrix representing item feature is used to obtain similarity matrix of movies. Then this similarity is used in traditional MF with regularization and biases. Regularization helps to prevent over fitting and biases give better results. I have used a real data set in this experiment.

5.2 Implication for Further Study

Because of time complexity, I could not check the code for more parameters and I also could not use more iteration.

So, more iteration can be applied and also a larger data set can be used in this experiment if there is enough technical support.

5.3 Conclusion

Although I Could not use a larger dataset because of lacking of technical support but I hope it will be very beneficial to the future researchers to do such kind of research on recommender systems.

References:

[1] Fangfang Li, Guandong Xu, Longbing Cao BC (2015) Two-level matrix factorization for recommender systems.

[2] Alter O, Brown PO, Botstein D (2000) Singular value decomposition for genomewide expression data processing and modelling. Proc Natl Acad Sci (PNAS) 97(18):10101–10106

[3] Agarwal D, Chen BC (2010) flda: matrix factorization through latent Dirichlet allocation. In: WSDM, pp 91100.

[4] Blei DM, Ng AY, Jordan MI, Lafferty J (2003) Latent Dirichlet allocation. J Mach Learn Res 3:2003.

[5] Deshpande M,Karypis G (2004) Item-based top-n recommendation algorithms. ACM Trans Inf Syst22(1):143177.doi: 10. 1145/963770.963776

[6] Gantner Z, Drumond L, Freudenthaler C, Rendle S, SchmidtThieme L (2010) Learning attribute-to-feature mappings for coldstart recommendations. In: ICDM, pp 176–185

[7] Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Li Y, Liu B, Sarawagi S(eds) KDD. ACM, pp 426–434

[8] Landauer TK, Foltz PW, Laham D (1998) An introduction to latent semantic analysis. Discourse Process 25:259–284

[9] Li F, Xu G, Cao L (2014) Coupled item-based matrix factorization. In: WISE (1), Lecture notes in computer science. Springer, pp 1–14

[10] Ma H (2013) An experimental study on implicit social recommendation. In: SIGIR, pp 73–82

[11] Ma H, Yang H, Lyu MR, King I (2008) Sorec: social recommendation using probabilistic matrix factorization. In: Shanahan JG, Amer-Yahia S, Manolescu I,
Zhang Y, Evans DA, Kolcz A, Choi KS, Chowdhury A (eds) CIKM. ACM, pp 931– 940

[12] Melville P, Sindhwani V (2010) Recommender systems. Encycl Mach Learn 2010:829–838

[13] Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Adv Artif Intell 2009:421425:1–421425:19

[14] Wang C, Blei DM (2011) Collaborative topic modeling for recommending scientific articles. In: KDD, pp 448–45

[15] Wang C, Cao L, Wang M, Li J, Wei W, Ou Y (2011) Coupled nominal similarity in unsupervised learning. In: CIKM, pp 973–978

[16] Yang X, Steck H, Liu Y (2012) Circle-based recommendation in online social networks. In: KDD, pp 1267–1275