Analysis of Collaborative Filtering Algorithms

Thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Karumoju Dileep[110cs0123]

and

Challa Mallikarjuna Rao[110cs0419]

under the guidance of

Dr. Korra Sathya Babu

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India
May 2014

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Dr. Korra Sathya Babu
Assistant Professor

May 11^{th}*, 2014*

Certificate

This is to certify that the work in the thesis entitled *Analysis of Collaborative Filtering Algorithms* by *Karumoju Dileep* and *Challa Mallikarjuna Rao*, bearing roll numbers *110cs0123* and *110cs0419*, is a record of an original research work carried out by them under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Dr. Korra Sathya Babu

Acknowledgement

We take this opportunity to express our profound gratitude and deep regards to our guide Dr. Korra Sathya Babu sir for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. Like a true mentor, he motivated and inspired us through the entire duration of work, without which this project could not have seen the light of the day. We convey our regards to all the faculty members of Department of Computer Science and Engineering, NIT Rourkela for their valuable guidance and advices at appropriate times. We would like to thank our friends for their help and assistance all through this project.

Last but not the least, we express our profound gratitude to the Almighty and our parents for their blessings and support without which this task could have never been accomplished.

Authors Declaration

We hereby declare that all the work contained in this report is our own work unless otherwise acknowledged. Also, all of our work has not been previously submitted for any academic degree. All sources of quoted information have been acknowledged by means of appropriate references.

(Challa Mallikarjuna Rao)

(Karumoju Dileep)

(NIT Rourkela)

Abstract

Recommender System is a subclass of information filtering system which predicts the rating given to an item by any user. Collaborative filtering is a key technique in recommender systems. This technique predicts the user rating of an item by collaboration of other users who have similar interests with this user.

Collaborative filtering approaches can be categorized as Memory-based, Model-based and Hybrid approaches. In memory-based approach similarity is computed between users or items which is used to make recommendations. Memory-based approach can be further classified as Item-based and User-based recommendations. Pearson correlation scheme belongs to user-based scheme and Slope one family of algorithms belong to item-based scheme. Slope one family consists of Normal slope one, Weighted slope one and Bipolar slope one. In model-based approach models are developed for making predictions of real-time data. Algorithms belonging to this approach are Singular value decomposition, Regularized Singular value decomposition, Probabilistic Matrix Factorization and Latent dirichlet allocation. In hybrid approach combination of memory-based and model-based approaches are used for making recommendations.

In this thesis an attempt has been made to analyze various algorithms in Memory-based and Model-based approaches. In model based algorithms, we analyzed Singular Value Decomposition (SVD) and Regularized Singular Value Decomposition (RSVD). By taking three different dataset sizes, we observed that RSVD outperforms SVD for all three dataset sizes. In memory based algorithms, we analyzed Pearson correlation scheme which takes the correlation between user vectors as similarity measure and Slope one family of algorithms. In slope one algorithms, we proposed an improvement to the existing scheme for determining Threshold value of Bipolar slope one algorithm. We used Median of user ratings and Average of min-max ratings which outperforms the existing user average scheme. Finally, we made an analysis of all these algorithms and concluded that RSVD outperforms rest of the algorithms in terms of accuracy of predictions.

Keywords: Collaborative filtering, SVD, RSVD, Slope one, Recommender systems

Contents

Certificate i

Acknowledgement ii

Authors Declaration iii

Abstract iv

List of Figures vii

1 Introduction 1

1.1 Introduction to Recommender Systems 1

1.2 Collaborative Filtering 2

1.2.1 Introduction to Collaborative Filtering 2

1.2.2 Work flow of CF algorithms 2

1.2.3 Types 2

1.2.4 Challenges of CF 3

1.3 Problem Statement 4

1.4 Our Contribution 4

1.5 Organization of the Thesis 4

2 Literature Survey 6

2.1 Memory Based Approaches 6

2.1.1 Pearson Correlation Scheme 6

2.1.2 Slope one family of algorithms 7

2.2 Model-based approaches 8

2.2.1 Singular Value Decomposition 9

2.2.2 Regularized Singular Value Decomposition 10

3 Proposed Work **11**

3.1 Memory-based approaches 11

3.1.1 Pearson Correlation Scheme 11

3.1.2 Slope One Algorithms 12

3.2 Model-based approaches 14

3.2.1 Singular Value Decomposition (SVD) 14

3.2.2 Regularized Singular Value Decomposition (RSVD) 15

4 Analysis and Results **16**

5 Conclusion and Future Work **20**

Bibliography **21**

List of Figures

4.1 RMSE of SVD for dataset of size 300 16
4.2 RMSE of SVD for dataset of size 3000 17
4.3 RMSE of SVD for dataset of size 30000 17
4.4 RMSE of RSVD for dataset of size 300 17
4.5 RMSE of RSVD for dataset of size 3000 17
4.6 RMSE of RSVD for dataset of size 30000 17
4.7 RMSE of Pearson for all datasets of size 300,3000 and 30000 17
4.8 RMSE of Normal Slope one for all datasets of size 300,3000 and 30000 18
4.9 RMSE of Weighted Slope one for all datasets of size 300,3000 and 30000 18
4.10 RMSE of Bipolar1 for all datasets of size 300,3000 and 30000 18
4.11 RMSE of Bipolar2 for all datasets of size 300,3000 and 30000 18
4.12 RMSE of Bipolar3 for all datasets of size 300,3000 and 30000 18
4.13 RMSE of All algorithms for all datasets of size 300,3000 and 30000 19

Chapter 1

Introduction

The growth of the internet and the cloud sources have made it difficult to extract the required useful information from all the available information. This large size of data require techniques for efficient extraction of necessary information. This is called information filtering. An information filtering system is a system that removes redundant and unwanted information from an information stream using some automated or computerized methods before presenting it to the users.

Recommender systems or recommendation systems are a sub class of information filtering systems that are used to predict the rating or the preference given by the user to an item. There are different kinds of approaches for implementing the recommender systems among them collaborative filtering is one such approach.

In this report we make a comparative analysis of different collaborative filtering algorithms and their nature on different datasets.

1.1 Introduction to Recommender Systems

Recommender systems are the sub-class of the information filtering systems that are used to predict the user's rating or preference for a particular item. Recommender systems are applied in a wide variety of applications such as movies, news, music, books, search queries, research articles and products.

Recommender systems uses two ways for producing a set of recommendations. One is Collaborative Filtering and other is Content-based Filtering.

1.2 Collaborative Filtering

1.2.1 Introduction to Collaborative Filtering

Collaborative filtering is a technique used by some of the recommender systems in which predictions are made automatically about the interest of the user by the information collected from users with similar interests.

Example: A group of users' rate different items (like videos, images, games). Generally, users do not give ratings for all the items. System predicts the ratings of items which are not yet rated.The motivation for CF is that people often get the best recommendation from the persons of similar interests. CF explores techniques of matching interests of different people and developing patterns in order to get recommendations.

1.2.2 Work flow of CF algorithms

Typically the work flow of CF system is

1. A user gives some rating to an item. This rating represents user's interest in that item .
2. The system searches for other users who gave similar ratings to this item .
3. Now, using the similarity information system predicts the ratings of items for which this user has not given any rating yet.

1.2.3 Types

Memory Based:

This computes the similarity between users or items based on user rating data which is used for making recommendations.

Advantages of this approach are implementation is easy and effective, results are explainable, new data can be added easily and scales well with co-rated items.

Disadvantages are its performance decreases when sparsity in data increases and even though it handles new users efficiently adding new items becomes complicated because it relies on a data structure.

Examples: User-based, Item-based recommendations.

Model based:

In this approach several algorithms are used to find patterns based on training data to develop models which are used to make predictions of real-time data.

Advantages of this approach are it handles sparsity better than memory-based algorithms, improves prediction performance and also gives an intuitive rationale for the recommendations.

There are some disadvantages with this approach which are expensive model building and loss of information.

Examples: Singular value Decomposition and Probabilistic latent semantic analysis.

Hybrid:

This approach is the combination of memory-based and model-based ones. Advantages with this approach are this overcomes the limitations of native CF approaches, problems such as sparsity and loss of information.

Disadvantage with this approach is that it is highly complex and expensive to implement.

1.2.4 Challenges of CF

Data Sparsity

In practice many data sets are very large as a result the user-item matrix is extremely large and a sparse one.

Sparsity in the data causes the cold start problem. In collaborative filtering methods recommendations are made based on users' previous preferences. Newly added users will have to give ratings for a number of items so that the system can give reliable recommendations. Similarly, there exists same problem with new items. New items have to be rated by substantial number of users to recommend them to users who have similar tastes with the ones rated them. In content-based recommendation, the recommendation of an item is based on its discrete set of descriptive qualities rather than its ratings. So new item problem doesn't occur in those systems.

Scalability

As the number of users and items grow CF algorithm suffer scalability issues i.e. the complexity of the algorithm is very high and there is a need for immediate response to the users hence it requires to maintain a large clusters of machines for making preferences.

Grey sheep

Grey sheep are the users whose opinion constantly differ from the other users' preferences which makes them not get benefitted by collaborative filtering. Black sheep are the ones those make opposite preferences which makes it almost impossible for the prediction.

Shilling attack

In a recommendation system it is possible that people give high ratings for their own items and very low ratings for their competitors. Hence there is a necessity for taking precautions to avoid such kind of manipulations.

1.3 Problem Statement

If a database collects preferences of n users and m objects as scores which is a rating between a range of values R_{min} to R_{max}. Usually a user does not score all the objects. Let A R^{n*m} be the matrix collected by the scores in the database. In most cases A is sparse because many users score only a few objects. The existing scores in A work as training data and the goal is to predict the missing values in the database.

1.4 Our Contribution

- In this thesis we implement a few Collaborative filtering algorithms on a particular dataset, analyze each of them and compare their performance with each other.
- We propose an improvement to the existing threshold selecting technique of Bipolar slope one algorithm.

1.5 Organization of the Thesis

- In Chapter 2, we have given the Literature Survey which includes the review of some memory-based and model based approaches and what are the existing algorithms in these approaches.
- In Chapter 3, we have discussed the work proposed and algorithms used in the implementation.

- In Chapter 4, results of the various implemented algorithms and the proposed algorithms are discussed as well as the results are analyzed.
- In Chapter 5, we have the results and the conclusions drawn from the results as well as the scope for future work is discussed.

Chapter 2

Literature Survey

2.1 Memory Based Approaches

Notation:

Following notation is used in implementing the following schemes [2].

The ratings from a given user called an evaluation is represented as an incomplete array u, where u_i is the rating of this user given to item i. The subset of set of items consisting of all those items which are rated in u is S(u). The set of all evaluations in the training set χ. The number of elements in a set S is called card(S). The average of ratings in an evaluation u is denoted by $\overline{u}$. The set $S_i(\chi)$ is the set of all evaluations u$\epsilon\chi$ such that they contain item i (iϵS(u)). Given two evaluations u and v, we define the scalar product $< u, v >$ as $\sum_{i\epsilon S(u)\cap S(v)} U_i * V_i$. $P(u)_i$ is the prediction of item i in evaluation u.

2.1.1 Pearson Correlation Scheme

In this method correlation between user rating vectors is calculated to predict the rating of a user for an item.

Concepts:

In Pearson scheme, prediction function takes the form of weighted sum over all evaluations in χ.

$$P(u)_i = \overline{u} + \frac{\sum_{v \epsilon S_i(\chi)} \gamma(u,v)(v_i - \overline{v})}{\sum_{v \epsilon S_i(\chi)} |\gamma(u,v)|}$$

Where γ is a similarity measure computed from Pearson correlation [2].
Pearson correlation is computed as follows [2],

$$Corr(u,w) = \frac{<u-\overline{u}, w-\overline{w}>}{\sqrt{\sum_{i \epsilon S(u) \cap S(w)} (u_i - \overline{u})^2 \sum_{i \epsilon S(u) \cap S(v)} (w_i - \overline{w})^2}}$$

$$\gamma(u,w) = Corr(u,w)|Corr(u,w)|^{\rho - 1}$$

Where ρ is the Case Amplification Power which reduces the noise in the data.

2.1.2 Slope one family of algorithms

Slope one is a rating based collaborative filtering algorithm family predicting how a user would rate a given item from other users' ratings. Slope one works on the intuitive principle of a popularity differential between items for users. This includes Normal slope one, Weighted slope one and Bipolar slope one [5].

Normal Slope one

Given a training set χ and any two items j an i with ratings u_j and u_i respectively in some evaluation u, then average deviation of item i with respect to item j is defined as,

$$dev_{j,i} = \sum_{u \epsilon S_{j,i}(\chi)} \frac{u_j - u_i}{card(S_{j,i}(\chi))}$$

Note that any user evaluation u not containing both u_j and u_i is not included in the summation. The symmetric matrix defined by $dev_{j,i}$ can be computed once and updated quickly when new data is entered. So given $dev_{j,i}$ and rating of other items in the evaluation u_i , prediction of u_j is computed as follows,

$$P(u)_j = \frac{1}{card(R_j)} \sum_{i \epsilon R_j} (dev_{j,i} + u_i)$$

Where $R_j = \{i | i \epsilon S(u), i \neq j, card(S_{j,i}(\chi)) > 0\}$ is the set of all relevant items [2].
Normal slope one doesn't depend on how the user rated individual items, but only on the user's average rating and crucially on which items the user has rated.

Weighted Slope one

In weighted slope one, the number of ratings observed are taken into consideration while calculating the prediction of a user for an item. Suppose to predict the user A's rating of item L given user A's rating of items J and K, if 2000 users rated the pair of items J and L where only 20 users rated pair of items K and L, then user A's rating of item J is likely to be far better predictor for item L than user A's rating of item K is.
Thus, the prediction function for weighted slope one is defined as follows,

$$P^{wS1}(u)_j = \frac{\sum_{i \epsilon S(u)-\{j\}} (dev_{j,i}+u_i)c_{j,i}}{\sum_{i \epsilon S(u)-\{j\}} c_{j,i}}$$

Where $c_{j,i} = \text{card}(S_{j,i}(\chi))$.

Bipolar Slope one

In Bipolar slope one prediction is divided into two parts. One prediction is derived from items users liked and another prediction using items that users disliked. Threshold for deciding whether an item is liked by a user or not is calculated by taking user's average.
The deviation function for liked or disliked items is defined as follows,

$$dev^{like}_{j,i} = \sum_{u \epsilon S^{like}_{j,i}(\chi)} \frac{u_j - u_i}{card(S^{like}_{j,i}(\chi))}$$

Now, the prediction of an item i in user evaluation u for Bipolar slope one is computed as follows,

$$P^{bpS1}(u)_j = \frac{\sum_{i \epsilon S^{like}(u)-\{j\}} p^{like}_{j,i} c^{like}_{j,i} + \sum_{i \epsilon S^{dislike}(u)-\{j\}} p^{dislike}_{j,i} c^{dislike}_{j,i}}{\sum_{i \epsilon S^{like}(u)-\{j\}} c^{like}_{j,i} + \sum_{i \epsilon S^{dislike}(u)-\{j\}} c^{dislike}_{j,i}}$$

Where the weights $c^{\text{like}}_{\text{j,i}} = \text{card}(S^{\text{like}}_{\text{j,i}})$ and $c^{\text{dislike}}_{\text{j,i}} = \text{card}(S^{\text{dislike}}_{\text{j,i}})$.

2.2 Model-based approaches

In Model based approaches, a model is developed by applying machine learning algorithms on training data. This model is used to predict ratings in real world. Model based approaches are slow processes as it takes too much time to train a model. Some of the popular model based approaches are Singular Value Decomposition (SVD) and Regularized Singular Value Decomposition (RSVD) [7].

2.2.1 Singular Value Decomposition

Introduction

Singular Value Decomposition is a process of decomposing a matrix into its factor matrices. Suppose M is a real or complex matrix of size m * n. Then singular value decomposition of M is defined as

$$M = U\Sigma V^*$$

Where U is a m*m unitary matrix, Σ is a m*n rectangular diagonal matrix and V* is conjugate transpose of n*n unitary matrix V. m columns of U are called left-singular vectors, n columns of V are called right-singular vectors, diagonal entries of Σ are called singular values of M [1].

Concepts

Given an input rating matrix M of size m*n which consists of ratings of m users and n movies. Low-rank matrix approximation of M using singular value decomposition gives two feature matrices corresponding to users and movies. User feature matrix P is of size m*k represents the associativity of a user with k features. Movie feature matrix Q is of size k*n represents the associativity of a movie with k features. To obtain P and Q, matrix M is decomposed into three matrices U, S, V. U is a m*m matrix, S is a m*n diagonal matrix and V is a n*n matrix. Now only the k left most columns are taken from U, k top most rows are taken from V and only k singular values are taken from S. Now P and Q are calculated as following,

$$P = U * \sqrt{S}, \text{ where dimension of U is m*k and S is k*k}$$

$$Q = \sqrt{S} * V, \text{ where dimension of S is k*k and V is k*n}$$

After obtaining P and Q, rating of user i for movie j is calculated as following,

$$\text{Pred(i,j)} = \text{dot product of } P_i \text{ and } Q_j$$

Where P_i is user feature matrix for user i, Q_j is movie feature matrix for movie j.
P and Q are updated using gradient descent approach in which negative gradients are calculated by differentiating objective function with respect to both user and movie feature matrices [1]. Objective function and negative gradients are defined as follows,

$$E = (1/2) * \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} * (M_{ij} - Pred(i,j))^2$$

$$-\Delta P_i = \sum_{j=1}^{n} I_{ij} * (M_{ij} - Pred(i,j)) * Q_j, \text{i=1...m}$$

$$-\Delta Q_j = \sum_{i=1}^{m} I_{ij} * (M_{ij} - Pred(i,j)) * P_i, \text{j=1...n}$$

Now,

$$P_i = P_i + \alpha * \Delta P_i$$

$$Q_j = Q_j + \alpha * \Delta Q_j$$

Where α is learning rate which is usually taken as 0.02, 0.01 etc.

2.2.2 Regularized Singular Value Decomposition

Regularized SVD is a technique used for collaborative filtering proposed by Simon Funk which includes regularization constants along with learning rate [3]. In this technique objective function and negative gradients are defined as follows,

$$E = (1/2) * \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} * (M_{ij} - Pred(i,j))^2 + (\beta/2) * (\sum_{i=1}^{m} |P_i|^2 + \sum_{j=1}^{n} |Q_j|^2)$$

$$-\Delta P_i = \sum_{j=1}^{n} I_{ij} * (M_{ij} - Pred(i,j)) * Q_j - \beta * P_i, \text{i=1...m}$$

$$-\Delta Q_j = \sum_{i=1}^{m} I_{ij} * (M_{ij} - Pred(i,j)) * P_i - \beta * Q_j, \text{j=1...n}$$

Now,

$$P_i = P_i + \alpha * \Delta P_i$$

$$Q_j = Q_j + \alpha * \Delta Q_j$$

Where α is learning rate which is usually taken as 0.02, 0.01 etc. and β is regularization coefficient which is taken as 0.002, 0.001 etc.

Chapter 3

Proposed Work

3.1 Memory-based approaches

3.1.1 Pearson Correlation Scheme

Algorithm 1 : Pearson Correlation

1: Form the input matrix AϵR^{m*n} from the given dataset.

2: Calculate Indicator matrix which indicates I$\epsilon\{0, 1\}^{m*n}$ which user have rated the item.

3: Calculate the prediction matrix PϵR^{m*n} by taking the weighted sum over all user vectors in the input matrix and correlation between user feature vectors.

$$P(u)_i = \overline{u} + \frac{\sum_{v \epsilon S_i(\chi)} \gamma(u,v)(v_i - \overline{v})}{\sum_{v \epsilon S_i(\chi)} |\gamma(u,v)|}$$

Where γ is a similarity measure computed from Pearson correlation.

4: Pearson correlation is computed as follows,

$$Corr(u,w) = \frac{<u-\overline{u}, w-\overline{w}>}{\sqrt{\sum_{i \epsilon S(u) \cap S(w)} (u_i - \overline{u})^2 \sum_{i \epsilon S(u) \cap S(v)} (w_i - \overline{w})^2}}$$

$$\gamma(u,w) = Corr(u,w)|Corr(u,w)|^{\rho-1}$$

Where ρ is the Case Amplification Power which reduces the noise in the data.

Explanation:

This algorithm takes a m*n rating matrix as input and produces the Root Mean Square Error of Existing and Predicted ratings.

3.1.2 Slope One Algorithms

Normal Slope one

Algorithm 2 :Normal Slope one

1: Form the input matrix AϵR^{m*n} from the given dataset.

2: Calculate Indicator matrix which indicates I$\epsilon\{0, 1\}^{m*n}$ which user have rated the item.

3: Calculate the prediction matrix PϵR^{m*n} by taking the average rating differences between ratings of item j and rest of the items.

$$P(u)_j = \frac{1}{card(R_j)} \sum_{i \epsilon R_j} (dev_{j,i} + u_i)$$

4: Calculate the Root Mean Square Error (RMSE) of Existing and Predicted ratings.

$$RMSE(P, A) = \sqrt{\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (A_{ij} - P_{ij})^2}{\sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}}}$$

Explanation:

This algorithm takes a m*n rating matrix as input and produces the Root Mean Square Error of Existing and Predicted ratings.

Weighted Slope one

Algorithm 3 :Weighted Slope one

1: Form the input matrix AϵR^{m*n} from the given dataset.

2: Calculate Indicator matrix which indicates I$\epsilon\{0, 1\}^{m*n}$ which user have rated the item.

3: Calculate the prediction matrix PϵR^{m*n} by taking the weighted average rating differences between ratings of item j and rest of the items.

$$P^{wS1}(u)_j = \frac{\sum_{i \epsilon S(u) - \{j\}} (dev_{j,i} + u_i) c_{j,i}}{\sum_{i \epsilon S(u) - \{j\}} c_{j,i}}$$

Where $c_{j,i} = \text{card}(S_{j,i}(\chi))$.

4: Calculate the Root Mean Square Error (RMSE) of Existing and Predicted ratings.

$$RMSE(P, A) = \sqrt{\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (A_{ij} - P_{ij})^2}{\sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}}}$$

Explanation:

This algorithm takes a m*n rating matrix as input and produces the Root Mean Square Error of Existing and Predicted ratings.

Bipolar Slope one

Algorithm 4 :Bipolar Slope one

1: Form the input matrix $A \epsilon R^{m*n}$ from the given dataset.

2: Calculate Indicator matrix which indicates $I \epsilon \{0, 1\}^{m*n}$ which user have rated the item.

3: Calculate the prediction matrix $P \epsilon R^{m*n}$ by taking the average rating differences between ratings of disliked pairs or liked pairs of items for each user.

$$P^{bpS1}(u)_j = \frac{\sum_{i \epsilon S^{like}(u)-\{j\}} p^{like}_{j,i} c^{like}_{j,i} + \sum_{i \epsilon S^{dislike}(u)-\{j\}} p^{dislike}_{j,i} c^{dislike}_{j,i}}{\sum_{i \epsilon S^{like}(u)-\{j\}} c^{like}_{j,i} + \sum_{i \epsilon S^{dislike}(u)-\{j\}} c^{dislike}_{j,i}}$$

Where the weights $c^{like}_{j,i} = card(S^{like}_{j,i})$ and $c^{dislike}_{j,i} = card(S^{dislike}_{j,i})$.

4: Calculate the Root Mean Square Error (RMSE) of Existing and Predicted ratings.

$$RMSE(P, A) = \sqrt{\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (A_{ij} - P_{ij})^2}{\sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}}}$$

Explanation:

This algorithm takes a m*n rating matrix as input and produces the Root Mean Square Error of Existing and Predicted ratings.

Improvised Threshold:

- Threshold Value: Determines whether an item is liked or disliked by a user.
- Existing scheme: Threshold value is computed by taking user rating averages.
- Proposed scheme: Threshold value is computed by taking Median of user ratings and by taking the average of min-max ratings.

3.2 Model-based approaches

3.2.1 Singular Value Decomposition (SVD)

Algorithm 5 :Singular Value Decomposition (SVD)

1: Form the input matrix $A \epsilon R^{m*n}$ from the given dataset.

2: Calculate the indicator matrix $I \epsilon \{0, 1\}^{m*n}$ that indicates which movies are rated by users.

3: A is given as input to SVD to get the feature matrices $U \epsilon R^{k*m}$ and $M \epsilon R^{k*n}$, where k is no. of features.

4: Calculate the prediction matrix as follows,

$$p(U_i, M_j) = \begin{cases} a & \text{if } U_i^T M_j < 0 \\ a + U_i^T M_j & \text{if } 0 \leq U_i^T M_j \leq b - a \\ b & \text{if } U_i^T M_j > b - a \end{cases}$$

where p is the prediction function which takes U_i, M_j are the feature vectors as arguments and computes the prediction value which lies in the range of (a, b).

5: Calculate the RMSE from the obtained prediction matrix.

6: To optimize the error, we use gradient descent approach i.e. the partial derivative of the squared error with respect to each parameter U_{ki} and M_{kj}.

$$U_{ki(t+1)} = U_{kit} + \alpha * (2 * (A_{ij} - P_{ij}) * M_{kit})$$

$$M_{ki(t+1)} = M_{kit} + \alpha * (2 * (A_{ij} - P_{ij}) * U_{kit})$$

Where α is the learning rate.

7: goto step4 until the RMSE is minimum.

Explanation:

This algorithm takes a m*n rating matrix as input and produces the Root Mean Square Error of Existing and Predicted ratings.

3.2.2 Regularized Singular Value Decomposition (RSVD)

Algorithm 6 :Regularized Singular Value Decomposition (RSVD)

1: Form the input matrix $A \epsilon R^{m*n}$ from the given dataset.

2: Calculate the indicator matrix $I \epsilon \{0, 1\}^{m*n}$ that indicates which movies are rated by users.

3: A is given as input to SVD to get the feature matrices $U \epsilon R^{k*m}$ and $M \epsilon R^{k*n}$, where k is no. of features.

4: Calculate the prediction matrix as follows,

$$p(U_i, M_j) = \begin{cases} a & \text{if } U_i^T M_j < 0 \\ a + U_i^T M_j & \text{if } 0 \leq U_i^T M_j \leq b - a \\ b & \text{if } U_i^T M_j > b - a \end{cases}$$

where p is the prediction function which takes U_i, M_j are the feature vectors as arguments and computes the prediction value which lies in the range of (a, b).

5: Calculate the RMSE from the obtained prediction matrix.

6: To optimize the error, we use gradient descent approach i.e. the partial derivative of the squared error with respect to each parameter U_{ki} and M_{kj}.

$$U_{ki(t+1)} = U_{kit} + \alpha * (2 * (A_{ij} - P_{ij}) * M_{kit} - \beta * U_{kit})$$

$$M_{ki(t+1)} = M_{kit} + \alpha * (2 * (A_{ij} - P_{ij}) * U_{kit} - \beta * M_{kit})$$

Where α is the learning rate and β is regularization coefficient.

7: goto step4 until the RMSE is minimum.

Explanation:

This algorithm takes a m*n rating matrix as input and produces the Root Mean Square Error of Existing and Predicted ratings.

Chapter 4

Analysis and Results

Dataset:

Here, we have taken the Movie lens dataset which consists of 100,000 movie ratings in the range of 1-5 given by 943 users for 1600 movies. After that we have considered 300, 3000 and 30000 ratings as different instances.

Error Metrics:

The performance of Collaborative Filtering can be measured by the error between prediction values and the ground-truth. A common and efficient measure is Root Mean Square Error (RMSE). Consider the prediction matrix $P \epsilon R^{m*n}$ and the ground-truth answer $A \epsilon R^{m*n}$. Let $I \epsilon \{0,1\}^{m*n}$ be the indicator of A.

$$RMSE(P, A) = \sqrt{\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(A_{ij} - P_{ij})^2}{\sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}}}$$

Results:

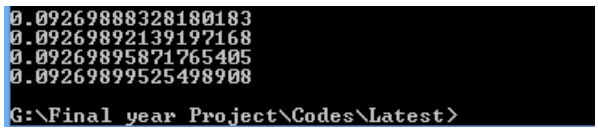

Figure 4.1: RMSE of SVD for dataset of size 300

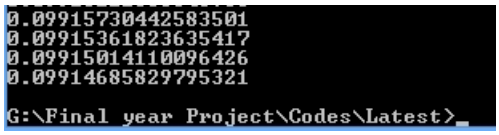

Figure 4.2: RMSE of SVD for dataset of size 3000

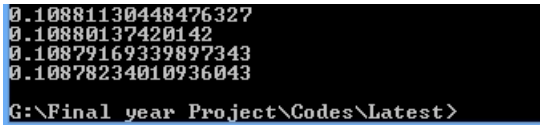

Figure 4.3: RMSE of SVD for dataset of size 30000

Figure 4.4: RMSE of RSVD for dataset of size 300

Figure 4.5: RMSE of RSVD for dataset of size 3000

Figure 4.6: RMSE of RSVD for dataset of size 30000

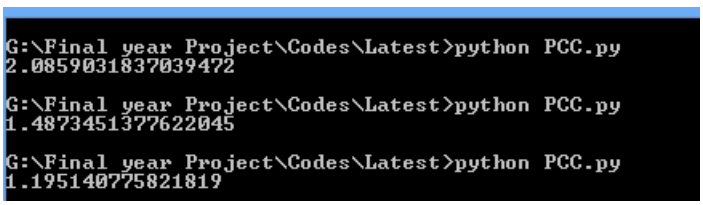

Figure 4.7: RMSE of Pearson for all datasets of size 300,3000 and 30000

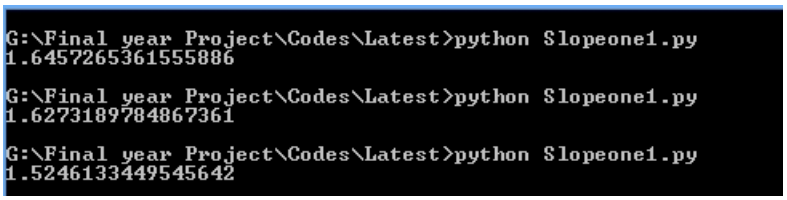

Figure 4.8: RMSE of Normal Slope one for all datasets of size 300,3000 and 30000

Figure 4.9: RMSE of Weighted Slope one for all datasets of size 300,3000 and 30000

Figure 4.10: RMSE of Bipolar1 for all datasets of size 300,3000 and 30000

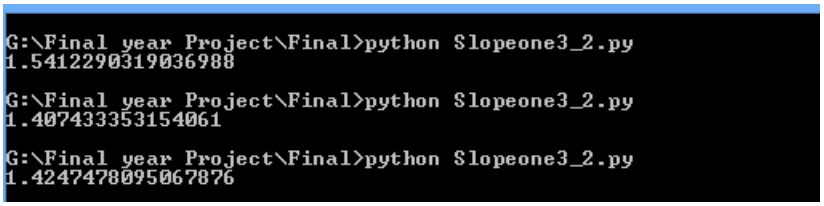

Figure 4.11: RMSE of Bipolar2 for all datasets of size 300,3000 and 30000

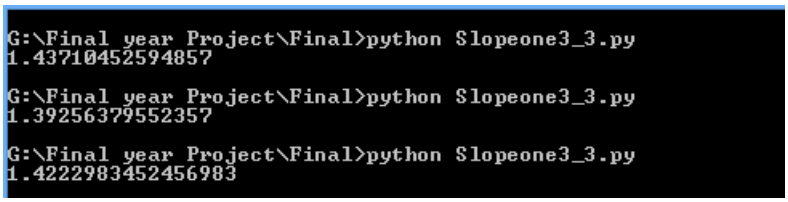

Figure 4.12: RMSE of Bipolar3 for all datasets of size 300,3000 and 30000

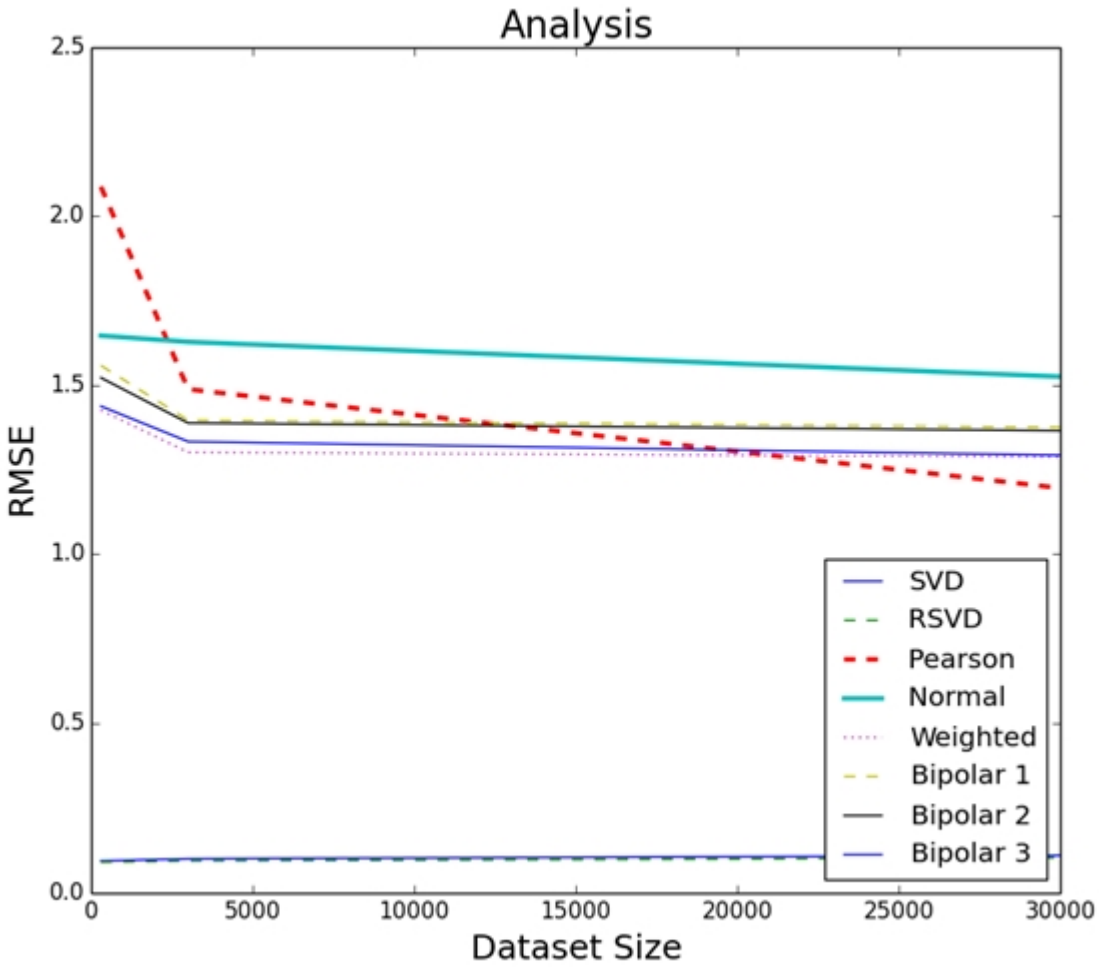

Figure 4.13: RMSE of All algorithms for all datasets of size 300,3000 and 30000

Chapter 5

Conclusion and Future Work

In this thesis we proposed different techniques for the calculation of the threshold value in the bi-polar slope one algorithm apart from the user average which has been used. We used median and min-max procedure for calculating the threshold value and it gives better performance i.e. a lower value of RMSE. We have conducted a comparative analysis of the different memory based and model based algorithms and deduced that the accuracy of the Model based algorithms are better when compared with Memory based algorithms. This is because we train the data in case of model based algorithms. But memory based algorithms are easier to implement and are faster when compared to the Model based algorithms although their performance is lesser. Memory based algorithms deals better with the problem of sparsity and the scalability when compared to that of Model based algorithms.

In future research can be done on implementing a hybrid approach which consists of both model based and memory based algorithms in order to increase the performance of the system as well overcome challenges such as sparseness and scalability.

Bibliography

[1] Chih-Chao Ma., A Guide to Singular Value Decomposition for Collaborative Filtering, Proceedings of seventh IEEE International Conference on E-commerce Technology, pp:1-7, 2005.

[2] Daniel L. and Anna M., Slope One predictors for Online Rating-based Collaborative filtering, Proceedings of SIAM Data Mining (SDM'05), Newport Beach, California, pp: 1-4, 2005.

[3] Paterek A., Improving regularized Singular Value Decomposition for collaborative filtering, Proceedings of KDD Cup and workshop , San Jose, California, USA, pp:5-8, 2007.

[4] Sheng Z., Weihong W., James F. and Fillia M., Using Singular Value Decomposition Approximation for Collaborative Filtering, Proceedings of the Seventh IEEE International Conference on E-Commerce Technology , pp:1-2, 2005.

[5] Tongqiang J. and Wei LU., Improved Slope One Algorithm Based On Time Weight, Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE), pp: 1-2, 2013.

[6] Drineas P., Kerenidis I. and Raghavan P., Competitive recommendation systems, Proceedings of the 34th ACM symposium on Theory of computing, pp: 82-85, 2002.

[7] Breese J., Heckerman D. and Kadie C., Empirical analysis of Predictive Algorithms for collaborative filtering, Technical Report, Microsoft Research, May, pp:1-8, 1998.

[8] Achlioptas D. and McSherry F., Fast computation of Low rank approximations, Proceedings of the 33rd Annual Symposium on Theory of Computing, pp:3-4, 2001.

[9] Rong J., Joyce Y. and Luo S., An automatic weighting scheme for collaborative filtering, Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, July, pp: 1-8, 2004.