CHAPTER 1

INTRODUCTION

1.1 Recommender System

A recommender system (or a recommendation engine) is a tool that is developed to predict what a user may or may not like amongst a given set of items. They prove to be an appropriate alternative to search fields as they help discover diverse products and contents that a user may not come across otherwise. Chris Anderson quoted in "The Long Tail" that "We are leaving the age of information and entering the age of recommendation".

Recommender systems can solve the problem of making personalized recommendations for information, products or services by applying knowledge discovery tools. This is done by applying "Collective Intelligence"[15]. Collective intelligence refers to the combination of knowledge, experience and insight of thousands of people to create a perspective rather than relying on a single person's perspective.

So what we do in most of the recommender systems is that we provide some learning / training data to the recommendation engine, it then analyses that information and applies some machine learning algorithms to it to help generate a set of N items that will be of interest to a certain user(top-N recommendation problem) or predict the chances with which the chosen item will be of interest to a certain user(prediction problem)[23].

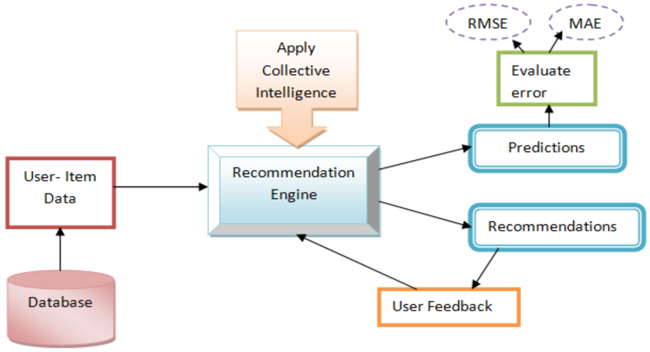


Figure 1Outlook of a recommender system

1.1.1 Uses of Recommender Systems

With the vast amount of information growth in the world, information is increasing more rapidly than our ability to process it. This has led to the problem of information overload. The feeling of being overwhelmed by the large number of new books, journal articles and conference proceedings coming out each year is pretty common. The earlier concepts of information retrieval have been turned upside down into the concepts of information filtering. Thus, the methods to help find the resources of interest have attracted a lot of attention from vendors and researchers all around the world. Recommender Systems provide one of the mechanisms to achieve this goal of effective information filtering.

The goal of a recommendation engine is to generate meaningful recommendations of items or products that might interest a collection of users. In recent years, recommender systems have been deployed in vast number of applications such as movies, e-commerce sites, TV programs, or music a user will find enjoyable, identify web pages that will be of interest or even suggesting alternate ways of searching for information[11].

Thus, a recommender system improves the user experience on a particular topic by providing a personalized environment around his or her search result. Thus the key roles of a recommender system[10] can be summed up as following :

i. Prediction : It is a numerical value, P_{ui} that expresses the computed value by feeding in the data into the recommender system for user u and item i.
ii. Recommendation : It is the list of N items $I_r \in I$, where I is the set of all items. The list I_r is the list of items to be recommended to a user u such that u has not already purchased the items in that set.

As a result, recommender systems are emerging as an interesting area of research and various researchers are proposing new and innovative techniques for the design of efficient recommendation engines.

1.1.2 Architecture of Recommender Systems

Below steps describes the basic architecture of a recommender system. It performs the following tasks sequentially:

i. Input data and convert it into the 2-D user-item matrix.
ii. Data Preprocessing- Removal of outliers, noise and global effects.
iii. Applying appropriate machine learning model to generate a utility function that automates the prediction of how a user will rate an item.
iv. Making recommendations to users based on the predictions made.
v. Collecting user feedback and incorporating improvements in recommender model.

A typical recommender system aims at evaluating the best possible predictions that a user will make for an item. In order to improve the prediction results, various models of machine learning are combined to make a hybrid system that gives better results.

1.1.3 Recommender System Evaluation Criteria

Prediction Accuracy: For a recommender system, prediction accuracy is by far the most discussed property. A prediction engine lies at the base of the recommender system. This engine is used to predict user opinions over items. A basic assumption that is held is that a recommender system making more accurate predictions will be preferred by a user. Thus, many researchers are exploring for algorithms that provide better predictions.

In-order to evaluate the accuracy of the prediction engine, following metrics isused:

1) RMSE: This is the most popular metric to evaluate accuracy of predicted ratings[15]. The system generates $\hat{r}_{ui}$ for a test set T of user-item pairs (u, i) for which the true rating r_{ui} is known. The formula for this metric is given by :

$$RMSE = \sqrt{\frac{\sum_{(u,i)}(r_{ui} - \hat{r}_{ui})^2}{n}}$$

Where n is the number of ratings of all users.

2) MAE: It is the measure of the deviation of the predicted recommendations from their true user-specified values. MAE is measured by summing up the absolute errors of the *n* corresponding rating prediction score and computing their average. The formula is as follows[10]:

$$MAE = \frac{\sum_{(u,i)}|r_{ui} - \hat{r}_{ui}|}{n}$$

3) Relative Error Method: Relative error is the absolute error that has been divided by the magnitude of the exact value. RE[25] is measured by summing up the squares of the absolute errors and then dividing by the squares of the actual values.

$$RE = \sqrt{\frac{\sum_{(u,i)}(r_{ui} - \hat{r}_{ui})^2}{{r_{ui}}^2}}$$

1.2 Motivation

The web has grown to be a crucial part of the global-village and its emergence paves a way to break the diverse barriers across global boundaries and physical variations. Embracing the web into our daily life activities has become inevitable. A large number of people rely on the web for various purposes like placing their own views and reading other people's views to commence their daily activities like e-learning, e-banking, e-library, e-commerce and many more.

However, the challenge of matching consumers with most appropriate products effectively and efficiently and to help them in the decision making process has increased the importance of online recommendation systems. Personalized recommender system can infer personal taste and recommend a list of items for a special user, provided prior knowledge of user purchase/rating profile. An efficient recommendation strategy not only enhances the user satisfaction and experience, rather also affect the results for a retailer in a positive manner. E-commerce leaders such as Amazon, Netflix, Google, TiVo and Yahoo are all adopting product recommendation engine to improve and enhance user experience [5].

Over the years, various techniques for building a recommender system have been developed that utilize either demographic, content or historical information. Among these, Collaborative Filtering that relies on historical data is probably the most successful and widely used recommender technique [11]. However, this traditional Collaborative Filtering technique poses a lot of key challenges that are to be dealt with. These are sparsity, scalability and cold-start[28]. To deal with these challenges, there is a need to build a system that is capable of diluting these challenges in order to give better results. Hybrid Recommender systems are becoming more and more famous that helps to combine the traditional methods with various machine learning models to deal with the key challenges of traditional recommender systems.

From the above discussion it is evident that there is a need for the development of an intelligent recommendation engine which is capable of adapting with the environment and also provides best possible results to the users according to their personalized needs. This has been the motivation to pursue research in the area of recommender system in order to address the issues present and design an intelligent recommender system.

1.3 Work by Other Researchers

Recommender Systems have been a hot topic of research from past few years. Recommendations and predictions have been improved by various techniques in previous researches.

ZeinabSharifi, Mahdi Nasiri and MansoorRezghi [1] proposed a recommender system using the idea of replacing the zero value of data with user median, item median and total median of ratings. SVD approach is then implemented and then evaluated through error metrics.

In [3], Robert Bell and Chris Volinsky have mentioned a recommender system based on matrix factorization model that uses user and item biases.

Qilong Ba, Xiaoyong Li and ZhongyingBai [4] have proposed a new recommender that combines K-means clustering and SVD algorithm. The users are classified using the clustering algorithm and then matrix is decomposed using SVD. Finally the results obtained from the two are fused together.

Sunitha Reddy and Dr. T. Adilakshmi [13] have proposed a music recommendation system that uses SVD as a technique for building the recommender and then using the Euclidean distance to identify the nearest items.

YiBoRen and SongJie Gong [7] devised a recommender that’s based on SVD smoothing. This technique predicts the item ratings that have not been implied by the users. Pearson Correlation similarity is then used to find out the target users and hence produce recommendations.

1.4 Thesis Organization

The remaining sections of the thesis are organized as follows:

Chapter 2 provides description of various recommendation engine models. It gives an insight to the advantages as well as disadvantages of the available techniques.

Chapter 3 proposes the formal Problem Statement and the approach chosen.

Chapter 4 describes the proposed solution along with the the detailed explanation of the novel Truncated SVD with conditional Collaborative Filtering Recommender System.

Chapter 5 presents Algorithm that has been designed for the proposed recommendation model, along with its implementation details.

Chapter 6 shows the evaluation of the proposed recommender system. It also compares the performance of Truncated SVD with conditional Collaborative Filtering Recommender System with two popular recommender systems:IBCF, SVD based recommender.

Chapter 7 concludes the thesis and depicts the possible improvements in this Research work in future.

CHAPTER 2

LITERATURE REVIEW

A literature survey on the existing literature about recommender systems proposed by researchers in the past few years is depicted here. A detailed description is provided for each recommender model surveyed, including the improvements with respect to previous models, basic technique incorporated and evaluation of the model.

2.1 Collaborative Filtering Based Recommender Systems

Collaborative filtering is also called as social filtering because it filters information based on the recommendations of other people. The idea behind this approach is that people who have agreed in their evaluation of certain items in the past are likely to agree again in the future.

The low- tech way to get recommendations for movies, entertaining web sites and various other products is to ask your friends. Some people are regarded with better taste than others, which could only be learned over time with the observation whether they like the same things. With the large amount of data available, this technique might have glitches as a small group of people may not be aware of all the options available. Thus, the concept of collaborative filtering emerged. It usually works by searching a large group of people and then selecting a smaller set of similar users to a particular user. Then, suggestions are made to the user based on a ranked list of those similar users[12].

In a typical CF based scenario, we have a set of m users $U = \{u_1, u_2, \ldots, u_m\}$ and n items$I = \{i_1, i_2, \ldots, i_n\}$. Each user u_i has a certain list of items for which it has expressed his opinions. Generally it is expressed within a certain scale and this particular rating can be called as a rating score. A user-item matrix can be derived from these user ratings. Further, the system matches the user's ratings against other users and find the people with most similar taste. And then make recommendations to the user based on the items liked by the similar users [10].

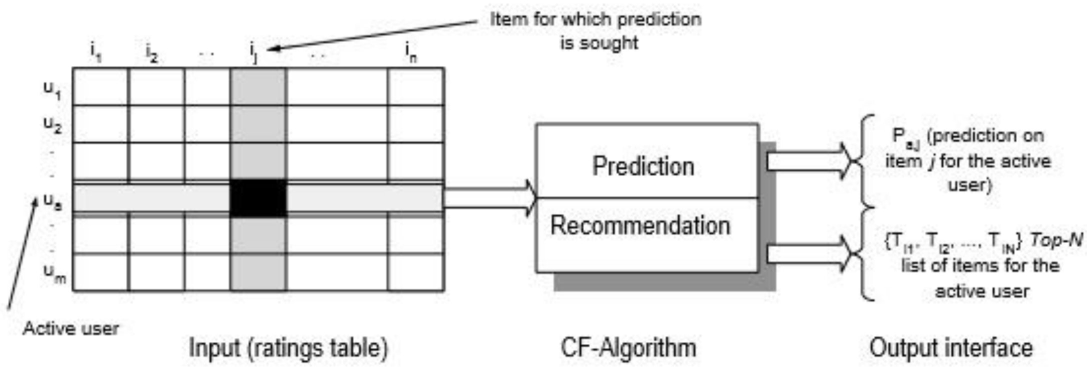


Figure 2The Collaborative Filtering Process[10]

The above figure describes the Collaborative Filtering technique for recommender systems. It depicts an input user-item matrix that helps generate recommendations and predictions for the users.

There are two methods in CF as UBCF and IBCF. These could be explained as follows:

- User-based CF: In the user-based approach the algorithm produces a rating for an item$i$ by a user u by combining the ratings of other usersu' that are similar to$u$. Similar here means that the two user's ratings have a high Pearson correlation or cosine similarity.

- Item- based CF: In the item-based approach we produce a rating for$i$ by$u$ by looking at the set of itemsi' that are similar to i (in the same sense as UBCF) that$u$ has rated and then combines the ratings by$u$ of i' into a predicted rating by u for i.

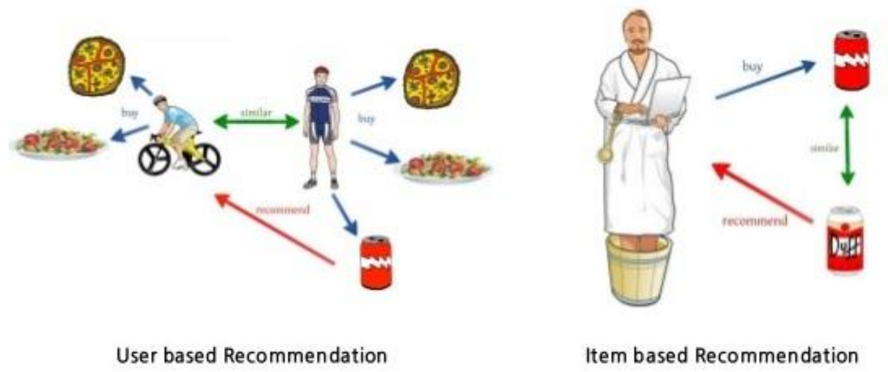


Figure 3 Difference between UBCF and IBCF

Collaborative Filtering methods have been classified into two broad categories by the researchers. These two main categories are memory-based and model-based methods. They have been explained further[27]:

Memory based CF: Memory based algorithms uses the complete user-item matrix to generate predictions. Statistical techniques are used by these systems. These techniques aim at finding a set of users (known as neighbors) which are similar to the target user. Once a neighborhood of a user is formed, these systems aims at reproducing predictions and recommendations.

Model based CF: These algorithms provide recommendations by first building a model of user ratings. Algorithms of this type take a probabilistic approach and make predictions using the user's rating data on other items. The model uses machine learning algorithms like Bayesian network, clustering, Rule-based etc.

Collaborative Filtering are the most traditional technique that are used to implement recommender systems. Though, they project the oldest mechanism of making the recommender systems, still they are widely used even now due to their high accuracy results. Using anyone of the CF technique can lead to a lot of key issues with the recommender. These are :

- **Sparsity:**The number of items to be evaluated is quite large. A single user may rate only a very small section of items available. CF algorithms aremainly based on the similarity measures computed over the co-rated set of items leading to large levels of sparsitythat can lead to less accuracy.

- **Scalability:** CF algorithms seem to be efficient in filtering in items that are interesting to users. However, they require computations that are very expensive and grow nonlinearly with the number of users and items in a database.
- **Cold-start:** An item cannot be recommended unless it has been rated by a number of users. This problem applies to new items and is particularly detrimental to users with eclectic interest. Likewise, a new user has to rate a sufficient number of items before the CF algorithm be able to provide accurate recommendations.

In order to deal with the above issues Xiao Yan Shi, HongWu Ye and SongJie Gong [28] combined these two CF methods and made a hybrid out of it. Using only a single CF technique takes only one-directional information from the user-item ratings matrix to generate recommendations. It means that the method may use only half of the total information from the given dataset. The method proposed uses IBCF to form a dense user-item matrix and then recommends using the dense matrix by applying UBCF. The experimental results showed that combining the two CF techniques resulted in better performance in terms of dealing with sparse matrix. But the scalability problem still persists.

2.2 Content based Recommender System

In content based RS, the recommendations are based on the content of the item rather than on users opinions on the items or their interaction with them. They tend to use a machine learning algorithm to build a model according to a user's preference. These preferences are evaluated based on the features and descriptions of the content. In this, the system recommends those items that a user may have liked in the past. A pure content based RS makes recommendations on the profile built of the user. This profile is build using the users past experiences [9].

The contents of an item can be explicit attributes or characteristics of an item. For example, for a movie, we can have its attributes as 'Genre', 'Year', 'Director' etc. This technique is used with the text-based products where items to be recommended are described by their associated features.

To evaluate the features of an item, a profile of a user $c$is generated containing preferences of that user. These preferences are obtained by analyzing the content of previous items and using keyword analysis techniques on them. Thus,

$$ContentBasedProfile(c) = \{w_{c1}, w_{c2}, \dots, w_{ck}\}$$

where weight w_d denotes the importance of keyword k_i to user c.

Utility Function $u(c, s)$ is usually defined as :

$$u(c, s) = score\big(ContentBasedProfile(c), Content(c)\big)$$

The advantages of this technique are that it has no cold-start or sparsity problems. Also, recommendations are made to users who have unique tastes and unpopular and new items also get a chance to be recommended. A detailed explanation can be provided as to why a specific item i is being referred to user u based on its content features [9].

Disadvantages of this technique are:

- Content is required that can be encoded as meaningful features.
- All items are not amenable to the methods of feature extraction.
- Quality judgments of other users cannot be extracted.
- Over-fitting problem.

AnandShankerTiwari, Abhay Kumar and Asim Gopal Barman proposed a book recommender system using content based filtering along with collaborative filtering in 2014. Further, they have performed association rule mining in order to find out the rules that recommend books to the users. The book recommender system has taken into account a lot of parameters like the content of the book and the quality of the book by doing collaborative filtering of ratings of other buyers as well. It also uses an associative model to give stronger recommendations. However, it still has scalability issues[9].

2.3 Neighborhood Based Recommender System

Neighborhood based approaches tends to capture local associations in the data. Recommender systems based on neighborhood uses the common principle of word-of-mouth in an automated fashion. Thus, to evaluate the value of an item, a user relies on the opinions of other like-minded people or other trusted sources[5].

Following are the advantages of Neighborhood based Recommender systems:

- Simplicity – These methods are intuitive and relatively simple to implement.
- Justifiability – They provide a concise justification for the computed predictions.
- Efficiency – Unlike the model-based systems, no costly training is required. They can be pre-computed offline.

KNN algorithm serves as the most common approach to build such neighborhood based recommender system[17]. The main idea that is followed here is that given a set of users$U = \{u_1, u_2, \dots, u_m\}$ and a set of items $I = \{i_1, i_2, \dots, i_n\}$, the main objective remains to find the 'k' nearest neighbors of a given user 'u' and then make recommendations based on those nearest neighbors preferences.

In-order to find k-nearest-neighbors for a user 'u', we calculate the similarity between the user 'u' and all the other users u'. Once the similarity vector is obtained that contains the weight of their similarity, it is sorted in decreasing order. The top k users having the maximum similarity to user 'u' are extracted out. The similarity function may use anyone of the popular methods like the Euclidean distance, Pearson correlation, Manhattan distance etc[20].

HaoJi, Xuan Chen, Miao He, Jifeng Li and ChangruiRen introduced a propagated neighborhood based collaborative filtering recommender system in 2014[5]. The motivation for this model came from the fact that the utilization of KNN method for single item/user always misses some nature neighbors which could be due to the presence if noisy data or due to data sparsity. Thus KNN alone results in poor prediction accuracy. A novel method of two levels of propagated neighborhoods is introduced in PNCF in contrast to the traditional KNN method. The method proposed faces challenges like sparsity issues for new users.

2.4 Trust Based Recommender System

Trust Based Recommender systems use trust as a way to give more weights to some users. The goal of this recommender system is to generate personalized recommenders by evaluating the opinions of other users in the trust network.

Trust is a very complex entity. It requires belief and commitment. Concepts like context, similarity, reputation, personal background and history of interaction are involved. Trust is therefore used as a similarity measure in the context of recommender systems[8].

GuibingGuo, Jie Zhang and Neil YorkeSmith proposed a trust based recommender system in 2013[8]. Their system was merged with the SVD algorithm and item based collaborative filtering. The system was successful in mitigating the cold-start and sparsity problems. Some observations were made on this model:

Observation 1: Trust information is very sparse , but it is complimentary to rating information.

Observation 2: Under the concept of trust-alike relationships, a user's ratings has a weak positive correlation with the average of its social neighbors. Also the user has a strong positive correlation with its trusted relationship.

Problem with the trust based recommenders is that not all kinds of data has the trust data available and extracting this trust data can be very expensive.

2.5 Case Intelligence Based Recommender System

Case based reasoning has originated from human experience learning. Analogy plays an important role for reasoning model. We can describe it as:

Object A has attributes a, b, c, d.

Object B has attributes a, b, c.

This may suggest that B might have attribute'd' as well. Analogy helps to process the new information entering our brains. The new knowledge is compared with the existing one that has been understood and that is how it then gets stored in a human brain[21].

Thus case intelligence is a very comprehensive expression that is an integrated representation of logics, human sense and creativity and poses the ability to acquire user's preferences from former stored cases.

CBR uses a wide variety of analogy measures such as the method analogy, graphic analogy, legend analogy etc. A mapping-analog is created that helps to compare any two similar things and tends to find their similar relations at some level so that new issues can be solved through appropriate knowledge transformation. In general, CBR uses four processes which are Retrieve, Reuse, Revise and Retain. For the purpose of recommender Systems, only Retrieve is used.

The advantages of this technique are:

- Helps provide a generic methodology that builds knowledge based systems rather than isolated technique that is capable of solving only specific tasks.
- It has great flexibility which indicates that we can use it as a hybrid system.

Yanhai Zhao and Jianyang Li proposed a personalized recommender system based on CBR in 2010 [16]. Their method had a logical outlook such that the conclusion is chained with preconditions, which is relative to the customer's experience. Therecommender explores all the kinds of knowledge from the rich case library and tend to acquire the solution from the former similar problems. Following disadvantages could be seen:

- Processing personalized information in case of inadequate amount of data available.
- Self-learning is difficult and expensive.

2.6 Matrix Factorization Based Recommender System

Matrix Factorization methods characterize both items and users by vectors of factors inferred from item rating patterns. A recommendation is based on the high correspondence function between user and item factors. This method relies on matrix type of input data where one dimension represents the users and the other represents the items. It includes explicit input by users regarding their interest in products. These explicit inputs by users are referred to as ratings.

Generally, a sparse matrix is present since a single user might have rated only a small set of possible items[13].

The matrix factorization method can be formalized into a number of steps given below:

i. Users and items are mapped to a joint latent factor sparse with dimensionality *f.*
ii. Inner products in that space are model by the user- item interactions.
iii. Each item *i* is associated with a vector $q_i$and each user *u* is associated with a vector p_u
    - $q_i$represents the extent to which the item possesses those factors.
    - $p_u$measures the extent of user's interest in that item.
    - The interaction between user *u* and item *i* is calculated by the dot product $q_i^T p_u$.
iv. Thus the user *u*'s rating for item *i* denoted by r_{ui} is approximated as $\hat{r}_{ui} = q_i^T p_u$.
v. The major challenge is the computation of the mapping of each item and users to the factor vectors q_i , p_u .
vi. The above steps are performed for all users and items.
vii. A low-dimensional representation of the original matrix is computed.
viii. A dimensionality reducing algorithm helps to compute the factorized matrices that further tries to reduce it using some optimization technique.
ix. Resultant matrices are further used to compute the predictions.

Predicting Task :

i. The recommendation score for any set of user and item is computed by the obtained resultant matrices.
ii. A rating score r_{ui}can be calculated by the dot product of row *i* of *q* and column *u* of *p*.

$$\hat{r}_{ui} = \overrightarrow{q_i} p_u$$

Disadvantages :

- Missing data in high portions caused by sparseness in the user-item rating matrix.
- When the knowledge of matrix is incomplete, conventional SVD is undefined.

Some of the common matrix factorization methods are gradient descent, SVD, alternative least squares etc.

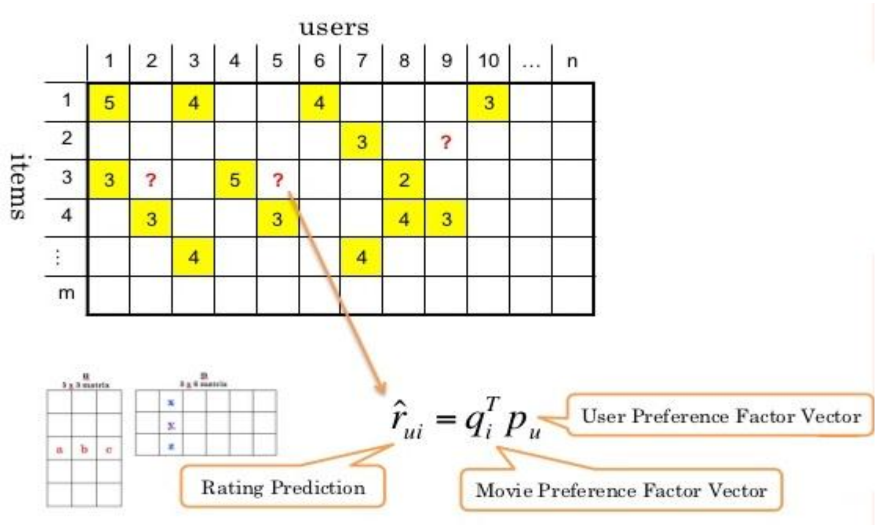


Figure 4Making Prediction as Filling Value[3]

SongJie Gong, HongWu Ye and YaE Dai proposed a recommender system combining the standard SVD and item based CF in 2009[2]. In their method they have used the symmetric similarity measures like pearson correlation, cosine method and adjusted cosine method. Their proposed work shows that collaborative filtering combined with a matrix factorization method helps lower down the prediction error as compared to traditional CF methods.

2.7 Hybrid Recommender system

Hybrid recommender systems combine two or more recommendation techniques in order to increase the overall performance.The main idea is using multiple recommendation techniques to suppress the drawbacks of an individual technique in a combined model.The taxonomy is based on the hierarchy and input/output relations of recommenders. The hybridization methods can be classified into the following:

- **Weighted:** In weighted hybridization, a mixture of experts framework is constructed for decision level fusion.Rating for a given item is computed as the weighted sum of ratings produced by a pool of recommenders.The weights are determined by training on previous ratings of the user and they may be adjusted as new ratings arrive.The performance of recommenders among the set of possible items is assumed to be homogeneous among the space of input items.This assumption does not hold for collaborative filters since CF's perform weaker on items with few ratings.However, despite the claim in the paper, this assumption is not crucial in all weighted hybrids.

- **Mixed:** Mixed hybrids combine recommendations of multiple systems rather than using them to predict rating of an individual item.Mixed hybrids can, also avoid problems of individual systems such as the new item problem. In mixed hybridization, the individual performances do not effect the overall systems performance at a local region.
- **Switching:** In switching hybridization, the system switches to one of the recommenders according to a heuristic reflecting the recommender’s ability to produce a good rating.The switching hybrid can avoid problems specific to one method, e.g. the new item problem of content-based recommenders, by switching to a collaborative recommendation system. A new level of complexity is introduced into the hybrid, deciding the switching criterion.
- **FeatureCombination:** In feature combination, the rating produced by one recommendation system is fed into another as a feature.The natural order is feeding the rating of a collaborative recommender to a content based recommender since content based systems act on feature representations of items.
- **FeatureAugmentation:** The strategy of feature augmentation is similar in some ways to feature combination. But instead of using raw features from the contributing domain, feature augmentation hybrid’s support their actual recommender with features passed through the contributing recommender. Usually, feature augmentation recommender are employed when there is a well-engineered primary component that require additional knowledge sources. Due to the fact that most applications expect recommendations in realtime, augmentation is usually done offline. In general, feature augmentation hybrids are superior to feature combination hybrids.
- **Cascade:** Cascade systems apply iterative refinement procedure for constructing a preference order among items.At every stage, a recommender takes a set of items that had equal preference in the higher level, and order them into bins of equal preference.At each

step, a finer refinement is obtained.The cascade systems are efficient and tolerant to noise due to the coarser-to-finer nature of iteration.

- **Meta-Level:** Meta-level hybrids feed the constructed model by one recommender to another as input.The constructed model is denser in information when compared to a single rating. Hence in meta level hybrids, more information is carried from one recommender to another.Meta level hybrids can address the sparsity problem of collaborative filters by providing models from a previous as inputs.

2.8 Problem Statement and Approach

Following section gives the problem statement of this research and also prives the approach used. The scope of this research is also given below.

2.8.1 Problem Statement

The stand-alone traditional recommender systems cannot deal with data sparsity, scalability and cold-start problems [28]. This research aims at building a system that helps preprocess the data so that all the unknown and missing values could be generated through some processing function in order to deal with the cold start problem, so that no item gets an unbiased recommendation result on the cost of being new to the system. Also, to deal with data sparsity, focus is on applying methods that helps evaluate the missing values by applying some predictive technique.

Therefore, problem of the thesis can be stated as:

Development of an intelligent and adaptable recommendation engine using Truncated SVD and Conditional probability based Collaborative Filtering that deals with sparsity, cold-start and scalability issues.

2.8.2 Scope of Work

For the development of effective recommender systems, various parameters have to be considered in advance. Several key challenges like data sparsity, scalability and cold-start are to be dealt with in order to build an effective recommender system. Since, collaborative filtering methods are the most widely used ones, they still suffer from the above mentioned challenges.

To improve the scalability of collaborative filtering algorithms, the system needs to be devised in a way that it is able to incorporate the arrival of new users and items without increasing the performance time exponentially. SVD algorithm helps to deal with these issues by devising a singular value matrix that helps give the predictions for new users in a timely manner.

In order to deal with the cold-start problem, data processing is performed such that the results are not biased towards the already existing users and items in the system. For this issue, while recommending new items to users, the recommendations are computated based on the hybrid approach of SVD and conditional-probability based model. This helps generate unbiased and appropriate results[11].

Therefore, scope of work can be summarized as:

- Design a recommender System that takes input query from the user and processes it.
- The system should be trained with examples so that it performs well when addressed to unknown data.
- Evaluate the quality of results retrieved by the recommendation engine by calculating the error percentage between the retrieved and actual results.
- Comparison of the build system with the existing recommender systems of the same domain.

2.8.3 Approach

A hybrid recommender system consisting of matrix factorization and traditional collaborative filtering is built here. Performance of the new hybrid system is evaluated on the Movielens dataset consisting of 943 users and 1682 items and contains about a 100,000 ratings.

Among the various matrix factorization techniques, SVD is one of the best technique to make efficient recommenders [3]. Very huge matrices can be reduced to a very low-rank matrix with the use of SVD. Dimensionality reduction helps process huge datasets that were initially taking a very large amount of time.

The collaborative filtering method works on the past history of users. Among the two available collaborative filtering methods, item-based CF is more efficient than user-based CF [22].

Traditionally, item-based CF was carried out with the help of symmetric similarity measures like cosine method, Pearson correlation etc. The proposed system uses conditional probability as a measure of similarity computation between the items.

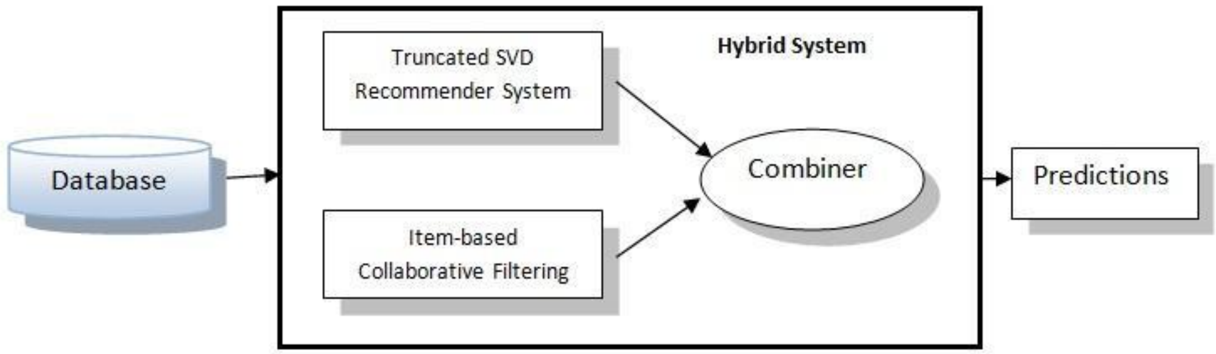


Figure 5Architecture of Proposed System

The above figure shows the basic architecture of the proposed system. It shows a two-part hybrid recommender system. The system combines the results of the two separate recommenders using a weighted scheme. The various parameters determining the success of the novel system proposed are determined through stepwise experiments. They reveal the optimal weight and parameter configuration for the proposed hybrid recommendation system.

CHAPTER 3

PROPOSED MODEL FOR RECOMMENDER SYSTEM

In this chapter, emphasis are done on dimensionality reduction algorithms in detail. It begins with the description of what is meant by dimensionality reduction. Further, singular value decomposition algorithm is explored in detail and how a given sparse input matrix is decomposed into its dense form using the SVD algorithm. Conditional probabilistic model is then explained in detail and why it is better than other similarity measures. An approach to build the item based recommender system using the conditional probability is also explained.

3.1 Dimensionality Reduction

In matrix factorization method, a large matrix M is decomposed into two matrices U *and* V such that UV is approximately equal to M. In dimensionality reduction, decomposition is done such that U has smaller number of columns and V has smaller number of rows, so each is significantly smaller than M. Together, U *and* V represent most of the information of M that is useful in predicting ratings of items by users[19].

3.1.1Singular Value Decomposition

SVD proposes an exact representation of any matrix through the elimination of less important parts of that representation. This is done to produce an approximate representation with any desired number of dimensions. Accuracy of the approximated matrix decreases with the decrease in the number of dimensions chosen.

Definition of SVD : Let M be an $m \times n$ matrix and the rank of M be r. Rank of a matrix is defined as the largest number of rows (or equivalently columns) that can be chosen for which no nonzero linear combination of the rows is the all-zero vector **0** (such that the set of rows or columns are independent). Then matrices $U, \sum, \ and\ V$ could be found as[20] :

1. U is a column-orthonormal matrix of dimension $\times r$, such that each of its column is a unit vector and the dot product of any two columns is 0.
2. V is a row-orthonormal matrix of dimension $\times n$, such that each of its row is a unit vector and the dot product of any two rows is 0. V is always used in its transpose form, so it is the rows of the matrix V^T that are orthonormal.
3. $\sum$is a diagonal matrix in which all the elements except the main diagonal are zero. The elements of $\sum$ are called the *singular values* of *M*.

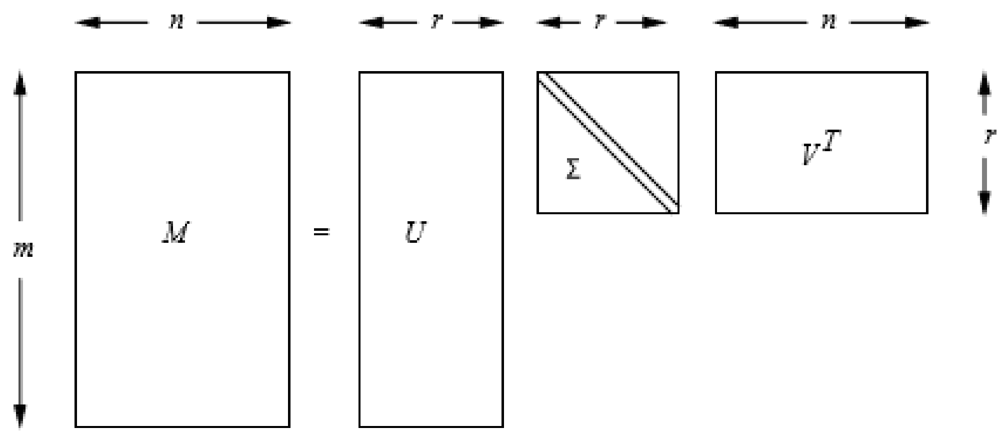


Figure 6 The form of a singular value decomposition[19]

The above figure shows the decomposition of matrix M into correspoing matrices $U, \sum$ *and* V according to the rank *r* of *M*.

3.1.2Dimensionality Reduction Using SVD

The matrices $U, \sum$ *and* V obtained by the SVD decomposition of original matrix *M* are also too large to store conveniently if *M* is very large. In case it is desired to represent *M* by its decomposed components, the best way to do that is reduce the dimensionality of those matrices further by some value. To do that , smallest of the available singular values are set to zero. If *s* smallest of the singular values are set to zero, then the corresponding *s* rows of U *and* V can also be eliminated[26].

Why Zeroing Low Singular Values Works?

The RMSE between the original matrix M and its approximation is minimized when the lowest singular values are dropped to reduce the dimensions. Since the square root is a monotone operation and the number of entries is fixed, the Forbenius norms of the matrices involved can be simplified and compared. Forbenius norm of a matrix M is denoted by $\|M\|$. It is the square root of the sum of the squares of the elements of M. If M is stated as the difference between a matrix and its approximation, then $\|M\|$ is proportional to the RMSE between the matrices.

How many singular values to be retained?

An efficient rule of thumb states that 90% of the energy in $\sum$ should be retained through the singular values present. This means that the sum of the retained singular values should evaluate to 90% of the sum of all the singular values.

3.1.3SVD Computation of a Matrix

The SVD of a matrix M is strongly associated with the eigenvalues of the symmetric matrices M^TM and MM^T. This association helps generate the SVD of M from the eigenpairs of the latter two matrices. To begin with the explanation of SVD of M ,$M = U\sum V^T$. Then,

$$M^T = (U\sum V^T)^T = (V^T)^T\sum^T U^T = V\sum^T U^T$$

Since $\sum$ is a diagonal matrix, transposing it has no effect. Thus,

$$M^T = V\sum U^T$$

Now, $M^TM = V\sum U^T U\sum V^T$. Since, U is an orthogonal matrix , so $U^T U$ is an identity matrix of appropriate size. Therefore,

$$M^TM = V\sum^2 V^T$$

The above equation is multiplied by V on both the sides, so we get

$$M^TMV = V\sum^2 V^T V$$

$V^TV$is an identity matrix as well because V is also an orthonormal matrix. Therefore,

$$M^T M V = V \Sigma^2$$

$\Sigma^2$is a diagonal matrix since Σ is a diagonal matrix. The entry in the$i$th row and column is the square of the entry in the same position of Σ . So, equation 3.6 states that V is the matrix of eigenvectors of $M^T M$ and Σ^2 is the diagonal matrix whose entries are the corresponding eigenvalues.

The same algorithm that computed the eigenpairs of $M^T M$ gives us the matrix V for the SVD of M itself. It also provides us the the singular values of this SVD, which could be computed by taking the square roots of the eigenvalues of $M^T M$.

U is also computed the same way V was found out. It could be started by this equation:

$$M M^T = U \Sigma V^T (U \Sigma V^T)^T = U \Sigma V^T V \Sigma U^T = U \Sigma^2 U^T$$

The above equation results in the following result:

$$M M^T U = U \Sigma^2$$

Thus, U is the matrix of eigenvectors of MM^T.

$MM^T$is a $m \times m$ matrix and $M^T M$ is a $n \times n$ matrix. Both m and n are at least as large as r, where r is the rank of the matrix M. Since r is the rank of the matrix , that means that there must be $m - r$ and $n - r$ additional eigen pairs in U, $\Sigma$and V. These additional values are not useful and maybe discarded.

3.1.4 Quering New Users

In this section focus is at how SVD can help answer certain queries efficiently, with good accuracy. Lets assume for example that a datset has 5 movies as its items. The original movie-rating data is decomposed into the SVD form. Quincy is not one of the people represented by the original matrix, but he wants to use the system to know what movies he would like. He has only seen one movie, The Matrix, and rated it 4. Thus, Quincy can be represented by the vector q = [4,0,0,0,0], as if this were one of the rows of the original matrix. By using a collaborative-filtering approach, Quincy would have been compared with the other users represented in the original matrix M. Instead, Quincy can be mapped into “concept space” by multiplying him by the matrix V of the decomposition. The aim is to find qV.Now Quincy has a representation in concept space, derived from, but different from his representation in the original “movie space.” One useful thing that can be done is to map his representation back into movie space by multiplyingby V T. Another sort of query that can be performed in concept space is to find users similar to Quincy. V can be used to map all users into concept space In general, the similarity of users can be measured by their cosine distance in concept space.

3.2 Truncated SVD

The singular value matrix $\sum$ is of the form $\begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$ where D is a diagonal matrix containing the singular values. D might be of the form :

$$D = \begin{bmatrix} \sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \sigma_3 & & & \\ & & & \sigma_4 & & \\ & & & & \ddots & \\ & & & & & \sigma_r \end{bmatrix}$$

Where $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_r \geq 0$ are the singular values of matrix M with rank r. Full rank decomposition of M is denoted by :

$$M_r = U_r * \sum_r * V_r^T$$

A reduced rank approximation of M also called as truncated SVD can be find out by setting all but the first *k* largest singular values to zero. This results in using only first *k* columns of *U* and *V*.

Use of Truncated SVD over Normal SVD:

There are some cases where the matrix *M* has one dimension quite bigger in comparison to the other. For instance, *m = 5* and *n = 10,000*(where *m* denotes the number of users and *n* denotes the number of items) such that *m<<n*. For such a case, most of the memory and computation required for the standard SVD are actually not needed. Instead a reduced version of it is preferred. This reduced version is called the Truncated SVD. This version of SVD is much faster and requires less memory to store the data.

3.3 Measuring the Similarity between Items

The effectiveness of the overall recommender system depends on the method used to compute the similarity between the various items. In general, the similarity between two items$i$ and $j$should be high if there exists a lot of customers who have purchased the two items, whereas it should be low if there are few such customers[11].

Two aspects can be highlighted here which are somewhat less obvious and may be overlooked a lot of times. These are :

- The first aspect corresponds to the fact that some customers are very active in comparison to some other customers who are very inactive. For example, consider two customers $C_1$and C_2. Both of these customers have purchased items$i$ and j.C_1has purchased only 5 additional items, whereas C_2 has purchased 50 additional items. The fact that both of them purchased items$i$ and j should not contribute equally while determining the similarity between this pair of items. Cases may arrive in which the co-purchasing information derived from customers that may have bought fewer number of itemsprove to be a more reliable indicator for the similarity of two co-purchased items than those customers who tend to buy a large number of items. This is all because the first group of people represents those consumers who are focused in certain product areas.

- The second aspect to be considered is whether or not the similarity between a pair of items should be symmetric $(i.e., sim(i,j) = sim(j,i)$ or not $(i.e., sim(i,j) \neq sim(j,i)$.The importance of this aspect arises when similarity needs to be computed between a pair of items that have been purchased at substantially different frequencies. For example, consider two items i and j such that i has been purchased much more frequently than j. Since there is a frequency difference, the number of times i would have been purchased along with j is much lower than the times i was purchased alone. From i's point of view the similarity with j is low because its co-occurrence with j is low. However for j, its similarity with i may be high. Thus, if an asymmetric similarity function is used $sim(i,j) < sim(j,i)$.

The advantage of an asymmetric function over a symmetric function can be stated as follows. In datasets that have no items that are frequently purchased by the majority of the customers, a symmetric similarity function would unnecessarily penalize the recommendation of those items whose frequency is relatively higher than the items that were currently purchased by the active user.

While studying similarity functions, two different types are present. One comes from the vector-space model and the other comes from the probabilistic model. The former one leads to symmetric similarity functions whereas the latter one leads to asymmetric similarity function.

3.3.1 Traditional Similarity Functions

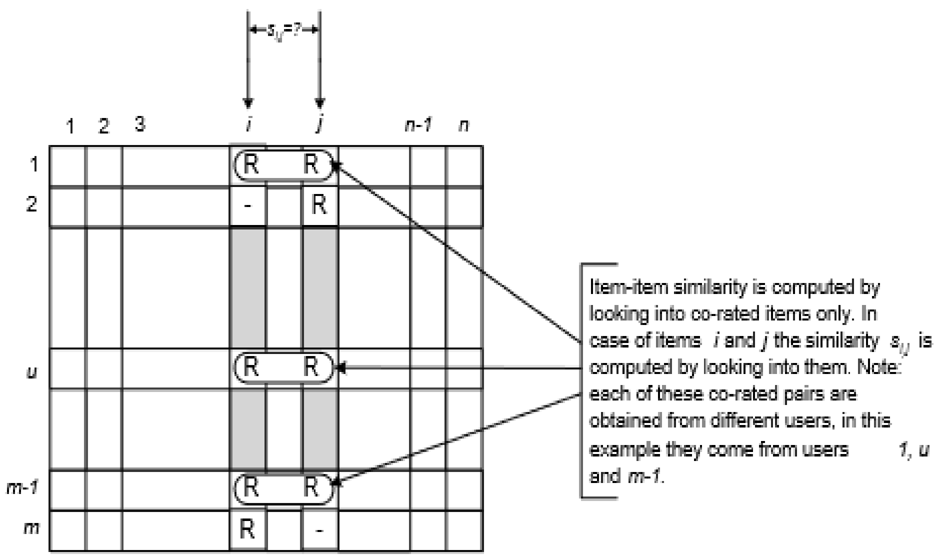


Figure 7Isolation of the co-rated items and similarity computation[11]

The above figure shows the computation of item-item similarity looking into co-rated items only. Each of the co-rated items comes from different user.

Cosine based Similarity:

In this case, items are treated as vectors in the m-dimensional user-space.The similarity between any two items are measured by computing the cosine of the angle between these two vectors. In the ratings matrix of size $m \times n$, the similarity between items i and j is denoted by $sim(i, j)$ is given by

$$sim(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}.\vec{j}}{||\vec{i}||^2 * ||\vec{j}||^2}$$

Correlation based Similarity:

In this case, the similarity between the two items *i* and *j* is measured by computing the Pearson-r correlation. In order to make the correlation computation accurate , the co-rated cases must be isolated as shown in Figure ##. Let the users who rated both *i* and *j* be denoted by *U*. The correlation similarity is given by

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_j)^2}}$$

$R_{u,i}$ represents the rating of user *u* on item *i* , $\bar{R}_i$ is the average rating of the i^{th} item.

Adjusted Cosine Similarity:

The basic cosine measure in the item-based similarity computation has a drawback that the difference in rating scale between users is not taken into account. The adjusted cosine similarity attempts to offset this drawback by subtracting the corresponding user average from each co-rated pair. Thus the formula of this scheme is given by

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_u)^2}}$$

$\bar{R}_u$ is the average rating of the u^{th} user's ratings.

3.3.2 Conditional Probability Based Similarity

Conditional probability $P(j|i)$ which states the probability of purchasing of item *j* given that *i* has already been produced is equal to the number of customers who have purchased both the items *i* and *j* divided by the total number of customers that purchased*i* . That is,

$$P(j|i) = \frac{Freq(ij)}{Freq(i)}$$

Where $Freq(X)$ is the number of customers that have purchased the items in Set X. In general $P(j|i) \neq P(i|j)$. This is thus used as a measure of similarity to generate asymmetric relations.

One of the main limitations of the asymmetric relation is that each item i will tend to have conditional probabilities with those items that are being purchased frequently. The problem can be corrected by dividing $P(j|i)$ with a quantity that depends on the occurrence frequency of item j.

$$sim(i,j) = \frac{Freq(ij)}{Freq(i) \times (Freq(j))^{\alpha}}$$

Here α may vary from 0 to 1. If $\alpha = 0$, $sim(i,j)$ becomes equal to $P(j|i)$. Whereas if $\alpha = 1$, it becomes to the formulation in which $P(j|i)$ is divided by $P(j)$.

3.3.3 Predictions From Item-Item Similarity

The prediction for a rating is one of the most important step for collaborative filtering technique. For that, the set of most similar items is isolated based on the similarity measure. Further the step is to look into the target user's ratings and obtain predictions.

Weighted Sum: The method computes predictions for an item *i* by a user *u* by evaluating the sum of the ratings given by the user *u* on the items that are similar to *i*. Further this sum is divided by the similarity measures of these similar items.Formula for this weighted sum is given by:

$$Prediction(u,i) = \frac{\sum_{all\ similar\ users,N} Similarity(i,N) *\ R_{uN}}{\sum_{all\ similar\ users,N} |Similarity(i,N)|}$$

The above formula helps compute a single prediction for an item I by user u. This formula is spanned across all the user-item pairs to generate the predictions for all the unknown ratings.

Chapter 4

IMPLEMENTATION

The chapter presents the proposed model for building an intelligent recommender system. This model is free from the problems of data sparsity, cold-start and matrix sparseness. This model uses matrix factorization method along with conditional probabilistic model to make predictions and recommendations. The proposed model helps to analyze the differences in any two users based on their purchasing behavior. Thus an asymmetric similarity function is being focused upon. Therefore, the proposed model is being called intelligent because it can differentiate between distinct users and make a personalized recommendation. The implementation details and evaluation of the recommender system will be discussed in chapters 5 and 6 respectively.

4.1 Problem Definition:

The aim of this report is to present a recommender system that analyses the given input data and then make predictions and recommendations. The predictions convey the computed ratings that a user would give to an item. Whereas the recommendations refer to those items that are referred to a user for future use that might be of his interest.

To deal with the various problems of collaborative filtering, matrix factorization method is used. This matrix factorization method helps generate low dimensional matrices of the original one. The decomposed matrices is used to generate predictions. Further item-item similarity is computed for all the items present. A hybrid recommender is made based on the combination of the prediction matrix and the item-item similarity prediction matrix. Top-N items are recommended to any user who wishes to know his areas of interests. Further, evaluation metrics are used to know the accuracy of the recommender. At last, the final evaluated results are compared with previously stated algorithms.

4.2 Important Steps Involved

This section gives an overview of the various important steps that shapes up the functioning of the proposed model. The detailed explanation pertaining to the implementation of each stepis explained in detail. Some of the steps even have a pseudocode written along with their explanation.

4.2.1 Smoothing

This step involves the conversion of the given data into an appropriate form that can be used by the recommender system. The data given in the format of unstructured file is first converted to a structured matrix. This matrix is called the user-item matrix. The data in the user-item matrix is very sparse[1]. To deal with this problem, an appropriate function is used to fill in the missing values. This function could be seen as :

Algorithm: Fill_in_sparse_matrix(X):

```
For each row i in X
find mode of X[i]
Store this value in mode_row[i]
For each column j in X
find mode of X[j]
Store this value in mode_column[j]
For each row i in X
        For each column j in X
                If X[i][j] =missing:
                        X[i][j]=average(mode_row[i], mode_col[j])
```

The above algorithm shows the imputation of the user item matrix with the avegare of the modal values of the users and items. Similarly, imputation is done with the mean and median values of users, items or both.

4.2.2 Employing the SVD model to reduce Dimensionality

SVD algorithm is applied to the processed matrix X that was created during the preprocessing part. The original matrix is decomposed into three new matrices U, S and V. An example showing such decomposition can be seen as follows :

	$Song_1$	$Song_2$	$Song_3$	$Song_4$	$Song_5$
$User_1$	2	0	0	0	0
$User_2$	0	4	0	1	0
$User_3$	1	3	0	0	0
$User_4$	0	0	0	1	1

Figure 8 Part of original Matrix X[13]

Matrix U

0.036697123	0.001598783	0.087417695	0.004824194
0.00504383	0.045794877	0.046870156	0.003087831
0.003826221	1.49E-04	1.27E-04	7.53E-04
0.004220399	0.001268878	0.10532634	0.021120419
0.001713249	0.003035872	4.57E-03	0.006122999

Matrix V

0.56444245	-0.024768703	0.82310289	-0.038961494
0.014803129	-0.031332759	0.02376704	4.13E-04
0.002896579	0.013851497	0.060281442	0.036031687
0.704171698	-0.704036471	-0.046819038	0.016944794

Matrix S

92.65579181	0	0	0
0	87.696145	0	0
0	0	56.97938947	0
0	0	0	27.5252295

Figure 9 Matrices U, V and S received as a result of SVD decomposition[13]

The above Figure 8 and Figure 9 shows the SVD decomposition of the matrix *X* into the three matrices *U, S and V*.

4.2.3 Using SVD to make Prediction Matrix

The matrices *U,S and V* received in the SVD decomposition are further used to create the Prediction matrix. For the creation of the same, dimensionality reduction is carried out in this step. For the following algorithm is used :

Algorithm: Create_prediction_matrix(U,S,V):

For each singular value is S
    Energy = energy + square(singular value)
Find k such that
    Energy_new = 90% of Energy
Compute new matrices $U_k,\ S_k and V_k$
For each row i in X
    find average of row i
    Store this value in A[i]
For each row i in X
    For each column j in X
        $P_{ij} = A_i + U_k\sqrt{S_k}(i) \times \sqrt{S_k}V_k^T(j)$

Here, P_{ij} refers to the prediction of rating of item *i* by user *u* .

4.2.4 Computing Prediction Matrix for Item-based CF

An important step towards making recommendations using collaborative filtering, is to compute the similarity matrix. In the proposed system, item-item similarity is calculated using conditional probability on the original user-item matrix R. The formula for such computation is :

$$sim(i,j) = \frac{Freq(ij)}{Freq(i) \times (Freq(j))^{\alpha}}$$

The above formula helps compute asymmetric similarities between items. The pseudo-code to find out the item-item similarity matrix is:

Algorithm: Item-item_prediction_matrix(R):

For each item i in R
    Compute $frequency(i)$

For each item i_1 in R
For each item i_2 in R
Compute $frequency(i_1 i_2)$
$Similarity(i_1, i_2) = \frac{frequency(i_1 i_2)}{frequency(i_1) \times (frequency(i_2))^{\alpha}}$
For each user u in R
For each item i in R
$Prediction(u, i) = \frac{\sum_{all\ similar\ users, N} Similarity(i, N) * R_{uN}}{\sum_{all\ similar\ users, N} |Similarity(i, N)|}$

4.2.5 Combiner

The prediction matrices received from both the recommenders are combined to give more efficient results. Here, the combiner is using the weighted hybrid technique. This technique in the form of the two recommenders can be viewed as:

$$Hybrid\_Prediction = \frac{\alpha.SVD + \beta.IBCF}{\alpha + \beta}$$

The above equation shows a weighted scheme for the proposed recommender. Experimental evaluations helps determine the values of $\alpha\ and\ \beta$.

4.2.6 Making Recommendations

Recommendations here refer to a list of those items that can be referred to a user assuming that it might be of his interest. The popularity of any recommendation system depends on the fact whether a user liked the items that were referred to him. Thus referring the best possible items is a task that should be carried out most crucially. The proposed algorithm in this report makes use of TruncatedSVD and conditional probability model to make the recommendations. Truncated SVD is used to make a prediction rating matrix PR_{SVD}. The item-item similarity matrix *IIS* is computed with the help of conditional probability model. Further, a prediction matrix,PR_{IBCF} is generated by the conditional IBCF.

4.2.7 Predicting For New User

Any user entering into the database does not have any ratings associated with him. In that case, UBCF would pose a greater problem. However, IBCF does not pose any problem with any user coming. Still, it does have a lot of accuracy issues. SVD here helps compute the user ratings using the singular values in the matrix S.

4.2.8 Evaluating Error Rate

Prediction accuracy is one of the most important measure to be evaluated for a recommender system. It is an independent entity and can thus be measured offline. In respect to a recommender system, error rate is found out by measuring the absolute error between the predicted and the actual values. Those values are taken into account that were formerly given by the user. Following metrics are used to evaluate the error rate :

$$RMSE = \sqrt{\frac{\sum_{(u,i)}(r_{ui} - \hat{r}_{ui})^2}{n}}$$

$$MAE = \frac{\sum_{(u,i)}|r_{ui} - \hat{r}_{ui}|}{n}$$

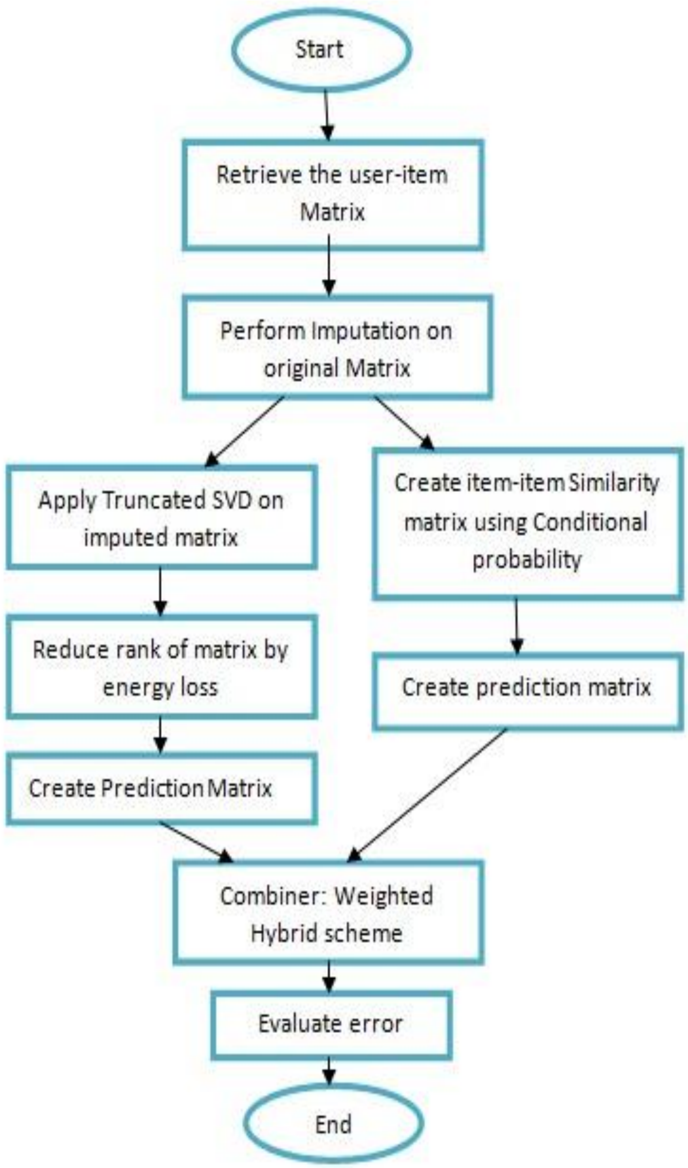
$$RE = \sqrt{\frac{\sum_{(u,i)}(r_{ui} - \hat{r}_{ui})^2}{{r_{ui}}^2}}$$

4.2.9 Comparison with Previous Strategies

For any new model proposed, one of the main purposes that should be served is its comparison with already existing algorithms. The novel hybrid proposed is compared with the individual strategies.

4.3 Flow-Chart of proposed Recommender Model

This section provides the steps necessary to apply the hybrid combination of SVD and IBCF recommender. The data set for the recommender model consists of a user-item matrix with rows defining the users and columns defining the items.



The above flow-chart represents the step-wise procedure carried out to create the hybrid recommender model.

4.4 Complete Algorithm

Given below is the stepwise procedure of creating the proposed hybrid recommender model.

Input: Original Rating data which specifies which customer has given how much rating score to a given set of items among all the items.

Output: A prediction matrix PR which specifies the predicted ratings of the unrated items from the original rating data.

Steps:

- Convert the original rating data into a user item matrix R whose number of rows is equal to the total number of customers and number of columns is equal to the total number of items.
- To fill in all the missing values in R, we do the following:
  - Convert the values represented by NAN to zero.
  - Apply a data preprocessing algorithm for the zero values in order to avoid the problem of data sparsity to get a new filled matrix as X.
- Find out the rank of the matrix X as r.
- Apply Truncated SVD on this matrix X to get the values of U,S and V by taking the number of components as X.
- Calculate the energy of all the singular values in matrix S.
- Reduce the dimensions of the obtained matrices U, S and V by the k component. K is achieved by eliminating a certain amount of singular values.
- Update the matrices U and V by reducing their dimensions by m-r and n-r respectively. Thus the new updated matrices obtained are of dimensions: $U_{m\times k}\,; S_{k\times k}\ and\ V^T{}_{k\times n}$.
- Compute the square root of the matrix S as $\sqrt{S}$ by simply calculating the square root of every value in S. Calculate two new matrices A and B as $A = U_k \times \sqrt{S_k}$ and $B = \sqrt{S_k} \times V_k^T$.
- Create the matrix PR, predict all the unrated items X_{ui} in X by using the formula : $PR_{ui} = \overline{X_u} + A_m \times B_n$ ($\overline{X_u}$ represents the average rating for user u, A_m represents the m^{th} row in the matrix A and B_n represents the n^{th} row in the matrix B).
- Calculate the item item similarity matrix I by using the conditional probability function.
- Create a prediction matrix for item based CF.
- Apply a weighted hybrid scheme on the two prediction matrices received.
- Evaluate error rate using relative error, RMSE and MAE.

In the following chapter the results obtained after applying the proposed algorithm, simulationenvironment require for algorithm execution are discussed.

Time Complexity: O(mn) where m is the number of users and n is the number of items.

Space Complexity: O(mn) where m is the number of users and n is the number of items.

Chapter 5

EXPERIMENTAL ANALYSIS

This chapter provides the implementation details of the proposed recommender model, Truncated SVD with conditional Collaborative Filtering System. The detailed explanation pertaining to implementation can be divided into three sections. The first section provides a brief description of recommender system's implementation platform, the second section discusses the implementation details of training phase and the third section presents the results.

5.1 Dataset

The dataset used is the popular MovieLens data. MovieLens data sets were collected by the GroupLens Research Projectat the University of Minnesota.This data set consists of:

- 100,000 ratings (1-5) from 943 users on 1682 movies.
- Each user has rated at least 20 movies.

The data was collected through the MovieLens web siteduring the seven-month period from September 19th, 1997 through April 22nd, 1998. This data has been cleaned up – userswho had less than 20 ratings or did not have complete demographicinformation were removed from this data set.

The data set was converted into a user-item matrix A that had 943 rows (i.e., 943 users) and 1682 columns (i.e., 1682 movies that were rated by at least one of the users). For the experiments, we also take another factor into consideration, sparsitylevel of data sets The sparsity level of the Movie dataset is, therefore, $1-(100{,}000/(943\times1682))$, which is 0.9369%.

5.2 Simulation Environment

The section provides information about the tools used in deriving the results in this research.Various parameters are experimentally evaluated using the libraries of python and with the use of csv files.

5.2.1 Brief Discussion about Python

Python is a widely used high-leveldynamic programming language. The design philosophy of python emphasizes on code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. We can list down the following characteristics of Python:

- general-purpose interpreted
- interactive
- object-oriented, and high-level programming language

Python has the advantage of a lot of in-built libraries that need to be installed on the system. The proposed model makes use of a lot of python libraries. These are:

Table 1Important Python Softwares used

Python Software	Features
Scikit-learn	It is a free machine learning software. It has incorporated various classification, regression and clustering algorithms.
Numpy	It is a fundamental package for scientific computing with Python. It contains sophisticated functions, powerful N-dimensional array object, Fourier transform, tools for incorporating C/C++ and Fortran code.
Scipy	SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering.
Pandas	Pandas is used for data manipulation and analysis. It offers data structures and operations that help manipulate numerical tables.

The above table describes the features of all the python softwares that were installed on the idle using pip command.

5.3 Results

The section shows the results obtained. The research was carried out in a system with 4 GB

RAM, 500 GB HDD and 1.6 GHz core i3 Intel Processor.

Effect of energy Preservation: Energy of singular values to be preserved is an important concept in the truncated SVD model. The more the energy is preserved, more is the number of singular values taken and thus the dimensionality reduced is inverse of it. Therefore we take the value of energy to be preserved in such a way that we maintain a balance between the accuracy model and the dimensions reduced.

Table 2 Energy Preserved v/s k value

Energy_preserved	K
50 % of energy	5
60% of energy	10
70% of energy	21
80% of energy	42
90% of energy	84

The above table shows a direct relationship between the energy of the singular values to the amount of singular values, referred to here as *k* preserved. We can see a direct relationship between them.

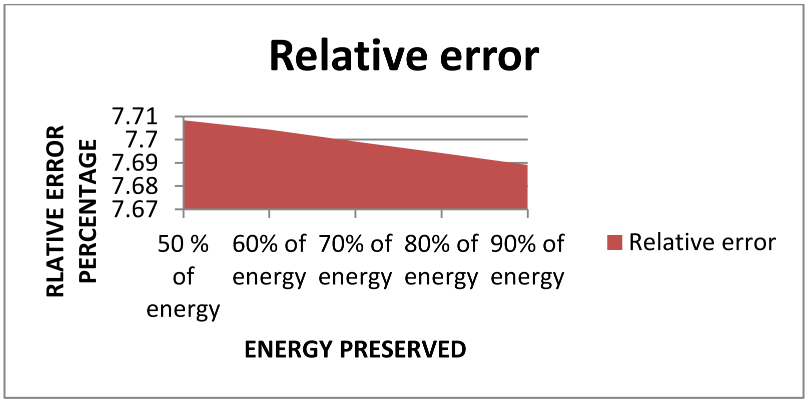


Figure 10 Relative error for different values of energy preservation

The above figure shows a direct relationship between the energy preservation of singular values and the amount of error found. We can represent this relationship as:

$$Energy\_preserved \propto \frac{1}{error\_found}$$

Effect of Imputation: Imputation has been a very important part of this research. Below analysis shows the difference between the accuracy results with and without imputation.

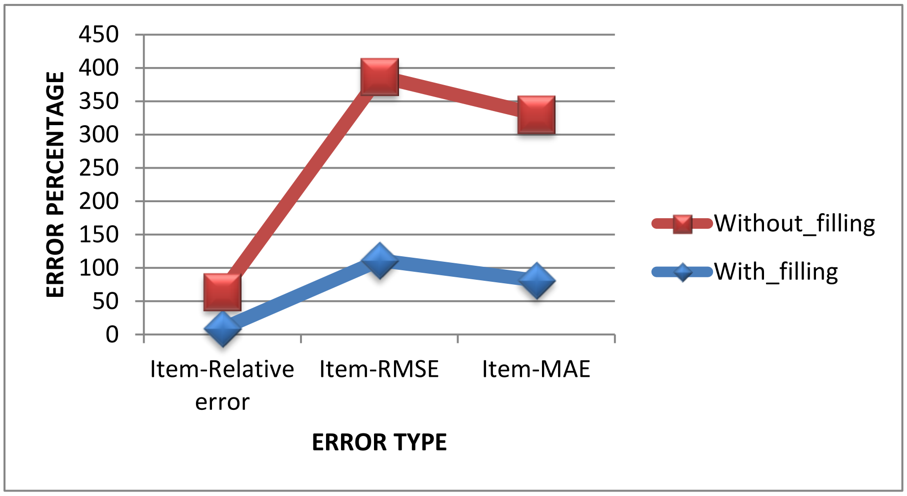


Figure 11Comparison of IBCF with and without imputation

The above line chart in Figure 10 shows that a IBCF recommender system performs far better when it is fed with some input values to reduce the sparseness. Without the imputation all the three error rates are quite high. For example, the RMSE error goes upto 2.7606 from being just 1.11 with imputation.

Comparison between traditional cosine based IBCF and conditional IBCF:

Traditional collaborative filtering methods used cosine similarity, pearson co-relation and and Euclidean distance as its popular metrics to calculate the similarity. Following figure shows the advances of conditional IBCF with respect to cosine based IBCF.

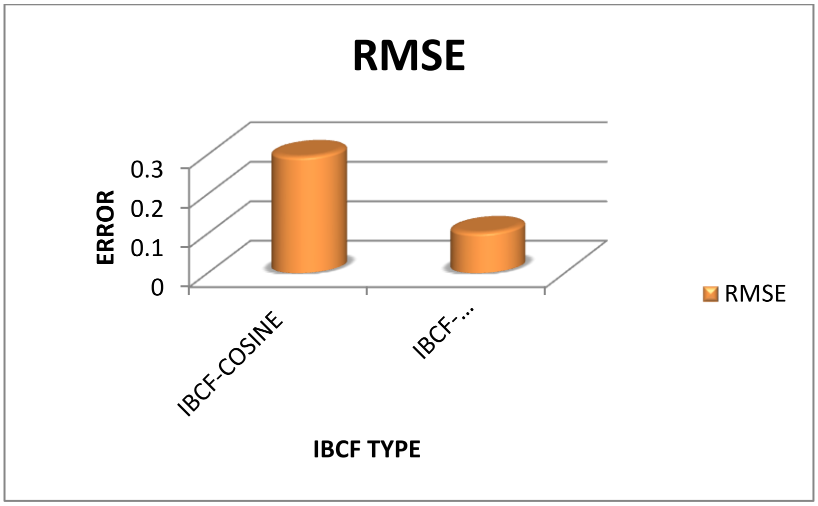


Figure 12RMSE error for cosine based IBCF v/s Conditional probability based IBCF.

Figure 12 shows that the proposed method of conditional probability is very efficient compared to the traditional methods of similarity computations.

Comparison Between IBCF, SVD and Proposed Recommenders

Three error types are chosen to compare the results of three recommenders. IBCF and SVD recommenders are made with one of the best composition, still the hybrid system out-performs both of them on an average. It shows very low error rate with all kinds of input data.

Input data is modified according to the imputations. Nine kinds of imputations have been taken here. All have been generated through Statistical methods. These imputations are:

- mean_comun_row: values filled with the average of mean values of rows and columns.
- median_comun_row: values filled with the average of median values of rows and columns.
- mode_comun_row: values filled with the average of mode values of rows and columns.
- median_row: values filled with the median values of rows.
- mode_row: values filled with the mode values of rows.
- mean_row: values filled with the mean values of rows.
- median_column: values filled with the median values of columns.

- mean_column: values filled with the mean values of columns.
- mode_column: values filled with the mode values of columns.

Following charts shows the error comparisons on the basis of these nine input types among the three recommenders.

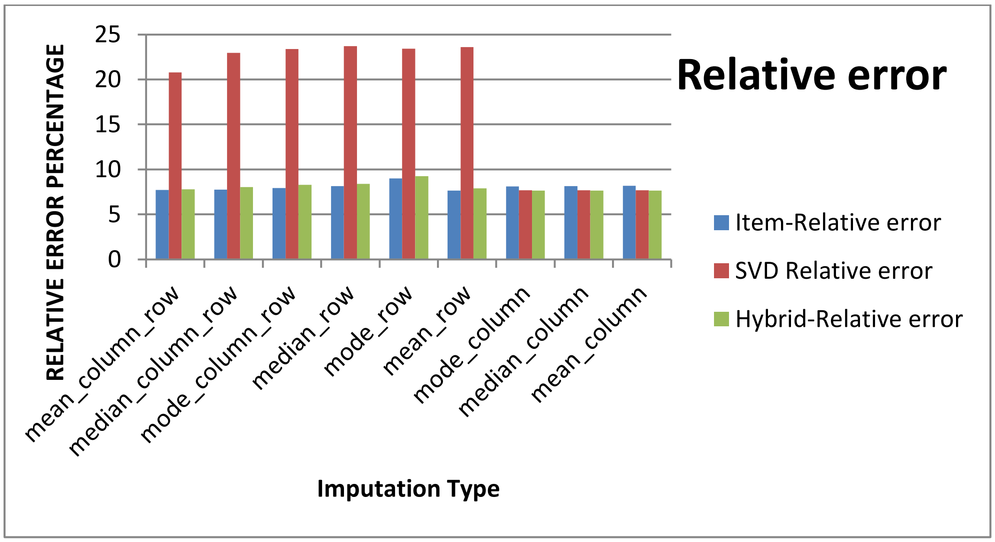


Figure 13Relative error for SVD, IBCF and hybrid models

The above figure shows the comparison between the relative error between the three techniques.

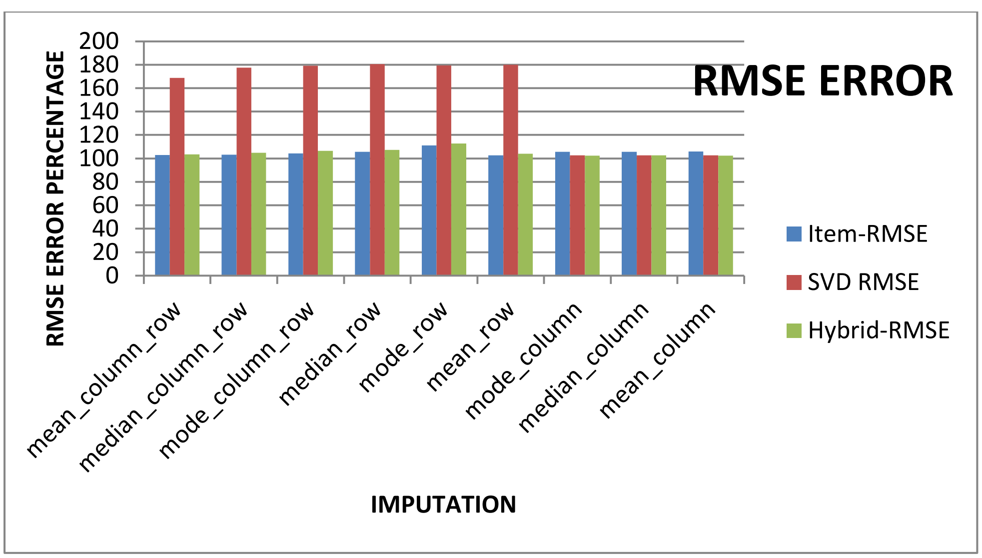


Figure 14 RMSE error for SVD, IBCF and hybrid models

The above figure shows a comparison between the RMSE error for the three techniques.

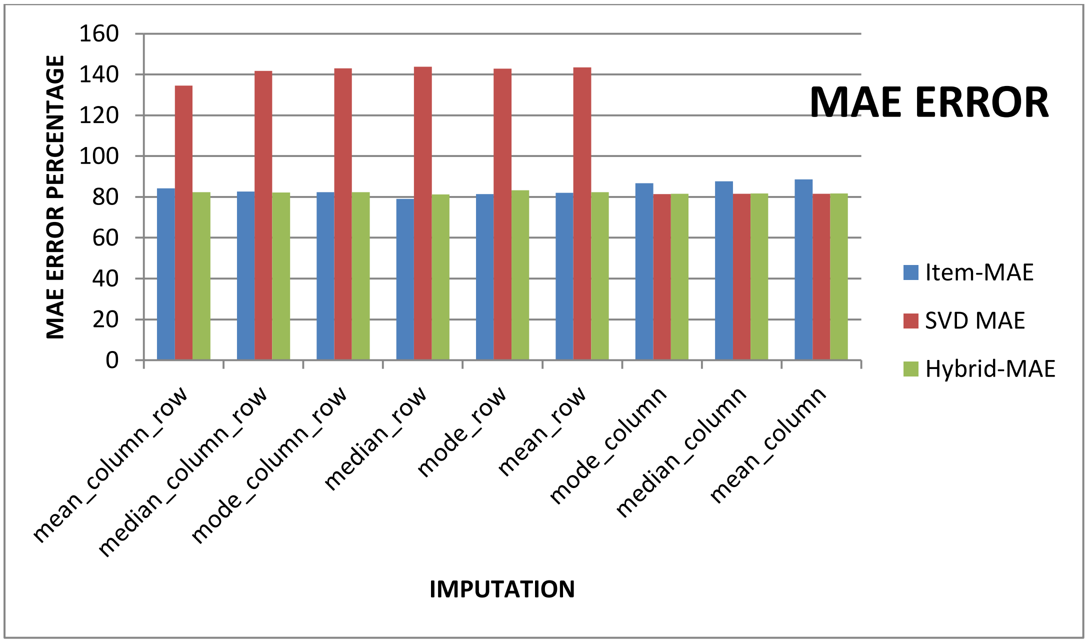


Figure 15 MAE error for SVD, IBCF and Hybrid models

The above figure shows a comparison between the MAE error for the three techniques.

Below table shows the average accuracy results for the SVD, IBCF and Hybrid model of Truncated SVD and conditional IBCF Recommenders.

Table 3 Average Error Comparison

Error/Technique	IBCF	SVD	Hybrid
Average RE	8.08216	17.87848	7.9981
Average RMSE	105.3017	152.608	105.1757
Average MAE	83.83951	121.5496	82.05208
Total	197.22337	292.03608	195.22588

The above table shows the average errors with respect to all the three recommenders. Lowest of all the errors have been highlighted. Thus the results prove the efficiency of the proposed recommender to be more than that of its components.

Time Complexity Comparison

Time complexity of the proposed novel model is very low compared to traditional collaborative filtering models. The proposed system makes use of a binary matrix that is computed using the user-item matrix. This binary matrix is used to compute a co-occurrence matrix which helps generate the item-item similarity. Given below is the analysis of time taken by the proposed system and traditional collaborative filtering method.

Table 4 Comparison of Time Complexity

Technique	**Time Taken**
Traditional IBCF	1 hour 58 minutes
Truncated SVD + Conditional IBCF	55 seconds

Above analysis shows that the proposed novel method, Truncated SVD with conditional Collaborative Filtering Recommender model is better than traditional IBCF techniques and SVD techniques as well in terms of both time complexity and accuracy analysis.

Chapter 6

CONCLUSION AND FUTURE WORK

This chapter discusses the conclusions inferred from this research and presents the possibilities of extension of this work in future.

6.1 Conclusion

The research work proposes a new hybrid model that implements a matrix factorized technique along with a traditional collaborative filtering technique.

The novel method proposed performs better than its two components on an average. The SVD method helps reduce the dimensionality of the matrix which helps deal with huge datasets. Truncated SVD helps to conduct the algorithm in a faster manner by reducing the rank of the input matrix even further. Conditional probability methods used for IBCF has proven to be very efficient when imputated with statistically computed values.

Both the individual methods, SVD and IBCF are very efficient methods in terms of accuracy. But they may lack sometimes with different types of imputations. However, the hybrid system excels with all the kinds of inputs and gives very low error rate as low as only 1.02 RMSE and .812 MAE.

Issues like cold-start are also dealt with the conditional similarity measure as not only the very popular items are recommended, rather new items are also recommended.

Time complexity of the proposed model is very low compared to the traditional collaborative filtering models. The conditional probability based CF is very fast and efficient. In collaborance with SVD model, it gives best results with all kinds of inputs.

6.2 Future Scope of Work

The proposed method could be made more efficient in terms of running time complexity by incorporating incremental SVD algorithm. Also it can be made dynamic in the sense that we can include the new data being generated in the incremental svd. The incremental method is famous among the LSI researchers to handle the situation of dynamic databases where the new documents and terms arrive once the model is built.

Thus, future work involves the simulation of incremental SVD along with the proposed hybrid model. This would not only allow the creation of a dynamic recommender system, but would also help increase the system performance as the model would be built in small incremental steps rather than taking the whole database at one.

REFERENCES

[1] Z. Sharifi, M. Rezghi and M. Nasiri, "New algorithm for recommender systems based on singular value decomposition method," Computer and Knowledge Engineering (ICCKE), 2013 3th International eConference on, Mashhad, 2013, pp. 86-91.
doi: 10.1109/ICCKE.2013.6682799

[2] S. Gong, H. Ye and Y. Dai, "Combining Singular Value Decomposition and Item-based Recommender in Collaborative Filtering," Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop on, Moscow, 2009, pp. 769-772.
doi: 10.1109/WKDD.2009.132.

[3] Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," in Computer, vol. 42, no. 8, pp. 30-37, Aug. 2009.
doi: 10.1109/MC.2009.263

[4] Qilong Ba, Xiaoyong Li and ZhongyingBai, "Clustering collaborative filtering recommendation system based on SVD algorithm," Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on, Beijing, 2013, pp. 963-967.
doi: 10.1109/ICSESS.2013.6615466

[5] H. Ji, X. Chen, M. He, J. Li and C. Ren, "Improved recommendation system via propagated neighborhoods based collaborative filtering," Service Operations and Logistics, and Informatics (SOLI), 2014 IEEE International Conference on, Qingdao, 2014, pp. 119-122.
doi: 10.1109/SOLI.2014.6960704

[6] X. Shen, H. Long and C. Ma, "Incorporating trust relationships in collaborative filtering recommender system," Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015 16th IEEE/ACIS International Conference on, Takamatsu, 2015, pp. 1-8.
doi: 10.1109/SNPD.2015.7176248

[7] Y. Ren and S. Gong, "A Collaborative Filtering Recommendation Algorithm Based on SVD Smoothing," Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on, Nanchang, 2009, pp. 530-532.
doi: 10.1109/IITA.2009.491

[8] D. I. Ignatov, S. Nikolenko, T. Abaev and N. Konstantinova, "Online Recommender System for Radio Station Hosting: Experimental Results Revisited," Web Intelligence (WI) and

Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on, Warsaw, 2014, pp. 229-236.
doi: 10.1109/WI-IAT.2014.38

[9] A. S. Tewari, A. Kumar and A. G. Barman, "Book recommendation system based on combine features of content based filtering, collaborative filtering and association rule mining," Advance Computing Conference (IACC), 2014 IEEE International, Gurgaon, 2014, pp. 500-503.
doi: 10.1109/IAdCC.2014.6779375

[10] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]// Proceedings of the 10th international conference on World Wide Web. New York:ACM press, 2001:pp. 285-293

[11] Karypis G. Evaluation of Item-based Top-N Recommendation Algorithms[R]. Minneapolis: Dept. of Computer Science, University of Minnesota, Technical Report: #00-046, 2000.

[12] Y. Jiang, J. Liu, M. Tang and X. Liu, "An Effective Web Service Recommendation Method Based on Personalized Collaborative Filtering," Web Services (ICWS), 2011 IEEE International Conference on, Washington, DC, 2011, pp. 211-218.
doi: 10.1109/ICWS.2011.38

[13] M. Sunitha Reddy and T. Adilakshmi, "Music recommendation system based on matrix factorization technique -SVD," Computer Communication and Informatics (ICCCI), 2014 International Conference on, Coimbatore, 2014, pp. 1-6.
doi: 10.1109/ICCCI.2014.6921744

[14] M. Zheng, F. Min, H. R. Zhang and W. B. Chen, "Fast Recommendations With the M-Distance," in IEEE Access, vol. 4, no. , pp. 1464-1468, 2016.
doi: 10.1109/ACCESS.2016.2549182

[15] Toby Segaran, " Programming Collective Intelligence", Chapter.1-3, pp. 1-53.

[16] Yanhai Zhao, Jianyang Li and XiuzhengXie, "An intelligent recommender derived from its characteristic case revision," 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, 2010, pp. V5-240-V5-244.
doi: 10.1109/ICCASM.2010.5619192

[17] B. Wang, Q. Liao and C. Zhang, "Weight Based KNN Recommender System," Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2013 5th International Conference on, Hangzhou, 2013, pp. 449-452.
doi: 10.1109/IHMSC.2013.254

[18] P. Nagarnaik and A. Thomas, "Survey on recommendation system methods," Electronics and Communication Systems (ICECS), 2015 2nd International Conference on, Coimbatore, India, 2015, pp. 1496-1501.
doi: 10.1109/ECS.2015.7124835

[19] Jure Leskovec ,AnandRajaraman,Jeffrey D. Ullman," Mining of Massive Datasets", Chapter -11,2014,pp. 405-437.

[20] L. Xiong, Y. Xiang, Q. Zhang and L. Lin, "A Novel Nearest Neighborhood Algorithm for Recommender Systems," 2012 Third Global Congress on Intelligent Systems, Wuhan, 2012, pp. 156-159.
doi: 10.1109/GCIS.2012.58

[21] S. Kumar and D. Raj, "A contemporary approach to hybrid expert systems case base reasoning," Computer and Communication Technology (ICCCT), 2010 International Conference on, Allahabad, Uttar Pradesh, 2010, pp. 736-740.
doi: 10.1109/ICCCT.2010.5640376

[22] A. Bilge and C. Kaleli, "A multi-criteria item-based collaborative filtering framework," Computer Science and Software Engineering (JCSSE), 2014 11th International Joint Conference on, Chon Buri, 2014, pp. 18-22.
doi: 10.1109/JCSSE.2014.6841835

[23] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, "Application of dimensionality reduction in recommender system–A case study," in Proceedings of the ACM WebKDD 2000 Web Mining for E-commerce Workshop, Boston, MA, USA, 2000

[24] M. Plantie, J. Montmain, and G. Dray, "Movies recommenders systems: Automation of the information and evaluation phases in a multicriteria decision-making process," in Database and Expert Systems Applications, ser. Lecture Notes in Computer Science, K. Andersen, J. Debenham, and R. Wagner, Eds. Springer Berlin Heidelberg, 2005, vol. 3588, pp. 633–644.

[25] Y. Blanco-Fernandez, J. J. Pazos-Arias, A. Gil-Solla, M. Ramos-Cabrer and M. Lopez-Nores, "Providing Entertainment by Content-based Filtering and Reasoning in Intelligent

Recommender Systems," 2008 Digest of Technical Papers - International Conference on Consumer Electronics, Las Vegas, NV, 2008, pp. 1-2.
doi: 10.1109/ICCE.2008.4587849

[26] Golub G H, Van Loan C F. Matrix Computations (3rd edition) [M] . Johns Hopkins University Press, 1996 .

[27] Y. Wu, Q. Yan, D. Bickson, Y. Low and Q. Yang, “Efficient Multicore Collaborative Filtering”, arXiv:1108.2580v2 [cs.LG] 17 Aug 2011.

[28] X. Shi, H. Ye and S. Gong, "A Personalized Recommender Integrating Item-Based and User-Based Collaborative Filtering," Business and Information Management, 2008. ISBIM '08. International Seminar on, Wuhan, 2008, pp. 264-267.
doi: 10.1109/ISBIM.2008.191

[29] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. (1997).GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM, 40(3), pp. 77-87.

LIST OF TABLES

LIST OF FIGURES

Table of contents