An Approach to a University Recommendation by Multi-criteria Collaborative Filtering and Dimensionality Reduction Techniques

Conference Paper · December 2015

DOI: 10.1109/iNIS.2015.36

CITATIONS
17

READS
1,703

3 authors:

Dheeraj Bokde
Blazeclan Technologies Pvt. Ltd.
4 PUBLICATIONS **297** CITATIONS
SEE PROFILE

Sheetal Girase
Maharashtra Institute of Technology
16 PUBLICATIONS **439** CITATIONS
SEE PROFILE

Debajyoti Mukhopadhyay
Bennett University
251 PUBLICATIONS **1,537** CITATIONS
SEE PROFILE

An Approach to A University Recommendation by Multi-Criteria Collborative Filtering and Dimensionality Reduction Techniques

Dheeraj kumar Bokde
Department of Information Technology
Maharashtra Institute of Technology
Pune, India
bokde.dheeraj@gmail.com

Sheetal Girase
Department of Information Technology
Maharashtra Institute of Technology
Pune, India
girase.sheetal@gmail.com

Debajyoti Mukhopadhyay
Department of Information Technology
Maharashtra Institute of Technology
Pune, India
debajyoti.mukhopadhyay@gmail.com

***Abstract* - Collaborative Filtering (CF) algorithms are most commonly used prediction technique in field of Recommender Systems (RS) for Information Filtering. It makes use of single criteria ratings that user have assigned to items which plays an important role in e-commerce to assist users in choosing items of their interest. For complex and massive dataset, Multi-Criteria Collaborative Filtering (MC-CF) frequently give better performance, accurate and high quality recommendations for users considering multiple aspects of items. CF algorithms need to be continuously updated because of a constant increase in the quantity of information, ways of access to that information, scalability and sparseness in rating matrix. Dimensionality Reduction techniques like: Matrix Factorization and Tensor Factorization techniques have proved to be a quite promising solution to the problem of designing efficient CF algorithm in the Big Data Era.**

This work aims at offering University Recommendation System, which combines Multi-Criteria Collaborative Filtering and Dimensionality Reduction technique to provide high quality University/College recommendation to Students. The proposed solution not only reduces the computation cost but also increases the prediction accuracy and efficiency of the MC-CF algorithms implemented using the Apache Mahout framework.

Keywords – University Recommendation System; Recommender System; Multi-Criteria Collaborative Filtering; Dimensionality Reduction; Apache Mahout

I. INTRODUCTION

On the Internet today, an overabundance of information can be accessed, it becomes difficult for the users to process and evaluate options and make appropriate choices, is termed as information overload. *Recommender Systems (RS)* are most commonly used techniques for Information Filtering, which play an important role in e-commerce [1-5]. Therefore, RS is the solution to the problem of information overload.

Recommender System are the software tools to help users' in decision making process. RS makes personalized suggestions for the users' to select items based on their preferences by applying Data Mining techniques and prediction algorithms. They bridge the gap between searching and sharing technologies. According to Burke [4]: "RS have the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options". Examples for such RS's includes Book recommendations by Amazon, Movie recommendations by Netflix and Yahoo!Movies etc.

Recommendation System can be classified into: *Content-Based (CB), Collaborative Filtering (CF)* and Hybrid RS [1][3]. The CB method classify the user-item metadata and gives recommendation according to classification results. CF predicts the overall rating for an item based on past ratings regarding both item overall and individual criteria, finally recommend an items to the user with best overall score [3-5]. Nowadays e-commerce industry is growing at exponential rate, and the users' are using these systems to a greater extent, their changing taste for selection of items based on multiple criteria's making business more complex. In such situation, customers face difficulty to find optimal information about products from available choices. For large and complex dataset MC-CF method frequently give better performance, accurate and high quality recommendations for users considering multiple aspects of items, which serves a win-win strategy to both user and e-commerce industry [6][11]. Earlier CF-algorithms have a very high time complexity and a very poor scalability, which utilize the association inferences. Recent methods make use of matrix operation which are more scalable and efficient.

The Netflix Prize Contest [12], an open competition for the best algorithm to predict user ratings for movies, based on former ratings. The contest proved the superiority of mathematical methods that discover latent factors which drives user-item similarity, with respect to classical CF algorithms. With the ever-increasing information available in digital media, the implementation of personalized filters becomes the challenge for designing an efficient algorithms. In recent years, latent factors models i.e., Dimensionality Reduction techniques like: *Matrix Factorization(MF)* and *Tensor Factorization (TF)* techniques have proved to be a quite promising solution to the problem of designing efficient CF algorithms in the *Big Data Era* [5-6].

This paper proposes a *University Recommendation System (URS)*, which provides the University or Engineering College recommendations to students, where should they apply for

admission. URS makes use of Multi-Criteria Item-Based Collaborative Filtering and Dimensionality Reduction techniques to generate high quality recommendations. The proposed solution not only reduces the computational cost but also increases the prediction accuracy and efficiency of the MC-CF algorithms implemented using the Apache Mahout framework.

A. University Recommendation Problem

Education plays very important in the development of any nation, the countries with an effective system of education lead the world, both socially and economically. Education system is the only key for the development of any nation. It's the basic need and birth right of every individual to get better education to survive in competitive world. To get the better education admission in best university is needed, therefore this topic is of great importance. Nowadays everyone is bothered about choosing a right University/College for admission of his/her choice and interest. Admission into University/College is a complex decision making process that goes beyond simply matching test scores and admission requirements. Online magazine research and surveys has suggested that students' backgrounds and other factors correlate to the performance of their tertiary education. How the student chooses the university to take admission and alternately university chooses a student to admit, determines the success of both sides in education system.

World over, Universities/Colleges have been ranked based on their research output and perception of University Ranking system. So far, India has not seen a ranking system for the Universities matching the style and standard of Times Higher Education (THE) or QS Ranking. There is need to provide a platform for the aspirants to get information about best Indian Universities/Colleges who wants to seek admission. So, we came up with the solution as University Recommendation System.

University Recommendation System intended to guide the student seeking admission to an Engineering College based on ratings given by the students in past. The interests of individual changes according to their lifestyle. Aspirants might be more interested to know information of the University/College, Establishment, University/College type, Affiliation, Courses offered and seats available, Infrastructure, Faculty, Placement potential, Campus Life, Ranking and Global exposure etc. So choosing the right institution makes one's life better at the later stages of life.

B. Organization

The rest of paper is organized as follows. In Section II background of Multi-Criteria Collaborative Filtering and Dimensionality Reduction Techniques along with Apache Mahout Framework is presented. URS system architecture and research methodology are discussed in subsections of Section III. In Section IV, the experiments to evaluate the accuracy and performance of our implementation and results are discussed. Finally, conclusion and future scope is presented in Section V.

II. BACKGROUND

A. Multi-Criteria Collaborative Filtering

Before introducing MC-CF, a brief information of traditional CF algorithm is needed. In this section, brief introduction to traditional CF and MC-CF is given.

1) Collaborative Filtering

The term Collaborative Filtering was first coined by David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry in 1992 to describe an email filtering system called "Tapestry". Tapestry [2] was an experimental mail system developed at the Xerox Palo Alto Research Center. One way to handle large volumes of mail is to provide mailing lists, allowing the users to subscribe only to those are of their interest by either rate mails ("good" or "bad"). This simply means that people collaborate to help one another performs filtering by recording their reactions to mailing list they subscribed. The CF, recommends an item to a user based on opinions of other like-minded users that are most liked by them. It works on the principle that the user' have same likings in the past will have similar choices in the future as well.

CF algorithms are further divided into User-Based and Item-Based approaches. In user-based approach similarity between users or group of users is determined, then algorithm recommends the items to the user suggested by the other users of the same community [3-5]. However this method face challenges like: large dataset size, sparsity of U-I rating matrix and scalability [5-6]. While in Item-Based approach similarity between items is determined. Here we explain the Item-Based CF algorithm proposed by Sarwar et al. [10]. In a single rating CF system the collected user rating is utilized to predict a rating given by a function:

$$R: Users \times Items \rightarrow R_0$$

Where, R_0 is the set of possible overall ratings

The task of prediction was accomplished in the Item-Based CF by forming each item's neighborhood, Sarwar et al. proposed the adjusted cosine-based similarity as a measure for estimating items distance. After similarities are computed then first form the nearest neighborhood of an item *i*, then calculate prediction for an active user for a target item. The predicted value is within the same scale that is used by all users for rating and then recommend a list of Top-N items that the active user will like the most.

2) Multi-Criteria Collaborative Filtering

Adomavicius and Kwon introduced to incorporate multi-criteria rating concept in Recommender System [6][11]. They consider MC-CF predict the overall rating for an item based on past ratings regarding both the item individual criteria and overall rating, and then recommend an item to the user with the best overall score. Thus, the algorithm for a MC-CF can be prolonged from a single-rating CF approach. In MC-CF problem, there are *m* users, *n* items and *k*-criteria in addition to the overall rating. Users have provided a number of explicit ratings for items; a general rating R_0 must be predicted additional *k*-criteria $(R_1, R_2, \ldots\ldots, R_k)$ ratings. Then in MC-CF there is more than one criteria's, therefore traditional prediction modified as follows:

$$R: Users \times Items \rightarrow R_0 \times R_1 \times R_2 \times \ldots\ldots \times R_k$$

Where,

R_0 is the set of possible overall ratings
R_i indicates the possible rating for each criteria *i*
k shows the number of criteria

The MC-CF process can be defined in two steps by predicting the target user's rating for a particular unseen item, followed recommendation. Recommendation is a list of Top-N items that the active user will like the most, usually list consists of the products not already purchased by active user.

B. Dimensionality Reduction Techniques

To address the challenges of MC-CF algorithms like scalability and sparsity, many researchers have used the Dimensionality Reduction Techniques with CF algorithms [5-6]. In general, the process of dimensionality reduction can be described as mapping a high dimensional input space into lower dimensional latent space. There are two techniques namely Matrix Factorization and Tensor Factorization.

1) Matrix Factorization

A special case of dimensionality reduction is MF where a data matrix A is reduced to the product of several low rank matrices. MF techniques fall in the class of CF methods and particularly in the class of latent factor models [12]. Assuming that similarity between users and items is induced by some factors hidden in the data. In case of latent factor models a matrix of users and items is build where each element is associated with a vector of characteristics. The resulting dot product $q_i^T p_u$ captures the interaction between user u and item i, the users' overall interest in the item characteristics. This approximates user u's rating of item i which is denoted by r_{ui} leading to the estimate is given by [12]:

$$\hat{r}_{ui} = q_i^T p_u \qquad (1)$$

A high correspondence between user and item factors leads to a recommendation. RS data are collected in a matrix called user-item rating matrix: rows are referred to users and columns to items, the intersection between one row and one column is the rating given by the user. Missing values correspond to items not yet rated by the user.

MF is specially used for processing large RS databases and providing scalable approaches to reduce the problem from high levels of sparsity. The model-based technique *Latent Semantic Index (LSI)* and the reduction methods *Singular Value Decomposition (SVD), Principal Component Analysis (PCA)* are well known technique for identifying factors.

a) Singular Value Decomposition

A powerful technique for dimensionality reduction is SVD and it is a particular realization of the MF approach. Applying SVD in the CF domain requires factoring the user-item matrix [12-14][16]. To find a lower dimensional feature space is the key issue in a SVD. This raises difficulties due to the high portion of missing values caused by sparseness in the user-item ratings matrix. SVD of matrix A of size $m \times n$ is of the form is given by Eq. (2) [13].

$$SVD\ (A) = U\Sigma V^T \qquad (2)$$

Where, U and V are $m \times m$ and $n \times n$ orthogonal matrices respectively, Σ is the $m \times n$ singular orthogonal matrix with non-negative elements and $\Sigma(\sigma_1, \sigma_2, \ldots, \sigma_n)$ are called the singular values usually placed in the descending order in Σ of matrix A. A_k is the linear approximation of matrix A with reduced rank k is given by Eq. (3) [13].

$$SVD\ (A_k) = U_k\, \Sigma_k\, V_k^T \qquad (3)$$

Where, U_k and V_k are produced by removing m-k columns, n-k rows from matrix U and V respectively. Matrix Σ_k is the $k \times k$ principle diagonal sub-matrix of Σ.

b) Principle Component Analysis

Similar to the dimensionality reduction technique SVD, PCA is also a particular realization of the MF approach leads to faster computation of recommendation [15-16]. PCA is a multivariate mathematical procedure which uses an orthogonal transformation that converts a set of observations of possibly correlated variables into a set of values which are linearly uncorrelated variables called *Principal Component* (PC). The number of PC is less than or equal to the number of original variables. This transformation can be defined as the first PC has the largest possible variance and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The PC are orthogonal because they are the eigenvectors of the covariance matrix. PCA is sensitive to the relative scaling of the original variables.

2) Tensor Factorization

The main limitation of MF techniques is that they only consider the standard profile of users and items, which doesn't allow to integrate context. Contextual information (the place where the user see the movie, the device, the company, etc.) cannot be managed with simple user-item matrices. Tensor can be seen as higher-dimensional arrays of numbers [6-9], might be exploited in order to include additional contextual information in recommendation. In standard multivariate data analysis, data are arranged in two-dimensional structure. However for more than two dimensional domains, more appropriate structures are required for taking into account. The techniques that generalize the MF can also be applied to tensor.

a) Higher Order Singular Value Decomposition

In RS literature, the most frequently used technique for TF is *Higher order Singular Value Decomposition (HOSVD)* or *Multi-linear Singular Value Decomposition (MLSVD)* was proposed by Lathauwer at el. (2000) [7], is a generalization of SVD that can be applied on three (or more) dimension called Tensor. The objective is to compute multi-rank approximation of the data. These approximations are expressed in terms of tensor decomposition. HOSVD is used where the factorization of a tensor is applied to manage data for users, items and user ratings. The HOSVD of a third-order tensor involves the computation of the SVD of three matrices called modes is shown in Fig. 1. The SVD of a real matrix A is given by: $SVD\ (A) = U\Sigma V^T$. In HOSVD, S is in general not sparse and diagonal as Σ in the SVD.

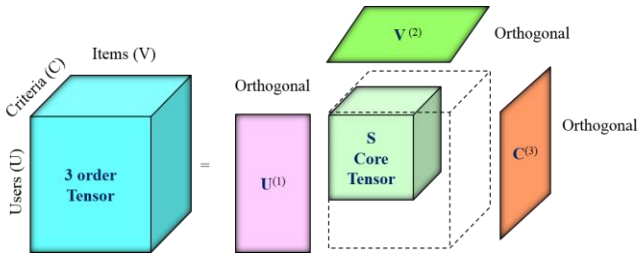

Fig. 1. Diagramatic representation of 3-order Tensor Decomposition

The procedure for obtaining HOSVD is given below [6-8]

Step 1: Unfolding the mode-d tensor $T \in R^{I_1 \times I_2 \times I_3}$ which yields matrices $A(1), \ldots, A(d)$. In the case of 3rd order $T \in R^{I_1 \times I_2 \times I_3}$, there exists three matrix unfolding as:

- **mode-1:** $j = i_2 + (i_3 - 1)I_3$
- **mode-1:** $j = i_3 + (i_1 - 1)I_1$
- **mode-1:** $j = i_1 + (i_2 - 1)I_2$

Step 2: Identifying the n left singular matrices as $U^{(1)}, \ldots, U^{(n)}$ obtained by:

$$A^{(n)} = U^{(n)}\, \Sigma^{(n)}\, V^{(n)}, \qquad n = 1, \ldots, d$$

Where,

- The matrices $U^{(n)} \in R^{I_n \times I_n}$ stands for left singular matrices
- $\Sigma^{(n)} \in R^{I_n \times (I_1 I_2 \ldots I_{n-1} I_{n+1} \ldots I_d)}$ stands for singular values in a diagonal matrix with descending order
- The matrix $V^{(n)}$ stands for right singular matrices such that $V^{(n)T}V^{(n)}=I$ and $U^{(n)T}U^{(n)}=I$, these singular matrices are orthonormal

Step 3: Finding the $S \in R^{I_1 \times I_2 \times \ldots \ldots \times I_d}$ (core tensor) through contracting the left singular matrices $U^{(n)}$ with original tensor T: $S = T \times {}_1U^{(1)T} \times {}_2U^{(2)T} \times {}_dU^{(d)T}$.

The major advantage of HOSVD is the ability of simultaneously taking into account more dimensions. This allows a better data modeling than standard SVD, since dimensionality reduction can be performed not only in one dimension but also separately for each dimension. But HOSVD is not an optimal tensor decomposition because of least squares data fitting. The computation of HOSVD needs standard SVD computation only, where truncating the first *n* singular values allows to find the best *n*-rank approximation of a given matrix. Despite this, the approximation obtained is not far from the optimal one and can be computed much faster. HOSVD cannot deal with missing values, as they are treated as "0" [6-9].

C. *Apache Mahout*

Apache Mahout is an open source Machine Learning library which provides sufficient framework utility for distributed and non-distributed programming [17-20]. It is scalable and can handle large amount of data compared to other Machine Learning framework. Apache Mahout is one of the Apache Hadoop [21] projects. Apache Mahout have support for three types of algorithms: Recommender System, Clustering and Classification. The implementations of recommender systems can be further categorized as non-distributed methods and distributed methods. One of the distributed implementation of RS uses MapReduce, which is scalable and suitable to handle massive and distributed dataset. Its scalability and focus on real world applications make Mahout an increasingly popular choice for organizations seeking to take advantage of large scale Machine Learning. Mahout also provides the support to evaluate the performance of the recommender system algorithms.

III. University Recommendation System

A. *System Architecture*

To the best of our knowledge, there are numerous Data Mining programs and technologies available, but not meant for generating recommendations in education domain i.e., for students by predicting university recommendation for them. The proposed architecture of University Recommendation System is shown in Fig. 2. The architecture of URS has two main components: Recommendation Engine and Graphical User Interface which provides an easy to use interface for the students to interact with the system.

The URS works by collecting Students feedback in the form of ratings for Universities over multiple criteria's like: Infrastructure Facility, Teachers, Placements, Admission Difficulty and Campus Life. Then it exploits similarities in rating behavior amongst several students in determining how to recommend University/College. For each student algorithm suggest the list of Engineering Colleges similar to the preferable ones based on students past preferences. It works on the principle that the Students' have same likings in the past will have similar choices in the future as well. The overall rating for University/College is predicted based on past ratings regarding both individual criteria's and overall rating, finally recommends university to the student with best overall score.

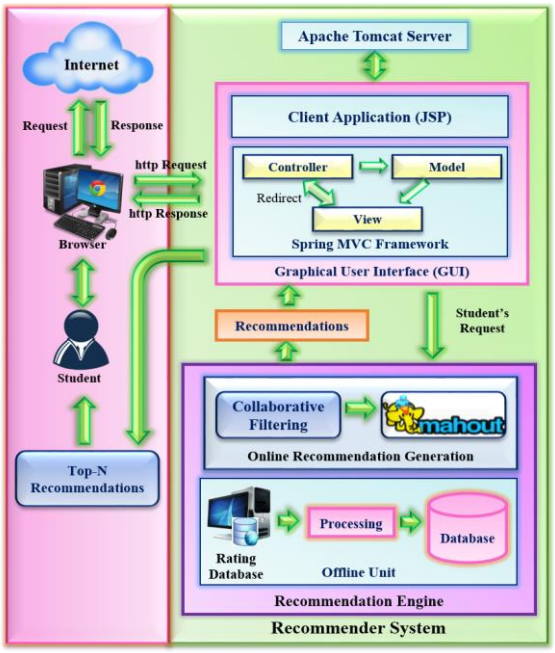

Fig. 2. System Architecture of University Recommendation System

The recommendation engine is the most important component of the URS. Where the actual processing for generating recommendation takes place. The recommendation engine has two main components namely Offline Unit and Online Recommendation Core. Offline unit is used for data processing i.e., to translate the students and colleges records into suitable format. Dataset preprocessing, Data cleaning and Data reformating operatins are performed in this phase. While the recommendation core uses the MC-CF algorithm to recommend Universities/Colleges to the Students. Proposed algorithm is implemented using the Apache Mahout framework. Mahout is an open source Machine Learning Library provides the framework for Recommender System. The algorithm also uses the Mahout's MapReduce to execute in distributed environment.

B. *Research Methodology*

This section presents the research methodology to create multi-criteria dataset and proposed methodology.

1) Multi-Criteria Dataset Creation: For URS we have created the multi-criteria dataset in two steps as follows:

a) Preparation of the list of Universities and Engineering Colleges: A comprehensive profiling of all the Universities and Engineering Colleges was undertaken to list out that award degrees in India. The process began with generating the list of Universities and Engineering Colleges using secondary data sources such as the internet, published survey reports, AICTE and the Association of Indian Universities websites. A list of more than 511 India Universities and 255 Engineering Colleges in Maharashtra State was drawn up. To be precise, we categorized all the Universities by their type as: Central, State, Deemed and Private Universities and University affiliated Engineering Colleges and Engineering Colleges of National Importance (IIT's, NIT's, IIIT's and Government Institutions).

b) Multi-Criteria Rating Collection: There is need to create multi-criteria dataset for the University/College recommendation for the aspiring students. This is to better understand student's choice and interest and to generate high quality recommendations for the aspirants. For this purpose we have collected the students review for the available list of colleges over the multiple criteria's shown in Table I. Table II shows the sample of multi-criteria dataset. Which shows how students have rated the Universities/Colleges over the 5 criteria's on the scale of 1 to 5 (where, 5 is considered as the best and 1 as the worst) mentioned below.

TABLE I. MULTIPLE CRITERIA'S FOR URS

Criteria 1 (C1): Infrastructure	Criteria 4 (C4): Admission Difficulty
Criteria 2 (C2): Teachers	Criteria 5 (C5): Campus Life
Criteria 3 (C3): Placements	

TABLE II. SAMPLE OF MULTI-CRITERIA DATASET

STUDENT_ID	COLLEGE_ID C_101					COLLEGE_ID C_102					COLLEGE_ID C_103					COLLEGE_ID C_104					COLLEGE_ID C_105				
	C1	C2	C3	C4	C5	C1	C2	C3	C4	C5	C1	C2	C3	C4	C5	C1	C2	C3	C4	C5	C1	C2	C3	C4	C5
101						3	5	3	2	4						3	5	4	3	4					
102	3	1	5	4	1	2	4	2	1	5						4	3	4	5	4					
103	2	3	4	3	3						3	4	4	5	4										
104						5	4	3	3	3															
105	4	1	5	3	1	2	3	4	1	1											4	2	4	3	4
106	4	4	4	4	4						3	4	5	4	4	5	4	3	4	4					
107	5	1	2	4	1	4	5	4	2	4											4	5	4	5	5

2) Proposed Methodology: Figure 3 shows the general framework for proposed method with a combination of Multi-Criteria Item-Based CF and Dimensionality Reduction techniques. Students are recommended with a list of Top-N Colleges using proposed algorithm discovering knowledge from student's multi-criteria ratings and predicting overall rating. Firstly we applied HOSVD with PCA option for dimensionality reduction on 3-order tensor of students' ratings. After applying HOSVD on tensor, we perform cosine-based similarity evaluation, followed by prediction and recommendation.

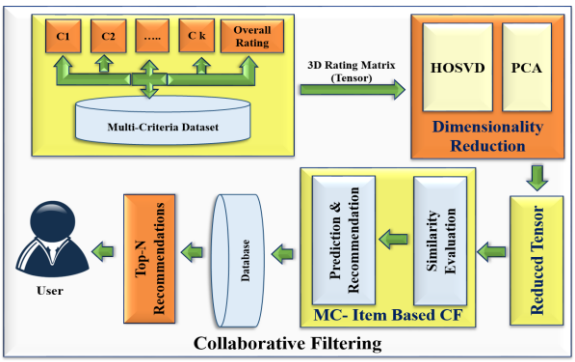

Fig. 3. System Architecture of University Recommendation System

The main task of the dimensionality reduction process by HOSVD with PCA option are reducing the dimension, obtaining the best approximation of the data in the tensor of students preferences about the Colleges on multiple aspects. Finding students preferences with similar preferences on colleges and multiple criteria's.

For the first time we proposed to use PCA mean option in HOSVD. Because HOSVD model in MC-CF cannot deal with missing values as they are treated as "0", to prevent overfitting we have used PCA mean option as regularization. PCA mean option in HOSVD algorithm is used to reduce the sparsity in rating matrix. After generating the unfolding of tensor using HOSVD we approach general PCA and dimensionality reduction with respect to input expressed as row matrix. Where data points are row vectors in such a matrix. Mean of rows in column-vectors is determined [20]. The mean of all rows are determined and subtracted from all data points of the input and then apply the SVD on the unfolding of the tensor. The main steps of proposed method are:

Step 1: Generate the 3-order (student-college-criteria) tensor the from the interaction record i.e., multi-criteria ratings submitted by students for Colleges

Step 2: HOSVD with PCA mean is applied on the 3-order tensor for dimensionality reduction to get best approximation of ratings

Step 3: After tensor decomposition and tensor approximation, the lower dimensional approximated data is used for similarity evaluation using cosine-similarity measure

$$Similarity = cos(A, B) = \frac{A \cdot B}{||A|| * ||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Step 4: Selecting the active student and active colleges and predicting the individual criteria rating using the neighborhood formation for predicting the unknown overall rating can be defined as:

$$Overall\ Rating = \frac{f(Criteria\ 1 + Criteria\ 2 + \cdots + Criteria\ K)}{K\ (No.\ of\ Criteria's)}$$

Step 5: After overall rating prediction, proposed algorithms makes the predictions and list of Top-N College recommendations for the students

IV. EXPERIMATS AND EVALUATION

A prototype of URS is implemented for experimental purpose. Student's feedback in the form of multi-criteria rating are used. Prototype system is a web based application implemented using Java, Java Server Pages (JSP) and Spring MVC Framework. Multi-Criteria Item-Based HOSVD Collaborative Filtering algorithm is implemented using the Apache Mahout Framework.

After almost two decades of research on RS's many researchers came up with various evaluation metrics. To evaluate Information Filtering performance best suitable measures are prediction accuracy and performance. To measure the quality of proposed URS system this section presents the evaluation metrics to evaluate the prediction accuracy and performance.

A. *Experimental Setup and Dataset*

For experiment purpose computing resources are provided using virtual machine. Our experiments were performed on Intel (R) Core (TM) i5-4200 CPU @ 1.60 GHz 2.30 GHz with 64-Bit processor with 8GB memory. The used libraries includes Apache Mahout 0.9, Apache Hadoop 0.2 and Java 7.

For experiment purpose, large real time data is required. For initial experiment purpose we are using Multi-Criteria University dataset that we have created for performing experiment. To recommend the list of Top-N Universities/Colleges to Students based upon their preferences we have performed experiments on the following datasets. There are total 511 Indian Universities and 255 Engineering Colleges of Maharashtra State.

B. *Quality of Recommendations*

Prediction accuracy metrics measures the recommender's predictions that are close to the true users rating. The two

commonly used metrics used for prediction accuracy are Precision and Recall [3-6].

1) Precision: Precision is defined as the ratio of relevant item to recommended item. Precision is given by the formula

$$Precision = \frac{|Interesting\ Items \cap Recommended\ items|}{|Recommended\ items|}$$

2) Recall: Recall is defined as the proportion of relevant items that have been recommended to the total number of relevant items. Recall is given by the formula

$$Recall = \frac{|Interesting\ Items \cap Recommended\ items|}{|Interesting\ Items|}$$

It is desirable for an algorithm to have high precision and recall values. However, both of these metrics are inversely related, such that when precision is increased recall usually diminishes and vice versa.

3) F1 Metric: F1 Metric is defined as the harmonic mean of precision and recall metric. Let β be a parameter that determines the relative influence of both precision and recall. To consider both Precision and Recall the measure F1 metric given by the formula

$$F1\ Metric = \frac{(1+\beta^2)\ (Recall \times Precision)}{\beta^2 \times Recall + Precision}$$

For β = 1

$$F1\ Metric = \frac{2\ (Recall \times Precision)}{Recall + Precision}$$

These evaluation metrics can be used to calculate and compare the prediction accuracy and quality of proposed MC-CF algorithm with existing ones and decide which algorithm performs better. Comparison of the result analysis is given in Fig. 4 for precision, in Fig. 5 for recall and in Fig. 6 for F1 Metric of Multi-Criteria Item-Based Collaborative Filtering and proposed solution implemented using Apache Mahout.

4) Execution Time: In this experiment, we compare the excution time of Multi-Criteria Item Based Collaborative Filtering considered as the baseline with the proposed algorithm, shown in Fig. 7.

C. Result Analysis

Result analysis of the experiments are presented in this section. Currently the experiment has been performed on the dataset with 20 Students and 15 Engineering Colleges. Each student have rated the Colleges they are currently studying or studied in past, over five criteria. Sample of multi-criteria dataset is given in Table II over the integer score ranged from 1 to 5, the higher score is better. All the 5 criteria's are considered throughout the course of experiments, while the number of students and number of colleges are variable.

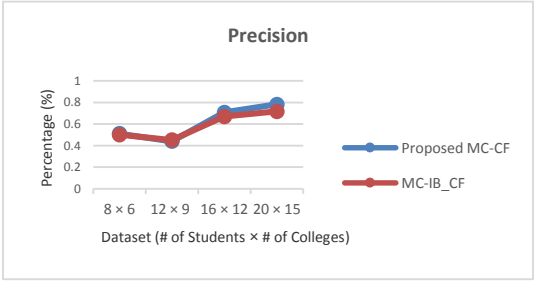

Fig. 4. Result analysis of MC-CF techniques with respect to Precision

From the Fig. 4, it can be seen that both the curve increases as the dataset size increases. However, the precision value of the proposed algorithm is larger than MC-IB_CF method. Which means more accurate results can be obtained using proposed method.

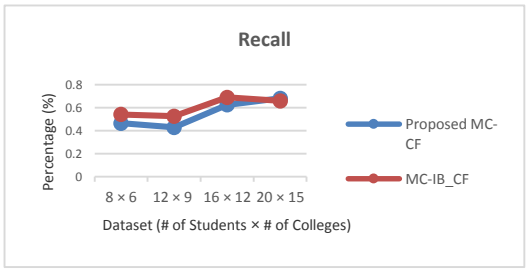

Fig. 5. Result analysis of MC-CF techniques with respect to Recall

From the Fig. 5, it can be seen that both the curve increases as the dataset size increases. However, the recall value of the proposed algorithm is less than MC-IB_CF method. Which means more accurate results can be obtained using proposed method.

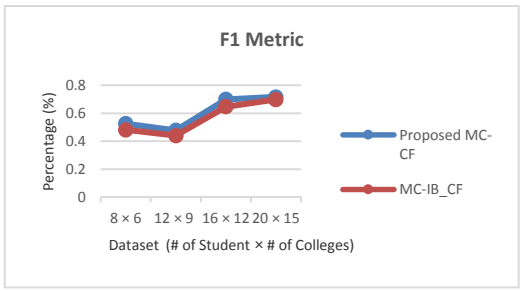

Fig. 6. Result analysis of MC-CF techniques with respect to F1 Metric

From the Fig. 6, it can be seen that both the curve increases as the dataset size increases. However, the F1 value of the proposed algorithm is larger than MC-IB_CF method. Which means more accurate results can be obtained using proposed method.

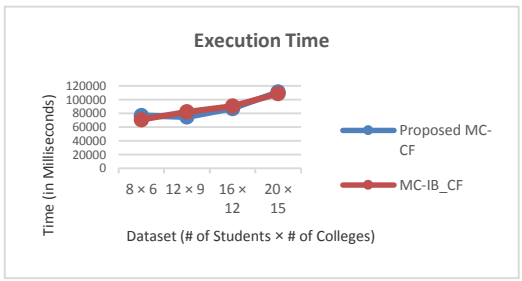

Fig. 7. Result analysis of MC-CF techniques with respect to Execution Time

From the curves in Fig. 7, we have observed that execution time of proposed algorithm is less or nearly same as that of MC-IB_CF. Hence, proposed algorithms performance is better than MC-IB_CF. Due to sparsity in rating i.e., few Students have rated the same University/College, which results slightly increase in the Execution Time, which can be outperformed with increasing the size of dataset. Due to the availability of limited time and dataset only few experiments can be performed and demonstrated in this paper. We expect that more experiments can be performed on massive dataset. We will work to collect more rating by the Student on additional Engineering Colleges and expand the system in future.

V. Conclusion and Future Scope

In this paper we have applied Dimensionality Reduction techniques to deal with the MC-CF challenges to implement the University Recommendation System. From this study we can say that, HOSVD is able to handle large dataset and scalability problem of MC-CF algorithm efficiently. HOSVD model in MC-CF cannot deal with missing values as they are treated as "0". To prevent overfitting in HOSVD, we have proposed to use PCA option as a regularization algorithm to tackle with the sparsity problem of rating matrix.

The URS has been evaluated by using students' multi-criteria rating. Form the experiment it is observed that proposed method generates the high quality University/College recommendations for the Students. It is being also observe that when the sparsity of the rating matrix reduces the HOSVD algorithm performs better. Thus, the proposed solution not only reduces the computation cost but also increases the prediction accuracy and efficiency of the MC-CF algorithm, implemented using the Apache Mahout. By utilizing Mahout API's, the response times across algorithm were significantly improved.

For future work we will plan to implement the University Recommendation System in distributed environment using Apache Spark, another open source project that is an engine for large-scale distributed data processing. Furthermore, we will focus on further improvement of the Multi-Criteria Collaborative Filtering by using different tensor decomposition techniques with dynamic process using massive multi-criteria dataset by considering Content-Based Recommendation.

References

[1] Francesco Ricci, Lior Rokach, Bracha Shapira and Paul B. Kantor, "Recommender Systems Handbook," *Springer*, ISBN: 978-0-387-85819-7, ©Springer Science + Business Media LLC, 2011.

[2] David Goldberg, David Nichols, Brian M. Oki and Douglas Terry, "Using Collaborative Filtering to Weave an Information Tapestry," *Communications of the ACM*, Vol. 35, No.12, December 1992.

[3] Fidel Cacheda, Victor Carneiro, Diego Fernandez, and Vreixo Formoso, "Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-Performance Recommender Systems," *ACM Transactions on the Web*, Vol. 5, No. 1, Article 2, ©ACM, February 2011.

[4] R. Burke, P. Brusilovsky, A. Kobsa and W. Nejdl, "Hybrid web recommender systems," In The Adaptive Web: Methods and Strategies of Web Personalization, *Volume-4321 of Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York*, pp. 377 – 408, 2007.

[5] Che-Rung Lee, Ya-Fang Chang, "Enhancing Accuracy and Performance of Collaborative Filtering Algorithm by Stochastic SVD and Its MapReduce Implementation," *Published in IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum*, IEEE Computer Society, IEEE 978-0-7695-4979-8/13, ©IEEE, 2013.

[6] Mehrbakhsh Nilashi, Othman bin Ibrahim and Norafida Ithnin, "Multi-Criteria Collaborative Filtering with High Accuracy using Higher Order Singular Value Decomposition and Neuro-Fuzzy System," *Published in Knowledge-Based Systems*, 0950-7051, ©Elsevier B.V., pp 82-101, Jan 2014.

[7] Lieven De Lathauwer, Bart De Moor and Joos Vandewalle, "A Multilinear Singular Value Decomposition," *SIAM. J. Matrix Anal. & Appl*., ©Society for Industrial and Applied Mathematics, Vol. 21, No. 4, pp. 1253-1278, 2000.

[8] Göran Bergqvist and Erik G. Larsson, "Higher-Order Singular Value Decomposition: Theory and an Application," *IEEE Signal Processing Magazine*, 1053-5888 , (27)3, pp. 151-154, ©IEEE, May 2010.

[9] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas and Nuria Oliver, "Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering," *ACM RecSys'10*, ©ACM 978-1-60558-906-0/10/09, pp.79-86, Barcelona, Spain, September 26–30, 2010.

[10] Sarwar B., Karypis G., Konstan J., Riedl J., "Item-based Collaborative Filtering Recommendation Algorithms," *Published in the Proceedings of the 10th international conference on World Wide Web*, Hong Kong, ACM 1581133480/01/0005, ©ACM, May 15, 2001.

[11] Alper Bilge and Cihan Kaleli, "A Multi-Criteria Item-Based Collaborative Filtering Framework," *Published in 11th (JCSSE) International Joint Conference on Computer Science and Software Engineering,"* IEEE 978-1-4799-5822-1/14, © IEEE, 2014.

[12] Yehuda Koren, "Matrix Factorization Techniques for Recommender Systems," *Published by the IEEE Computer Society*, IEEE 0018-9162/09, pp. 42- 49, ©IEEE, August 2009.

[13] M.G. Vozalis and K.G. Margaritis, "Using SVD and demographic data for the enhancement of generalized Collaborative Filtering," *Published in An International Journal of Information Sciences 177(2007)*, 0020-0255, pp. 3017-3037, © Elsevier Inc., February 2007.

[14] SongJie Gong, HongWu Ye and YaE Dai, "Combining Singular Value Decomposition and Item-based Recommender in Collaborative Filtering," *Second International Workshop on Knowledge Discovery and Data Mining*, 978-0-7695-3543-2/09, pp. 769-772, © IEEE, 2009.

[15] Manolis G. Vozalis and Konstantinos G. Margaritis, "A Recommender System using Principal Component Analysis," *Published in 11th Panhellenic Conference in Informatics*, Patras, Greece, pp. 271-283, 18-20 May, 2007.

[16] Dheeraj kumar Bokde, Sheetal Girase, Debajyoti Mukhopadhyay, "Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey," *4th International Conference on Advances in Computing, Communication and Control (ICAC3-2015) Proceedings*, Mumbai, India, © Elsevier B.V. Procedia Computer Science, USA, April 03-04, 2015, ISBN xxx-x-xxxx-xxxx-x

[17] Sebastian Schelter, Sean Owen, "Collaborative Filtering with Apache Mahout," ACM RecSysChallenge'12, Dublin, Ireland, September 13, 2012.

[18] Carlos E. Seminario, David C. Wilson, "Case Study Evaluation of Mahout as a Recommender Platform," ACM RecSysChallenge'12, Dublin, Ireland, pp. 45-50, September 13, 2012.

[19] Sachin Walunj, Kishor Sadafale, "An Online Recommendation System for E-commerce Based on Apache Mahout Framework," ACM SIGMIS-CPR'13, 978-1-4503-1975-1/13/05, Cincinnati, Ohio, USA, May-30-June-1, ©ACM, 2013.

[20] Apache Mahout, Available at URL: *http://mahout.apache.org/*

[21] Apache Hadoop, Available at URL: *http://hadoop.apache.org/*