

# Cosine Based Latent Factor Model for Precision Oriented Recommendation

Bipul Kumar

Fellow Program in Management,  
Operation Management and Decision  
Sciences  
India Institute of Management, Ranchi  
Ranchi, India

Pradip Kumar Bala

Associate professor, Information  
Systems  
India Institute of Management, Ranchi  
Ranchi, India

Abhishek Srivastava

Fellow Program in Management,  
Information Systems  
India Institute of Management, Ranchi  
Ranchi, India

**Abstract**—Recommender systems suggest a list of interesting items to users based on their prior purchase or browsing behaviour on e-commerce platforms. The continuing research in recommender systems have primarily focused on developing algorithms for rating prediction task. However, most e-commerce platforms provide ‘top-k’ list of interesting items for every user. In line with this idea, the paper proposes a novel machine learning algorithm to predict a list of ‘top-k’ items by optimizing the latent factors of users and items with the mapped scores from ratings. The basic idea is to learn latent factors based on the cosine similarity between the users and items latent features which is then used to predict the scores for unseen items for every user. Comprehensive empirical evaluations on publicly available benchmark datasets reveal that the proposed model outperforms the state-of-the-art algorithms in recommending good items to a user.

**Keywords**—collaborative filtering; recommender systems; precision; e-commerce; machine learning

## I. INTRODUCTION

In the age of ‘internet of things’, there is a growing importance of personalized recommender systems (RS). RS are typical software solutions used in E-commerce for personalized services [1]. It helps customers to find interesting items by providing recommendations based on their prior preferences viz., amazon.com suggests a list of items based on the purchase history of a user. It also benefits E-commerce portals that offer millions of products for sale by targeting the right customer for the right product [2]. Due to growing significance of RS, several techniques for developing the recommendation systems have been studied. These include content-based filtering (CBF), collaborative filtering (CF), and hybrid based recommender system. Among them, the CF technique has been widely used due to its simplicity and effectiveness, and has also proven to be useful in many practices [3].

Recommender Systems (RS) collect information on the preferences of its users for a set of items. The information can be collected explicitly and/or implicitly. RS may also use demographics of users such as age, location, gender. There is a growing trend of utilizing social information like followers, followed, tweets for personalized recommendation [4].

The fundamental assumption of CF is that “if users X and Y rate n items similarly, or have similar behaviors (e.g.,

buying, watching, listening), and hence will rate or act on other items similarly” [5]. CF techniques use a database of preference for items (e.g., movies, songs, books, travel destination) by users to predict additional items that may be of interest to users. In a typical CF database, there is a list of users (say m users) and items (say n items) where each user either explicitly (typically, by extracting users’ preferences in form of star rating) or implicitly (typically, by monitoring purchase history, browsing history or even mouse clicks) indicate their preferences corresponding to items [6]. Since, every user cannot look into every item when there are millions of items in E-commerce setup therefore; the preferences are also not available for most of user-item pair. In order to generate recommendation list, an active user can be recommended items with help of other users who have indicated similar preferences for items in CF database.

CF is classified into two types: memory based CF and model based CF. Memory based CF are those which generates recommendation list based on similarity measures either between user-user or between item-item. The similarity measures generally employed are cosine similarity or Pearson correlation similarities which are quite effective. Model based CF learns parameters of models using data mining, machine learning algorithm on training data. The learned parameters are used to predict real data. Latent factor models, Bayesian networks, latent dirichlet allocation(LDA) and Markov decision process based models are frequently researched models in model based CF[5].

Latent factor models, such as Singular value decomposition (SVD), has been quite popular in research of RS as it has been regarded to be the best single method in improving the accuracy in Netflix prize[7]. SVD transforms both items and users to the same latent factor space, thus making them directly comparable. The SVD model as used in Netflix prize learns item bias and user bias which are independent of the features being used for characterisation for items and users.

The previous works in RS literature have focussed more to solve rating prediction task; however the prime objective of RS is to present ‘top-k’ good items in the recommendation lists for every user. Therefore, this work focuses on ‘top-k’ recommendation instead of rating prediction. This means that formulation of the problem has to be transformed as

classification problem, where the task is to classify the good and the bad items. Based on the above arguments, this paper has proposed an innovative algorithm which is a fusion of similarity concepts and latent factor model. The latent factors of user and items are learnt based on the degree of similarity between user and item. The assumption of the proposed model is, more the similarity between user and items, the probability of liking the item by the user would be higher and vice-versa. Experiments for validating the effectiveness of our approach were conducted using benchmark datasets in RS.

The rest of this paper is organized as follows. Section 2 reviews previous studies regarding latent factor models implemented in recommendation systems. Section 3 describes the problem at hand in formal manner. Section 4 describes the proposed model with pseudo code; next section explains how our approach makes a difference based on the results from experiments and also describes the implications of the experiments. Lastly, conclusion is drawn based on observations.

## II. RELATED WORK

Since the user-item matrix is often sparse due to unavailability of feedback from most of the users for most of the items, it is often difficult to incorporate memory based CF techniques in RS successfully.

One of the approaches to deal with the problem of sparseness is by adapting model based approach in CF. SVD is used in model based CF which reduces dimensionality of user-item matrix and identifies latent factor in the data [8]. An application of SVD in the context of information retrieval has already been patented and is named as Latent Semantic Indexing (LSI).

Some of the early works in RS by applying SVD has been adopted with appropriate modifications which are different from applications in information retrieval. Daniel Billsus and Michael J. Pazzani in their paper [9] described CF algorithm as classification problem. At first sparse user item matrix is first converted to Boolean feature matrix for every user based on items rated by the user. Subsequent to it the Boolean feature matrix is decomposed using SVD by taking 'k' number of dimensions to be retained. Neural network is used to train the singular vectors and thereafter for prediction [9]. Since the method described is a bit complex and not scalable for real time recommendations, Sarwar, Karypis, Konstan, & Riedl in their paper describe about the methodology of SVD which is directly applied in RS [10]. The user-item matrix which is sparse has to be filled up by user average rating or item average rating. After this pre-processing step SVD is applied on the resultant filled matrix. SVD decomposes the matrix into two matrices; two of which are orthogonal matrices and one is diagonal matrix or singular matrix.

The user-item matrix has to be imputed (assigned value) at the first step before proceeding to SVD which has been criticized by researchers in the field, as imputations led to over-generalization and accuracy of the method is lost. However, the start of SVD was remarkable in the context of recommender system and it solved the problem of sparsity to an extent [11] but there arises a different set of problem as it

happens in very large data set, which often is the case of real world, the complexity and computation of user-item matrix increases exponentially with increasing user item dataset. There is also a need to update the recommendation real time in order to have the most accurate recommendation. In order to address the complexity and computation time problem of SVD can be solved by following a technique proposed by author known as folding-in in SVD [11].

However, it was only until the Netflix prize (Netflix, 2006) that the SVD approach was accepted to be the best single method in RS. Simon Funk popularized the regularized SVD method for the first time to explore the Netflix prize data in order to make accurate prediction [12]. Subsequent to this, modification to the basic regularized SVD was proposed for the Netflix prize dataset. The top prize winner in Netflix prize [13], stressed on augmenting the basic SVD with popular neighborhood based technique. The author suggested incorporating implicit feedback as well as explicit feedback in the same model for the best prediction which was being evaluated on RMSE [6].

Singular value decomposition as a method has also been incorporated along with other available feature of dataset to accurately predict the ratings in case of movie recommender system. SVD combined with demographic data is also proposed to improve the approach of collaborative filtering. The reason of using demographic data along with SVD is to supplement the collaborative filtering algorithm [14].

While deterministic latent factor models such as SVD have been successfully implemented and made popular, probabilistic latent factor models also were considered in information retrieval and subsequently in RS. Thomas Hofmann in his paper utilized the statistical base as a primary reason of using probabilistic latent semantic analysis (pLSA) [15].

Since pLSA has a drawback that exact estimating the ratings is intractable, which means that potentially slow or inaccurate approximations are required for computing the posterior distribution over hidden factors in such model [16]. Full Bayesian analysis of model was done later in 2008 by same authors and called it as probabilistic matrix factorization (pmf) to overcome the problem of inaccurate prediction. The model can be viewed as probabilistic extension of SVD. Using Markov Chain Monte Carlo (MCMC), pmf training is also done to avoid tuning of parameters manually which is required to avoid over-fitting [17].

A relatively similar approach to pLSA is Latent Dirichlet Allocation (LDA). Latent Dirichlet Allocation (LDA) is similar to pLSA in the sense that latent variables are present in a probabilistic way. While pLSA does not assume a specific prior distribution over number of dimensions in hidden variable, LDA assumes that priors have the form of the Dirichlet distribution [18]. Gibbs sampling is used to estimate the parameters in LDA model [19]. The Expectation Maximization (EM) algorithm and its variation can also be used in solving the parameters of the model.

Continuing with matrix factorization method to discover latent factor models, there are other approaches as well which

have been used in the field of RS. One more way of utilizing the matrix factorization so that sparse data can be handled more effectively is a model named Eigentaste that uses principal component analysis for optimal dimensionality reduction and then clusters users in the lower dimensional subspace. As these are model based collaborative filters, they are operated in two modes; online and offline mode. The online mode uses Eigen vectors to project new users into clusters and a lookup table to recommend appropriate items so that run time is independent of the number of users in the database [20].

Matrix factorization is not the only way to handle latent factor models. Discrete Wavelet Transform (DWT) has been used for data reduction without deterioration in signal processing and image processing earlier. Influenced by the technique in handling sparse data DWT has also been used in RS. The technique illustrated is the unique way applied in data reduction in RS to best of our knowledge. The argument presented by author based on previous research illustrates that PCA and SVD find feature combinations that model the largest contributions in a dataset, but these may not be the same features that differentiate attributes, as weaker relationships may be lost [21].

Restricted Boltzmann Machine (RBM) has also been used in order to solve the sparse and large data set such as that of Netflix [22]. RBM introduced for learning the Netflix dataset used a class of two-layer undirected graphical models, suitable for modeling tabular or count data, and presented efficient learning and inference procedures for this class of models.

This sums up the related work using latent factor models in RS. Since, the previous models based on latent factors are guided by the loss function that optimizes the actual ratings; they couldn't quite assist in 'top-k' prediction task. In order to build a model that can handle the prediction of 'top-k' good items to every user, this paper proposes a loss function based on cosine similarity between user and item latent feature. In the next section we will describe the problem at hand in formal and detailed manner.

### III. PROBLEM SETTING

In a typical E-commerce setup, there are millions of users and thousands of products listed in database. The user specifically searches for products which he is willing to purchase; with each transaction of a user we can build his purchase history and behavior so forth. The building of a user's preference based on purchase history is termed as implicit feedback. Also, a user may show his explicit preference for a product by providing ratings; viz. 1 to 5 stars. The building of explicit preferences for a user-item pair is termed as explicit feedback. Based on the feedbacks user-item matrix is obtained, consisting of rows representing the users, columns representing the items and elements of matrix are ratings of user for an item.

Practically, not all users may show their preferences for all the items either implicitly or explicitly, which gives rise to sparsity in user-item matrix. This poses a challenge in the recommendation task. In order to model such practical scenarios in research we have tested our models on

MovieLens (ml100k) data set and FilmTrust dataset. The dataset is publicly available for research and has been used in many research papers dealing with recommender system [4]. The proposed model first learns the latent features of users and items using cosine similarity as loss function, and later score of the unseen items for a user is generated. Top 'k' items based on predicted scores can be recommended to a user in descending order of the predicted score.

#### A. Notations

For distinguishing users from items special indexing letters have been used for user and items – a user is denoted by "i", and an item is denoted by "j". A rating  $r_{ij}$  indicates the preference of a user i for item j, where high values mean stronger preference and low values mean low preference or no preference for an item i. For example in a range of "1 star" to "5 stars", "1 star" rating means lower interest by a particular user u for a given item i and "5 stars" rating means high interest by user u for a given item i. A mapped score s is obtained from the ratings by passing through a suitable function is described in next section. The parameters  $\mathbf{P}_i$  and  $\mathbf{Q}_j$  denotes the user and item features respectively and are in form of a vector. In this paper the 'bold' notations denotes the vector and the corresponding elements of the vectors are 'normal'.

### IV. PROPOSED MODEL

This section will cover the model building phase of the innovative algorithm for classification of good and bad items. The algorithm is primarily build for classification task but we can also extend this to rating prediction task.

#### A. Cosine based latent factor model

There are a few disadvantages of using matrix factorization to learn the latent factors of users and items. One of the disadvantages is that the function is not bounded; hence there is a possibility of obtaining the predicted values out of range [16]. Since the predicted values may get out of range the predicted values are clipped [12] or passed through a bounded function such as logistic function [16]. This may not be appropriated since the mapping function of actual rating in training set of data is not mapped according to the bounded function and are generally normalized [16].

To furnish a mathematical solution to this problem this work introduces a cosine based latent factor model. The cosine function is bounded between -1 and 1 which gives an advantage to map the actual ratings in train set using cosine function and use cosine latent factor model to learn the features of users and items.

The intuition behind use of pervious latent factor models such as regularized SVD states that interaction between user and item features results in ratings of an item by a user [7]. In the proposed cosine latent factor model the intuition is the degree of similarity between user and item features defines the interest of user for an item. So if a user is highly interested in an item the similarity between the user and item features is close to 1 otherwise, the similarity is close to 0. In order to map the actual ratings  $R_{ij} \in \{1 \dots r\}$  in between 0 and 1 we passed it using a function  $\phi$ . This function has to be defined

such that the minimum rating shall be close to 0 and maximum rating shall be close to 1. One such function is defined below:

$$\emptyset = \frac{e^{(r-r_{\text{mean}})}}{e^{-(r-r_{\text{mean}})} + e^{(r-r_{\text{mean}})}}$$

where,  $r_{\text{mean}}$  is the average of  $\{1 \dots r\}$ . The ratings are passed through this function to obtain a mapped score  $s$ . For obtaining the latent factors of each item and user the proposed cosine latent factor model is set equivalent to obtained mapped scores ( $s$ ). This leads to minimizing the following objective function.

$$f = \min_{P_i, Q_j} \sum_{(ij \in \kappa)} \left( s_{ij} - \frac{P_i Q_j^T}{\sqrt{P_i^2} \sqrt{Q_j^2}} \right)^2 + \lambda (\|P_i\|_{\text{Fro}}^2 + \|Q_j\|_{\text{Fro}}^2) \quad (1)$$

Here,  $\|\cdot\|_{\text{Fro}}$  denotes the Frobenius norm.

Regularization parameter  $\lambda$  is introduced to make a balance between over-fitting and variance. The optimum value of the minimization function can be obtained by using stochastic gradient descent method. For every iteration, learning rate ( $\alpha$ ) is multiplied against the slope of descent of the function in order to reach local minima. The partial derivatives with respect to  $P_i$  and  $Q_j$  results in gradient of descent for this function.

$$\frac{\partial f}{\partial P_i} = -2 \left( s_{ij} - \frac{P_i Q_j^T}{\sqrt{P_i^2} \sqrt{Q_j^2}} \right) \left( \frac{\sqrt{P_i^2} Q_j^T - \left( \frac{P_i Q_j^T \sqrt{Q_j^2} P_i}{P_i^2 Q_j^2} \right)}{P_i^2 Q_j^2} \right) + 2\lambda P_i \quad (2)$$

$$\frac{\partial f}{\partial Q_j} = -2 \left( s_{ij} - \frac{P_i Q_j^T}{\sqrt{P_i^2} \sqrt{Q_j^2}} \right) \left( \frac{\sqrt{P_i^2} Q_j^2 P_i - \left( \frac{P_i Q_j^T \sqrt{P_i^2} Q_j^T}{\sqrt{Q_j^2}} \right)}{P_i^2 Q_j^2} \right) + 2\lambda Q_j \quad (3)$$

### 1) Model learning

Algorithm:

Input:

- $R$  : A matrix of rating, dimension  $N \times M$  (user item rating matrix)
- $\kappa$  : Set of known ratings in matrix  $R$
- $P_i$  : An initial vector of dimension  $N \times F$  (User feature vector)
- $Q_j$  : An initial vector of dimension  $M \times F$  (item feature matrix)
- $F$  : Number of latent features to be trained
- $s_{ij}$  : mapped scores obtained after passing through function  $\emptyset$

Parameters:

- $\alpha_1$  : learning rate

- $\lambda$  : over fitting regularization parameter
- Steps : Number of iterations

Output: A matrix with scores to generate recommendation list

Method:

- 1) Initialize random values to vector  $Q_j, P_i$
  - 2) Fix value of  $F, \alpha_1$  and  $\lambda$ .
  - 3) do till error converges [ error(step-1) - error(step) <  $\epsilon$  ]
- error (step) =

$$\min_{P_i, Q_j} \sum_{(ij \in \kappa)} \left( s_{ij} - \frac{P_i Q_j^T}{\sqrt{P_i^2} \sqrt{Q_j^2}} \right)^2 + \lambda (\|P_i\|_{\text{Fro}}^2 + \|Q_j\|_{\text{Fro}}^2)$$

for each  $R \in \kappa$

Update training parameters

$$P_i \leftarrow P_i + \alpha_1 \left( \frac{\partial f}{\partial P_i} \right)$$

$$Q_j \leftarrow Q_j + \alpha_1 \left( \frac{\partial f}{\partial Q_j} \right)$$

end for

- 1) Return  $P_i, Q_j$

The obtained  $P_i, Q_j$  for each user and items are used to predict a score using the following equation.

$$\hat{s}_{ij} = \frac{P_i Q_j^T}{\sqrt{P_i^2} \sqrt{Q_j^2}} \quad (4)$$

Based on these, the scores are arranged in descending order and top 'k' items for each user can be generated in the recommendation list.

Further, to extend this model for rating prediction task, we will use the calculated similarity score between the user and unrated item, user and all other items rated by active user, based on learned latent features. The top n-nearest neighbours to the unrated items are scanned based on calculated similarity score and their average is used to predict the rating for the unrated item.

## V. EXPERIMENTATION AND EVALUATION

In this section the experimental setup and evaluation protocol to test the proposed model on two publicly available datasets have been presented. The proposed model is compared with baseline and other state-of-art algorithms. The proposed algorithms are evaluated both on classification and rating prediction using appropriate performance measures.

### A. Datasets

For the experimental evaluations of the proposed method, two different datasets are used. The first one is a publicly available Movie Lens dataset (ml-100k). The dataset consists of ratings of movies provided by users with corresponding user and movie IDs. There are 943 users and 1682 movies

with 100000 ratings in the dataset. Had every user would have rated every movie total ratings available should have been 1586126 (i.e. 943×1682); however only 100000 ratings are available which means that not every user has rated every movie and dataset is very sparse (93.7%). This dataset resembles an actual scenario in E-commerce, where not every user explicitly or implicitly expresses preferences for every item.

The second dataset consists of movie reviews from FilmTrust [23]. There are 1508 users and 2071 movies with only 35497 ratings. The sparsity levels (98.86%) are more than movieLens dataset.

**B. Cross-Validation**

The dataset is partitioned into 5 equal disjoint sets with 4 datasets used for training and one left out dataset for testing the model. The process is repeated five times, as a procedure adopted for 5-fold cross-validation. On testing dataset the accuracy measure such as RMSE, and precision is calculated and averaged over the 5-folds which is a procedure adopted to nullify the effect of biasness of partitioning the sample

**C. Performance Metrics**

In a classification task the performance metrics that determines the top ‘k’ as used in recommendation systems are

1) *Precision*: Precision is defined as the ratio of relevant items, Nrs, recommended to the total number of items, Ns, recommended to a user.

$$\text{Precision} = \text{Nrs} / \text{Ns}$$

In rating prediction task, the goal is to minimize the difference in ratings between predicted and actual ratings. In order to evaluate the accuracy, RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) are popular metrics. The variants of RMSE and MAE such as Normalized RMSE and Normalized MAE or average RMSE and average MAE are also used. The predicted ratings ( $\hat{r}_{ij}$ ) for a test set ‘ $\Gamma$ ’ of user-item pairs (i, j) for which true item ratings ( $r_{ij}$ ) are known, the RMSE is given by

$$\sqrt{\frac{1}{|\Gamma|} \sum_{(i,j) \in \Gamma} (\hat{r}_{ij} - r_{ij})^2}$$

$|\Gamma|$  is number of observations in test set

MAE on the other hand is given by

$$\frac{1}{|\Gamma|} \sum_{(i,j) \in \Gamma} |\hat{r}_{ij} - r_{ij}|$$

**D. Evaluating the performance of models**

In this section, the performances of the proposed model with already existing state-of-the-art algorithm in this field are evaluated. One of the state-of-the-art algorithms is RSVD that was designed primarily for rating prediction task. For experimentation purposes, the number of latent features (F) is varied from 10 to 100 in steps of 10 for proposed model and RSVD algorithm. Firstly, the focus is on classification task where the idea is to present ‘top-k’ items to each user. Based on the obtained predicted ratings, in case of RSVD, and

obtained scores, in proposed model, top 5 and top 10 items are presented to the user. The predicted rating and predicted scores respectively in descending order are presented to every user and then accuracy measures such as precision are obtained [24].

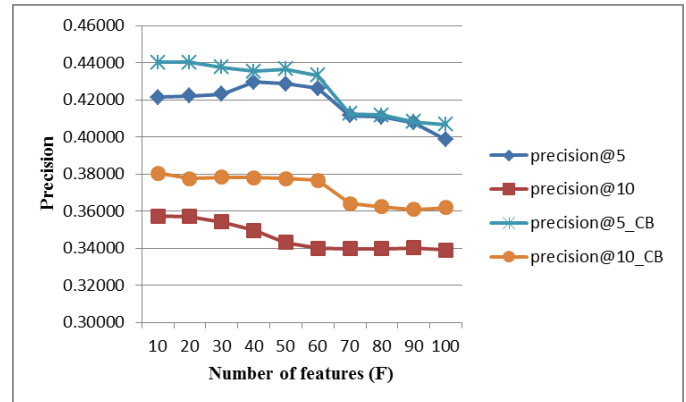


Fig. 1. precision of proposed cosine latent factor model and RSVD on ml-100k dataset

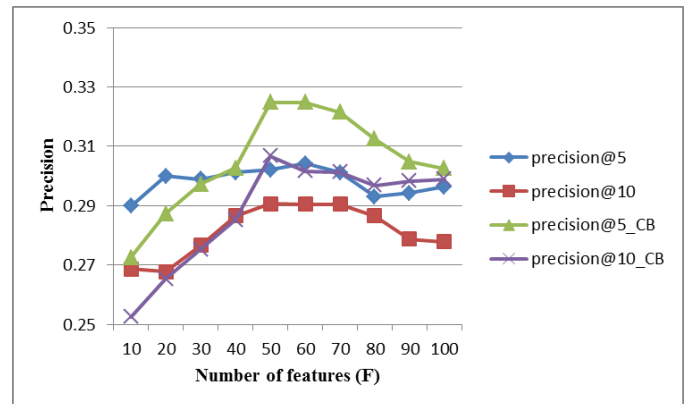


Fig. 2. precision of proposed cosine latent factor model and RSVD on FilmTrust dataset

Figure 1 and figure 2 show the precision of the proposed cosine based latent factor model and RSVD on ml-100k dataset and FilmTrust dataset respectively. Since the precision are computed for top 5 and top 10 items presented to each user Precision@5 and Precision@10 are used to denote in the figure 1 and 2. Precision@5\_CB shows the precision as obtained on datasets by applying cosine based latent factor model, while precision@5 shows the precision as obtained by RSVD. In ml-100k dataset, the highest precision for cosine based latent factor model occurs when the number of features (F) for user and item are 10. Correspondingly, the highest value of precision for RSVD occurs for F=10 but the values of precision at both top 5 and top 10 items is better for the proposed cosine latent factor model than state-of-the-art RSVD. There is an improvement of 4.5% for precision@5 and 5% for precision@10 over RSVD algorithm on ml-100k dataset. In case of FilmTrust dataset, the maximum precision@5 for cosine based latent factor model and RSVD occurs when the F value is 60. For precision@10, the maximum value occurs at F=50 for both cosine based latent factor model and RSVD. Here, cosine based latent factor

model outperforms RSVD for precision@5 by approximately 6.5% and for precision@5 by approximately 5%.

In rating prediction task, the predicted ratings obtained from both RSVD and proposed cosine latent factor model with varying latent features (F) are compared with actual ratings in the test set using 5-fold cross-validation. The latent features (F) in both RSVD and proposed cosine latent factor model are varied from 10 to 100 in steps of 10, the cross-validated MAE and RMSE are obtained for both the two datasets and compared.

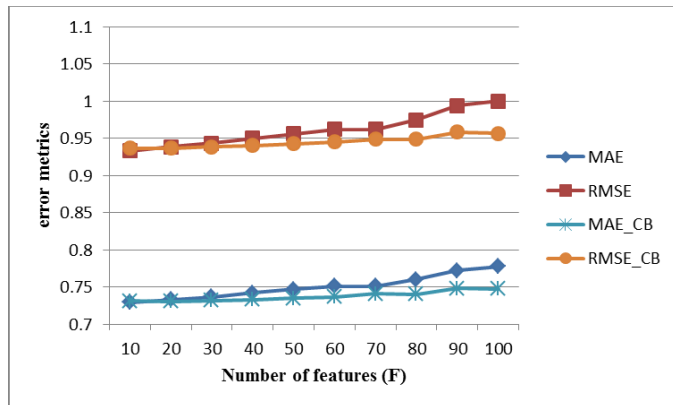


Fig. 3. Error metrics of proposed cosine latent factor model and RSVD on ml-100k dataset

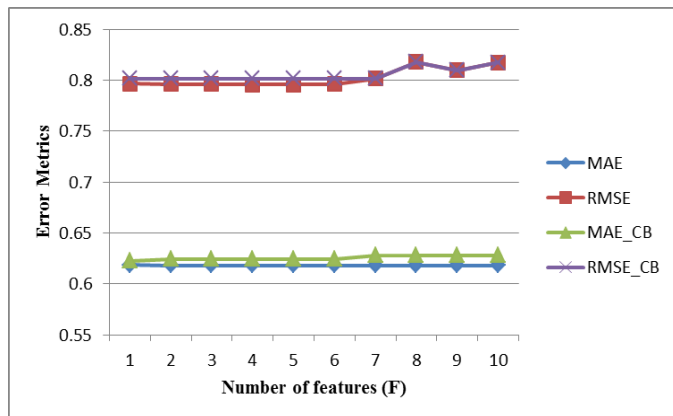


Fig. 4. Error metrics of proposed cosine latent factor model and RSVD on FilmTrust dataset

From the figure 3 and figure 4, one can see that although, MAE and RMSE for RSVD is better than cosine based latent factor model on both the datasets, the difference is negligible for the best value obtained from both these algorithms.

Thus, this work has shown through empirical experimentation that the proposed cosine latent factor model outperforms state-of-the-art algorithm in RS in terms of precision. Also, the proposed model gives comparable results in terms of MAE and RMSE for rating prediction task. In modern e-commerce retail, like amazon, alibaba, the users are presented with a set of recommended products based on their prior purchase and browsing behaviour. Therefore, our work focuses primarily on this aspect of recommending top 'k' items for every user.

## VI. DISCUSSIONS AND CONCLUSIONS

In this research work, we have proposed a novel algorithm that caters to recommending top 'k' items for each user. Our work primarily falls in domain of model based RS with focus on classification of good and bad items. This work introduces the concept of cosine similarity based latent factor model which is a unique algorithm in itself. Also, the rationale behind using cosine similarity latent factor model over RSVD is theoretically sound. As mentioned in the paper that the rating prediction using RSVD often goes out of bounds as the loss function is not bounded, the loss function used in the proposed model is bounded and therefore the prediction do not go out of the bounds. One more advantage in using the proposed model is its ability to handle difficult (outlier) data points due to its inherent property of bounds which are not observed in RSVD model.

In future, we look forward to utilize the proposed cosine latent factor model in field of information retrieval and also in ranking prediction task for both recommender system and information retrieval. The learning method of optimization can also be suitably modified to learn the parameters of the proposed model a bit faster. One of the other approaches of using the proposed model for the above task can be by ensemble of weak learners generated by varying the number of latent factors of the model. The present work is primarily designed for top 'k' recommendation task but has also been extended to rating prediction task by using simple average of the ratings of most similar items are applied. The rating prediction using more sophisticated techniques like clustering of the item features can also be obtained to check any improvement. The techniques being applied has to be carefully chosen as they may increase the complexity without improving the accuracy adequately.

### REFERENCES

- [1] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of Recommendation Algorithms for e-Commerce," in Proceedings of the 2Nd ACM Conference on Electronic Commerce, 2000, pp. 158–167.
- [2] Y. S. Kim, B.-J. Yum, J. Song, and S. M. Kim, "Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites," *Expert Syst. Appl.*, vol. 28, no. 2, pp. 381–393, Feb. 2005.
- [3] M. D. Ekstrand, "Collaborative Filtering Recommender Systems," *Found. Trends@ Human-Computer Interact.*, vol. 4, no. 2, pp. 81–173, 2010.
- [4] J. Bobadilla, F. Ortega, A. Hernando, and A. Guti rrez, "Recommender Systems Survey," *Knowl. Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.
- [5] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1–19, Jan. 2009.
- [6] Y. Koren, "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model," in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 426–434.
- [7] Y. Koren and R. Bell, "Advances in Collaborative Filtering," in *Recommender Systems Handbook*, 2011, pp. 145–186.
- [8] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A Constant Time Collaborative Filtering Algorithm," no. August, 2000.
- [9] D. Billsus and M. J. Pazzani, "Learning Collaborative Information Filters," in Proceedings of the Fifteenth International Conference on Machine Learning, 1998, pp. 46–54.
- [10] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-a case study," in *ACM WebKDD 2000 Workshop*, 2000.

- [11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Incremental singular value decomposition algorithms for highly scalable recommender systems," *Fifth Int. Conf. Comput. Inf. Sci.*, pp. 27–28, 2002.
- [12] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proc. KDD Cup Workshop at SIGKDD'07*, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining, 2007, pp. 39–42.
- [13] Y. Koren, "The bellkor solution to the netflix grand prize," *Netflix Prize Doc.*, no. August, pp. 1–10, 2009.
- [14] M. Vozalis and K. Margaritis, "Using SVD and demographic data for the enhancement of generalized Collaborative Filtering," *Inf. Sci. (Ny)*, vol. 177, no. 15, pp. 3017–3037, Aug. 2007.
- [15] T. Hofmann, "Latent Semantic Models for Collaborative Filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 89–115, Jan. 2004.
- [16] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization," pp. 1–8, 2007.
- [17] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 880–887.
- [18] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
- [19] R. Krestel, P. Fankhauser, and W. Nejdl, "Latent Dirichlet Allocation for Tag Recommendation," 2009.
- [20] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A Constant Time Collaborative Filtering Algorithm," *Inf. Retr. Boston*, vol. 4, no. 2, pp. 133–151, Jul. 2001.
- [21] S. Russell and V. Yoon, "Applications of Wavelet Data Reduction in a Recommender System," *Expert Syst. with Appl.*, vol. 34, no. 4, pp. 2316–2325, May 2008.
- [22] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann Machines for Collaborative Filtering," in *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 791–798.
- [23] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel bayesian similarity measure for recommender systems," *IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 2619–2625, 2013.
- [24] A. Bellogin, P. Castells, and I. Cantador, "Precision-oriented Evaluation of Recommender Systems: An Algorithmic Comparison," in *Proceedings of the Fifth ACM Conference on Recommender Systems*, 2011, pp. 333–336.