Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 49 (2015) 136 – 146

Procedia Computer Science

ICAC3'15

Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey

Dheeraj Bokde[a], Sheetal Girase[b], Debajyoti Mukhopadhyay[c]*

[a, b, c] *Department of Information Technology*
Maharashtra Institute of Technology, Pune 411038, India

Abstract

Recommendation Systems (RSs) are becoming tools of choice to select the online information relevant to a given user. *Collaborative Filtering* (CF) is the most popular approach to build Recommendation System and has been successfully employed in many applications. Collaborative Filtering algorithms are much explored technique in the field of Data Mining and Information Retrieval. In CF, past user behavior are analyzed in order to establish connections between users and items to recommend an item to a user based on opinions of other users. Those customers, who had similar likings in the past, will have similar likings in the future. In the past decades due to the rapid growth of Internet usage, vast amount of data is generated and it has becomea challenge for CF algorithms. So, CF faces issues with sparsity of rating matrix and growing nature of data. These challenges are well taken care of by *Matrix Factorization* (MF). In this paper we are going to discuss different Matrix Factorization models such as Singular Value Decomposition (SVD), Principal Component Analysis (PCA) and Probabilistic Matrix Factorization (PMF). This paper attempts to present a comprehensive survey of MF model like SVD to address the challenges of CF algorithms, which can be served as a roadmap for research and practice in this area.

Peer-review under responsibility of organizing committee of the 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15)

Keywords: Collaborative Filtering ; Matrix Factorization ; Recommendation System

* Corresponding author.
E-mail address: debajyoti.mukhopadhyay@gmail.com

Peer-review under responsibility of organizing committee of the 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15)
doi:10.1016/j.procs.2015.04.237

1. Introduction

In these days E-Commerce industry is growing at the exponential rate. As the users are using these systems their changing needs and variety of products is making business more and more complex. At the same time they are providing ease and flexibility to users'. In such environment, customers have difficulty to find optimal information about products from the tremendous amount of information. To help the buyers, the major e-business companies are going to develop their Recommender System to help their customers to choose items more efficiently; this serves win-win strategy in E-Commerce.

Recommendation Systems are becoming tools of choice to select the online information relevant to a given user. Recommendation System can be classified into: Content-Based (CB), Collaborative Filtering (CF) and Hybrid Recommendation System[1,2]. CF is the most popular approach to build recommender system and has been successfully employed in many applications. CFis much explored technique in the field of Data Mining and Information Retrieval. In *Collaborative Filtering* (CF), past user behavior are analyzed in order to establish connections between users and items to recommend an item to a user based on opinions of other users'. Customers who had similar likings in the past, will have similar likings in the future. Many E-Commerce companies have already incorporated RS with their services. Examples for such recommendation systems include Product and Book recommendation by Amazon, Movie recommendations by Netflix, Yahoo!Movies and MovieLens, Product advertisements shown by Google based on the search history.

For large and complex data, CF methods frequently give better performance and accuracy than other RS techniques. Early CF algorithms for recommendation systems utilize the association inferences, which have a very high time complexity and a very poor scalability. Recent methods that use matrix operations are more scalable and efficient. The implementations and algorithms of CF for the applications of recommendation systems face several challenges. First is the size of processed datasets. The second one comes from the sparseness of rating matrix, which means for each user only a relatively small number of items are rated. So, these challenges are been well taken care by Matrix Factorization[2, 3].

Matrix Factorization (MF) methods have recently received greater exposure, mainly as an unsupervised learning method for latent variable decomposition and dimensionality reduction[3]. It has successfully applied in spectral data analysis and text mining. Most of the MF models are based on the latent factor model. In a latent factor model[2, 3], rating matrix is modeled as the product of a user factor matrix and an item factor matrix. The Matrix Factorization approach is found to be most accurate approach to reduce the problem from high levels of sparsity in RS database, certain studies have used dimensionality reduction techniques.

MF is specially used for processing large RSs databases and providing scalable approaches. In the model-based technique Latent Semantic Index (LSI) and the dimensionality reduction method Singular Value Decomposition (SVD) are typically combined[2, 3, 14]. SVD and PCA are well-established technique for identifying latent semantic factors in the field of Information Retrieval to deal with CFchallenges.

1.1. Related Work

This paper mainly study the Matrix Factorization models like SVD and PCA, with CF algorithms such as user-based and item-based CF. As we know that from past two decades lots of research work is going in the field of CF. CF is a promising research field in Information Retrieval so many researchers have contributed to this area.

Many CF researchers have recognized the problem of sparseness i.e., many values in the ratings matrix are null since all users do not rate all items. Computing distances between users is complicated by the fact that the number of items users have rated in common is not constant. An alternative to inserting global means for null values or significance weighting is Singular Value Decomposition (SVD), which reduces the dimensionality of the ratings matrix and identifies latent factors in the data[8].

In 2006, the online DVD rental company Netflix announced the Netflix Prize contest with a $1 million reward to the first team who can improve its recommender system's root mean square error (RMSE) performance by 10 percent or more[3, 19]. Contestants were allowed to build model based on released training set consisting of about 100 million movie ratings, on a scale of 1 to 5 stars, submitted by 500,000 anonymous users on more than 17,000 movies. The participating teams need to submit their predicted ratings for a test set consisting of approximately 3 million ratings and Netflix calculated the RMSE based on a held-out truth set. This large size of publicly available data create a perfect setting for standardized benchmarking experiments and attracted significant attention to the field of recommender system in particular CF.

The team Yehuda Koren et al.[3] originally called BellKor, took over the top spot in the competition in the summer 2007 and won the 2007 Progressive Prize with the best score at the time 8.43 percent better than Netflix. Later in 2008 they aligned with team BigChaos to win the Progressive Prize with a score of 9.46 percent. Factorizing the Netflix user-movies matrix allows them to discover the most descriptive dimensions for predictive movie preferences. They identified the first few most important dimensions from a matrix decomposition and explore the movies in new space.

Sarwar et al. (2001)[5] applied Matrix Factorization model SVD to reduce the dimensionality of a ratings matrix. Using the MovieLens dataset, they selected 943 users to form a 943 × 1682 matrix that is 95.4% sparse (each user on average rates 5% of the 1682 movies). They first fill missing values using user and movie rating averages, and then apply SVD. For thislarge dataset Item-Based technique provide optimal accuracy with significantly faster and high quality online recommendationsthan user-user (k-Nearest-Neighbor) method.

Goldberg et al.[15] proposed an approach to use Principal Component Analysis (PCA) in the context of an online joke recommendation system. Their system, known as Eigentaste[15], In Eigentaste they addressed sparseness using universal queries, which insure that all users rate a common set of k-items. So, resulting sub-matrix will be dense and directly compute the square symmetric correlation matrix and then did linear projection using Principle Component Analysis (PCA), a closely-related factor analysis technique first described by Pearson in 1901. Like SVD, PCA reduces dimensionality of matrix by optimally projecting highly correlated data along a smaller number of orthogonal dimensions.

Royi Ronen et al.[17] proposed a project Sage, Microsoft's all-purpose recommender system designed and developed as an ultra-high scale cloud service. The main focus of project Sage is on both state of the art research and high scale robust implementation. A novel Probabilistic Matrix Factorization (PMF) model was presented by Royi et al. for implicit one-class data as new evaluation framework. Their service Sage is deployed on the Microsoft Azure cloud which provides easy to use interface to integrate a recommendation service into any website. Recommender Systems based on MF models have repeatedly demonstrated better accuracy than other methods, such as Nearest-Neighbor models and restricted Boltzmann machines. The dashboard allows users to choose any subset of items and generate high quality recommendations as well as explore item-to-item relation.

1.2. Organization

The rest of this paper is organized as follows. In Section 2, the background of Collaborative Filtering, CF techniques, MFis presented. In Section 3, MF models are discussed. In Section 4, MF model like SVD with user-based, item-based CF algorithms is presented Section 5, presents the various evaluation metrics to evaluate accuracy and implementation of MF models with CF algorithms. The conclusion and future work are given in the last Section.

2. Background

This section presents the overview of CF and MF with their methodologies. GenerallyCF techniques are classified as: Memory-Based, Model-Based and Hybrid approaches with representative algorithms (detailed explanation in Section 4).

2.1. Collaborative Filtering (CF)

The CF algorithms of Recommendation System (RS) works by collecting user feedback in the form of ratings for items. Then it exploits similarities in rating behavior amongst several users in determining how to recommend an item. It works on the principle thatthe user' have same likings in the past will have similar choices in the future as well. The term CF 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[4] was an electronic messaging system that allows users' to rate messages "good" or "bad" or associate text annotations with those messages. CF systems recommend an item to a user based on opinions of other users. In a recommendation application, CF system tries to find other like-minded users and then recommends the items that are most liked by them.

The E-Commerce industry is growing and becoming complex therefore information about products is also increasing with exponential rate, demands more efficient and scalable algorithms and implementations. For large and complex data, CF methods frequently give better performance and accuracy than Content-Based technique of Recommendation System. Earlier CF algorithms for recommendation systems used to utilize the association inferences, which have a very high time complexity and a very poor scalability. Recent methods make use of matrix operations which are more scalable and efficient.

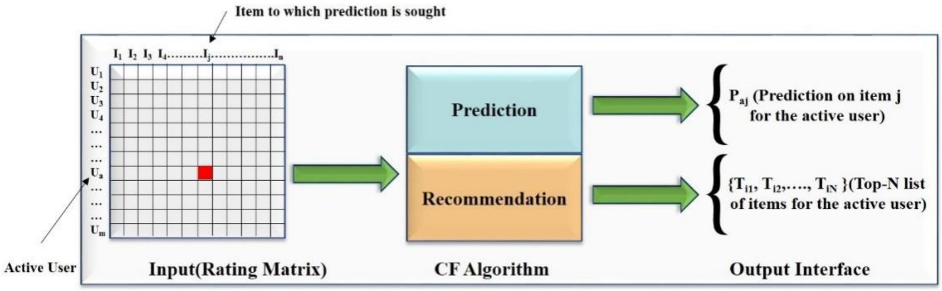

Fig. 1. The Collaborative Filtering Process.

The task of CF algorithm is to find an item likeliness that can be well described by schematic diagram of collaborative filtering process shown in Fig. 1:

- Predict a numerical value *Paj* expressing the predicted score of an item *'j'* for the user *'a'*. The predicted value is within the same scale that is used by all users for rating
- Recommend a list of *Top-N* items that the active user will like the most

Collaborative Filtering (CF) Techniques

The CF techniques can be classified into various categories, such as: a) Memory-Based Collaborative Filtering, b) Model-Based Collaborative Filtering, c)Hybrid Collaborative Filtering. Overview of these categories is shown in the Table 1: (referred from the *Xiaoyuan Su homepage for CF).

Table 1. Overview of Collaborative Filtering Techniques*

Collaborative Filtering Technique	Representative Algorithm	Advantages	Limitations
Memory-Based Collaborative Filtering (Neighborhood Based)	• User-Based CF • Item-Based CF	• Easy implementation • New data can be added easily and incrementally • Need not consider the content of items being recommended • Scales well with correlated items	• Are dependent on human ratings • Cold start problem for new user and new item • Sparsity problem of rating matrix • Limited scalability for large datasets
Model-Based Collaborative Filtering	• Slope-One CF • Dimensionality Reduction (Matrix Factorization) Eg. SVD, PCA	• Better addresses the sparsity and scalability problem • Improve prediction performance	• Expensive model building • Trade-off between the prediction performance and scalability • Loss of information in dimensionality reduction technique (SVD)
Hybrid Collaborative Filtering	• Combination of Memory-Based and Model-Based CF	• Overcome limitations of CF such as sparsity and grey sheep • Improve prediction performance	• Increased complexity and expense for implementation

2.2. Matrix Factorization (MF)

The most successful latent factor models are based on MF[3, 6, 14, 24]. In its basic form MF characterizes both items and users by vectors of factors inferred from items rating patterns. The high correspondence between user factors and item factor leads to a recommendation. These methods have become popular recently by combining good scalability with predictive accuracy. They offers much flexibility for modeling various real-life applications.

Matrix factorization models map both users and items to a joint latent factor space of dimensionality *f*, user-item interactions are modeled as inner products in that space. Accordingly, each item *i* is associated with a vector q_i □ R^f, and each user *u* is associated with a vector $p_u \in R^f$. For a given item *i*, the elements of q_i measure the extent to which the item possesses those factors positive or negative. For a given user *u* the elements of p_u measure the extent of interest the user has in items that are high on the corresponding factors positive or negative[3]. 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[3]:

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

The Matrix Factorization approach is found to be most accurate approach to reduce the problem from high levels of sparsity in RS databases, certain studies have used dimensionality reduction techniques. Matrix factorization is specially used for processing large RS databases and providing scalable approaches[3, 14, 24]. The model-based technique Latent Semantic Index (LSI) and the reduction method Singular Value Decomposition (SVD) are typically combined. SVD and PCA are well-established technique for identifying latent semantic factors in Information Retrieval. Applying SVD in the CF domain requires factoring the user-item rating matrix[3, 8]. This raises difficulties due to the high portion of missing values caused by sparseness in the user-item ratings matrix. The conventional SVD is undefined when knowledge about the matrix is incomplete.

To learn the factor vectors (p_u and q_i), the system minimizes the regularized squared error on the set of known ratings as[3]:

$$ \min_{q^*,\ p^*} \sum_{(u,i) \in K} (r_{ui} - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2) \qquad (2) $$

Here, K is the set of the (u,i) pairs for which r_{ui} is known the training set.

The system learns the model by fitting the previously observed ratings. The goal is to generalize those previous ratings in a way that predicts future unknown ratings. The constant λ controls the extent of regularization and is usually determined by cross-validation.

3. Matrix Factorization (MF) Models

Matrix decomposition is a powerful technique to find the hidden structure behind the data. SVD, PCA, PMF and NMF,etc.,are popular decomposition models. They can be reformulated as an optimization problem with a loss function and constraints, for example, NMF imposes the no negativity on factor matrices. How to choose the loss function and the constraints are dependent on the property of the data.There are various matrix factorization models, some commonly are as shown in Fig. 2 above are:

- Singular Value Decomposition (SVD)
- Principal Component Analysis (PCA)
- Probabilistic Matrix Factorization (PMF)
- Non-Negative Matrix Factorization(NMF)

3.1. Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) is the powerful technique of dimensionality reduction. It is a particular realization of the MF approach and also related to PCA. The key issue in an SVD decomposition is to find a lower dimensional feature space.

SVD of an $\boldsymbol{m} \times \boldsymbol{n}$ matrix **A** is of the form[2, 7, 8, 9]:

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

Where,

$\boldsymbol{U}$ and $\boldsymbol{V}$ are $\boldsymbol{m} \times \boldsymbol{m}$ and $\boldsymbol{n} \times \boldsymbol{n}$ orthogonal matrices respectively
$\boldsymbol{\Sigma}$ is the $\boldsymbol{m} \times \boldsymbol{n}$ singular orthogonal matrix with non-negative elements

An $\boldsymbol{m} \times \boldsymbol{m}$ matrix $\boldsymbol{U}$ is called orthogonal if $\boldsymbol{U^T U}$equals to an $\boldsymbol{m} \times \boldsymbol{m}$ identity matrix. The diagonal elements in $\boldsymbol{\Sigma}$ $(\sigma_1, \sigma_2, \sigma_3, \ldots\ldots \sigma_n)$ are called the singular valuesof matrix $\boldsymbol{A}$. Usually, the singular values are placed in thedescending order in Σ. The column vectors of $\boldsymbol{U}$ and $\boldsymbol{V}$ arecalled the left singular vectors and the right singular vectorsrespectively.

SVD has many desirable properties and is used in many important applications. One of them is the low rank approximation of matrix $\boldsymbol{A}$. The truncated SVD of rank k is defined as[2, 7, 8, 9]:

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

Where,

$\boldsymbol{U_k}$ and $\boldsymbol{V_k}$ are $\boldsymbol{m} \times \boldsymbol{k}$ and $\boldsymbol{n} \times \boldsymbol{k}$ matrices composed by the first $\boldsymbol{k}$ columns of matrix $\boldsymbol{U}$ and the first $\boldsymbol{k}$ columns of matrix $\boldsymbol{V}$ respectively. Matrix $\boldsymbol{\Sigma_k}$ is the $\boldsymbol{k} \times \boldsymbol{k}$ principle diagonal sub-matrix of $\boldsymbol{\Sigma}$.
$\boldsymbol{A_k}$represents the closest linear approximation of the original matrix $\boldsymbol{A}$ with reduced rank $\boldsymbol{k}$.

Once the transformation is completed, user and items can be thought off as points in the $\boldsymbol{k}$-dimensional space.

3.2. Principal Component Analysis (PCA)

The Principal Component Analysis (PCA) is also the powerful technique of dimensionality reduction and is a particular realization of the MF approach. PCA is a classical statistical method to find patterns in high dimensionality data sets.

Goldberg et al.[15]proposed an approach to use PCA in the context of an online joke recommendation system. Their system, known as Eigentaste[15], starts from a standard matrix of user ratings to items. PCA allows to obtain an ordered list of components that account for the largest amount of the variance from the data in terms of least square errors[2, 6, 10, 12]. The amount of variance captured by the first component is larger than the amount of variance on the second component and so on. We can reduce the dimensionality of the data by neglecting those components.

3.3. Probabilistic Matrix Factorization (PMF)

Probabilistic Matrix Factorization (PMF) is a probabilistic linear model with Gaussian observation noise[11]. The Probabilistic Matrix Factorization (PMF) models the user preference matrix as a product of two lower-rank user and item matrices. Suppose we have N users and M movies. Let R_{ij} be the rating value of user i for movie j, U_i and V_j represent D-dimensional user-specific and movie-specific latent feature vectors respectively[11, 13].

The conditional distribution over the observed ratings $R \in \mathbb{R}^{N\times M}$and the prior distributions over $U \in \mathbb{R}^{D\times N}$ and V $\in \mathbb{R}^{D\times M}$ are given by in[11, 13, 18, 20]:

$$p(R|U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2\right]^{I_{ij}}$$

$$p(U|\sigma_U^2) = \prod_{i=1}^{N}\mathcal{N}(U_i|0,\sigma_U^2 I) \qquad p(V|\sigma_V^2) = \prod_{j=1}^{M}\mathcal{N}(V_j|0,\sigma_V^2 I)$$

Where,

$\boldsymbol{\mathcal{N}(x|\mu,\sigma^2)}$denotes the Gaussian distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma^2}$
$\boldsymbol{I_{ij}}$ is the indicator variable that is equal to **1** if user $\boldsymbol{i}$ rated movie$\boldsymbol{j}$ and equal to **0** otherwise.

3.4. Non-Negative Matrix Factorization (NMF)

Non-negative matrix factorization (NMF) was first proposed by Paatero and Tapper in 1994and was greatly popularized by Lee and Seung (Lee and Seung, 1999), also called as non-negative matrix approximation. NMF is a group of algorithms in multivariate analysis and linear algebra where a matrix X is factorized into two matrices P and Q, with the property that all three matrices have no negative elements[22, 23].

Let the input data matrix X = ($x_1, x_2, \ldots\ldots, x_n$) contain a collection of n data vectors as columns.We consider factorizations of the form:

$$X \approx PQ^T$$

Where, $X \in \mathbb{R}^{N\times M}$, $P \in \mathbb{R}^{D\times N}$ and $Q \in \mathbb{R}^{D\times M}$

For example, the SVD can be written in thisform. In the case of the SVD, there are no restrictions on the signs of P and Q, moreoverthe data matrix X is also unconstrained. NMF can also be written in this form, where thedata matrix X is assumed to be nonnegative, as are the factors P and Q.

As compared to other matrix factorization approaches, NMF takes into account the fact that most types of real-world data, particularly all images or videos are nonnegative and maintain such non-negativity constraints in factorization[22,23].

4. Matrix Factorization (MF) Model inNeighborhood-Based Collaborative Filtering (CF)

To address the challenges of CF algorithms many researchers have used the Matrix Factorization models like SVD and PCA with CF algorithms.The Neighborhood based or Memory-Based CFalgorithms usually use similarity metrics to obtain the similarity between two users, or two items based on each of their ratios as explained in Section 2.1.1. CF algorithms can be further divided into user-based and item-based approaches. This section explores the user-based CF and item-based CF as well as their implementation with MF model like Singular Value Decomposition (SVD)[8, 14].

4.1. User-Based Collaborative Filtering using Singular Value Decomposition (SVD)

4.1.1. User-Based Collaborative Filtering

User-Based Collaborative Filtering approach was proposed in the end of 1990s by the professor Jonathan L. Herlocker of University of Minnesota. In the user-based approach, the users perform the main role[2]. In this algorithm we first bind the similarity among users or group of users' based on their similarity. After that the algorithms recommend each user the items suggested by the other users of the same group.

In User-Based Collaborative Filtering systems[2, 5]:

Step 1: Build similarity between the users who shares the same rating pattern with active user.

Step 2: Prediction for the active user is calculated by using rating from those like-minded users found in Step 1.

There are some challenges of the user-based CF algorithm. First is sparsity of the user-item rating matrix which can cause the mining of users' similarity difficult and inaccurate. Second is the scalability. When user changes their preferences re-computing is required. To overcome these problems many researchers have used the Matrix Factorization model like SVD in User-Based CF algorithm. In[8]M. G. Vozalis and K.G. Margaritis used SVD and demographic data to perform a series of user-based CF. They showed SVD cannot only solve the sparsity problem but also enhance the accuracy of user-based CF.

4.1.2. User-Based SVD Collaborative Filtering

The User-Based SVD Collaborative Filtering algorithm is as follows[8]:

Step 1 [Preprocessing]:

Build a user-item matrix from the interaction records, and define the size of dimensionality reduction.

Step 2 [Similarity Evaluation]:

This step refers to the Neighborhood formation with each table entry including the corresponding similarity metric equation. Here SVD is applied on the user-item matrix using a slightly different similarity metric equation with the meta-ratings taken from the reduced user-item matrix.

Step 3 [Rating Process]:

The calculations of the correlations using the enhanced correlation equation on the rating based correlation using original user-item matrix and similarity matrix.

Step 4 [Recommendation]:

Conclude the filtering process with prediction generation formulas using the SVD applied on the user-item matrix.

4.2. Item-Based Collaborative Filtering using Singular Value Decomposition (SVD)

4.2.1. Item-Based Collaborative Filtering

The Item-Based CF approach was proposed by the researchers of University of Minnesota in 2001. Item-Based CF method explores the similarity between the items first. For each user they suggest the item similar to the preferable ones of the user. According to the long tail theory[2], as long as the number of users is large enough, each item can have significant similarity computation even the user-item matrix is sparse. Item-Based CF used by Amazon.com proceeds in an item-centric manner. In the user-item rating matrix items are usually much less than those of users and do not change on the fly. Therefore the item-based CF method have better scalability than user-based CF method.

The algorithm for item-based CF is as follows[2, 5]:

Step 1: Find the relationship between the pair of items from the item-item matrix

Step 2: The tastes of active user is examined by matching rating matrix with users' preferences

The SVD a Matrix Factorization model is used in the item-based CF to classify user/item information to avoid the loss of transitive similarity relation. M. G. Vozalis and K.G. Margaritis used SVD and demographic data to perform a series of item-based CF[8]. They showed SVD cannot only solve the scalability and sparsity problem but also enhance the accuracy of item-based CF.

4.2.2. Item-Based SVD Collaborative Filtering

The Item-Based Collaborative Filtering algorithm with SVD is as follows[8]:

Step 1 [Preprocessing]:

Build a user-item matrix from the interaction records, incorporate the application of SVD depending on the implementation and define the size of dimensionality reduction.

Step 2 [Similarity Evaluation]:

This step refers to the Neighborhood formation with each table entry including the corresponding similarity metric equation. Here implementation of SVD is applied on the user-item matrix using a slightly different similarity metric equation with the meta-ratings taken from the reduced user-item matrix.

Step 3 [Rating Process]:

(a)The calculations of demographic correlations and computing the vector similarity between the corresponding item vectors. Those vectors are taken from either from original demographic matrix or from the reduced demographic user-item matrix.

(b)Apply the enhanced correlation equation on the rating-based correlations and on the demographic correlations from Step 3 (a).

Step 4 [Recommendation]:

Concludes the filtering process with prediction generation formulas using the SVD applied on the user-item matrix by utilizing the item ratings.

4.2.3. Item-Based Stochastic SVD (SSVD) Collaborative Filtering

The biggest problem of using SVD in CF algorithms is its high computational cost. To overcome this problem Nathan Halko's dissertation and Halko, Martinsson, Tropp contributed in their research work to Mahout[16]. Mahout's StochasticSVD (SSVD) algorithm implementation can be parallelized easily is more suitable in distributed computing environment specially w.r.t. Mahout Lanczos implementation on a typical corpus data set[16].

The algorithm for an $\boldsymbol{m} \times \boldsymbol{n}$ matrix $\boldsymbol{A}$ using item-based CF using SSVDis as follows[16]:

Given:

An $m \times n$ matrix A, a target rank k, an oversampling parameter p, and a number of power iterations q, the following algorithm computes an approximate rank k and T is the transpose of matrix. Then Singular Value Decomposition (SVD) of matrix A is given by, **SVD (A)** $\approx U\Sigma V^T$.

Algorithm:

Draw an $n \times (k+p)$ random matrix $\mathbf{\Omega}$
Form the product $\mathbf{Y} = \mathbf{A\Omega}$
Orthogonalize the columns of $\mathbf{Y} \rightarrow \mathbf{Q}$
for $i = 1 \ldots\ldots q$
Form the product $\mathbf{Y} = \mathbf{AA^TQ}$
Orthogonalize the columns of $\mathbf{Y} \rightarrow \mathbf{Q}$
end
Form the projection $\mathbf{B} = \mathbf{Q^TA}$
Compute the factorization $\mathbf{\widetilde{U}\Sigma^2\widetilde{U}^T} = \mathbf{BB^T}$
If needed, compute $\mathbf{U} = \mathbf{Q\widetilde{U}}$
If needed, solve $\mathbf{V} = \mathbf{B}_q^{\mathbf{T}}\mathbf{\widetilde{U}\Sigma^{-1}}$

The Item-Based CF algorithm using Stochastic SVD (SSVD) not only provide accurate results but also reduces the computational cost[16]. The only potential limitation of Item-Based SSVD CF is that it is potentially less precise. To overcome this challenge we need the improved implementation of an efficient algorithm and deeper research.

5. Evaluation Metrics

After almost two decades of research on CF algorithms various researchers came up with many evaluation metrics. This section presents the various evaluation metrics used to evaluate the prediction accuracy, effective implementation of the MF models with CF algorithms in Recommendation System (RS).

5.1. Coverage

Coverage is defined as the percentage of items the Recommendation Systems able to recommend to the user. Coverage can be used to detect algorithms accuracy, although they recommend only a small number of items. These are usually very popular items with which the user is already familiar without the help of the system. The term coverage is mainly associated with two concepts[1]:

1. The percentage of the items for which the system is able to generate a recommendation.
2. The percentage of the available items which effectively are ever recommended to a user.

5.2. Prediction Accuracy

Prediction accuracy matrices measures the recommender's predictions that are close to the true users rating. There are various matrices used by the Collaborative Filtering researchers to check the prediction accuracy of their implemented algorithms are as follows[1, 2]:

5.2.1. Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) measures the difference as absolute value between the prediction of the algorithm and the real rating would be given by the user. It is computed using the formula[1]:

$$\mathbf{MAE} = \frac{\sum_i^k (p_i - r_i)}{k}$$

Where,

p_i is the prediction for user i
r_i is the prediction for how user i will rate item i.e., real or true rating value
k is the number of items user i has rated i.e., {user, item} pair

5.2.2. Root Mean Squared Error (RMSE)

The RMSE is related to previous metric i.e., MAE. The reason of using this metric is that these errors can have the greatest impact on the user decision. This can be computed using the formula[1]:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_i^k (p_i - r_i)^2}{k}}$$

5.2.3. Precision and Recall

Precision is defined as the ratio of relevant items to recommended items. Precision can be calculated using the formula:

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

Recall is defined as the proportion of relevant items that have been recommended to the total number of relevant items. Recall is calculated 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 the metrics are inversely related, such that when precision is increased recall usually diminishes and vice versa.

5.2.4. F1 Metric

To consider both Precision and Recall the measure F1 metric is defined by the formula[1]:

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

With the help of these evaluation metrics we can calculate the prediction accuracy and efficiency of the various Collaborative Filtering algorithms and we can decide which algorithm performs better compared to other ones.

6. Conclusion

Collaborative Filtering algorithms are most commonly used in Recommendation System (RS). Due to the use of Internet huge amount of information is generated, it becomes a very tedious task for users to find their preferences. As users' preferences for items are stored in the form of rating matrix, which are used to build the relation between users and items to find users' relevant items. Thus, CF algorithms nowa day face the problem with large dataset and sparseness in rating matrix.

In this paper we have studied variousMatrix Factorization model to deal with the CF challenges. From this study we can say that, SVD is able to handle massive dataset, sparseness of rating matrix, scalability and cold-start problem of user-based/item-based CF algorithm efficiently. Use of SVD model in Neighborhood-Based CF algorithms increases computation cost. To overcome this problem researchers have come up with Stochastic SVD (SSVD) MF model. The SSVD not only reduces the computation cost of Neighborhood-Based CF algorithm but also increases the accuracy, preciseness and efficiency of the CF algorithms.

References

1. 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.
2. 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.
3. Yehuda Koren, "Matrix Factorization Techniques for Recommender Systems," *Published by the IEEE Computer Society*, IEEE 0018-9162/09, pp. 42- 49, ©IEEE, August 2009.
4. 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.
5. 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.
6. Daryl Lim, "Matrix Factorization Presentation," University of California, San Diego, February 7, 2013. Available at URL: http://www.academia.edu/3695261/Matrix_Factorization_Presentation

7. Huseyin Polat and Wenliang Du, "SVD-based Collaborative Filtering with Privacy," *ACM Symposium on Applied Computing SAC'05*, Santa Fe, New Mexico, USA, ACM 1581139640/05/0003, ©ACM, March 13-17, 2005.
8. 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.
9. 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.
10. 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.
11. Ruslan Salakhutdinov and Andriy Mnih, "Probabilistic Matrix Factorization," *Published in the Proceedings of Neural Information Processing Systems Foundation (NIPS'07)*, December 4, 2007.
12. Manolis G. Vozalis, Angelos Markos, Konstantinos G. Margaritis, "Collaborative filtering through SVD-based and hierarchical nonlinear PCA," *Published in ICANN'10 Proceedings of the 20th international conference on Artificial neural networks: Part I*, ISBN: 3-642-15818-8 978-3-642-15818-6, Pages 395-400, ©Springer-Verlag Berlin, Heidelberg, 2010.
13. Zhijun Zhang and Hong Liu, "Application and Research of Improved Probability Matrix Factorization Techniques in Collaborative Filtering," *International Journal of Control and Automation (IJCA)*, ISSN: IJCA 2005-4297, Vol.7, No.8, pp. 79-92, ©SERSC, 2014.
14. Yehuda Koren, "Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model," *Published in the proceedings of KDD'08 of ACM,* Las Vegas, Nevada, USA, ACM 978-1-60558-193-4/08/08, August 24–27, ©ACM, 2008.
15. Ken Goldberg, Theresa Roeder, Dhruv Gupta and Chris Perkins, "Eigentaste: A Constant Time Collaborative Filtering Algorithm," *Published in the Journal of Information Retrieval*, Vol. 4, Issue 2, pp. 133–151, ©Kluwer Academic Publishers Hingham, MA, USA, 2001.
16. Apache Mahout, Available at URL: https://mahout.apache.org/users/dimreduction/ssvd.html
17. Royi Ronen, Noam Koenigstein, Elad Ziklik, Mikael Sitruk, Ronen Yaari, Neta Haiby-Weiss, "Sage: Recommender Engine as a Cloud Service," ACM RecSys'13 Hong Kong, China, ACM 978-1-4503-2409-0/13/10,October 12–16, 2013.
18. Ruslan Salakhutdinov, Andriy Mnih, "Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo," Published in Proceedings of the 25 th International Conference on Machine Learning, Helsinki, Finland, 2008.
19. The Netflix Prize, http://www.netflixprize.com
20. Mohsen Jamali, Martin Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," ACM RecSys '10, New York, NY, USA, ISBN: 978-1-60558-906-0, Pages 135-142, ACM ©2010.
21. Nicolas Gillis, Robert Luce,"Robust Near-Separable Nonnegative Matrix Factorization Using Linear Optimization," Journal of Machine Learning Research 15 (Apr), pp. 1249-1280, 2014.
22. Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan, "NeNMF: An Optimal Gradient Method for Nonnegative Matrix Factorization," Published in IEEE Transactions on Signal Processing, Vol. 60, No. 6, pp. 2882-2898, June 2012.
23. Nguyen Thai-Nghe, Lucas Drumond, Tomáš Horváth, Alexandros Nanopoulos and Lars Schmidt-Thieme,"Matrix and Tensor Factorization for predicting Student Performance," Published in Proceedings of the 3rd International Conference on Computer Supported Education (CSEDU), 2011.